HYBRID ENCRYPTION ALGORITHMS FOR MEDICAL DATA STORAGE SECURITY IN CLOUD DATABASE

Fenghua Zhang[1]， Yaming Chen[2]， Weiming Meng[3] and Qingtao Wu[4]

[1]Computer Technology, Henan University of Science and Technology, Luo Yang, China
[2,3]Computer Science and Technology, Henan University of Science and Technology, Luo Yang, China
[4]Professor, Henan University of Science and Technology, Luo Yang, China

ABSTRACT

Cloud database are derivatives of Cloud computing. At present, most medical institutions store data in cloud database. Although the cloud database improves the efficiency of use, it also poses a huge impact and challenge to the secure storage of data. The article proposes a hybrid algorithm to solve the data security problem in the hospital cloud database. First, the AES algorithm is improved. The improved algorithm is called P-AES algorithm. The P-AES algorithm is then combined with the RSA algorithm, called a hybrid algorithm. The experimental results show that the hybrid encryption algorithm has the advantages of fast encryption and decryption speed, high security, good processing ability for longer data, and can solve the data security problem in cloud database to a certain extent. The algorithm has been successfully applied to hospital information management system.

KEYWORDS

Hospital System, AES, RSA, Hybrid Encryption

1. INTRODUCTION

Hospital information construction is an important measure to improve the level of medical services. Nowadays, most hospitals have a set of perfect medical information management system. The system is divided into several subsystems, such as His, Lis, Pace, etc. These systems work together to ensure the normal operation of hospitals. In addition, it also has a large number of mobile devices and sensors to provide a variety of data for the hospital information management system. Although these software and equipment improve the quality of hospital services, hospitals are also facing tremendous pressure of data storage. If the hospital continues to follow the traditional data storage mode, it will not only make the hospital construction cost more, but also can’t solve the problem of "information island" between hospitals, which does not meet the development requirements of hospital information construction.

The application of cloud computing can solve the above problems. It provides users with low cost, high computing and massive storage of network application services [1]. Cloud database is a kind of database that deploys and virtualizes in cloud computing environment. It enhances the storage capacity of the database, eliminates the repeated configuration of personnel, hardware and software, and makes it easier to upgrade software and hardware. However, users lose absolute control over the storage of data, and the security of the system depends on the information security strategy of cloud system [2, 3]. As a result, data will be attacked by hackers, peeped by service providers, large-scale leaks and other issues. Especially the hospital database stores a large number of users' private information, such as medical record information, doctor's advice

information and prescription information and so on. Protecting data security in cloud databases has become an important issue in the field of information security [4].

2. RELATED WORK

2.1. LITERATURE RESEARCH

Encryption data can effectively solve the data security problem of cloud database. Even if the data is leaked, the criminals only get encrypted data. If they cannot be deciphered, the data will not be of any use to them. At present, the research work mainly focuses on the confidentiality of stored data, security audit and access control [5] and so on.

According to the different types of secret keys, common data encryption algorithms can be divided into symmetric encryption algorithm and asymmetric encryption algorithm (public key encryption algorithm). Symmetric encryption algorithm uses the same secret key to encrypt and decrypt data. It has the characteristics of short secret key length and fast encryption and decryption speed, but it is difficult to manage the secret key [6]. Asymmetric encryption algorithm uses public key encryption and private key decryption. The algorithm features high security, but encryption and decryption involves complex operations, so the speed is slow.
Data Encryption Standard (DES) is a commonly used symmetric encryption algorithm. It is a block encryption algorithm. The standard DES algorithm encrypts data in groups of 64 bits, and the same algorithm is used for encryption and decryption. The key length is 64 bits, which is actually generated by 56 bits of secret key and additional 8 bits of parity check bits. The algorithm has high encryption and decryption efficiency, but the key length is too short and the security of the algorithm is relatively low [7].

In order to solve the problem of short length and low security of DES secret key. A triple DES (3DES) encryption method was proposed. This algorithm is equivalent to performing DES encryption algorithm three times for each data block, which is a distortion of DES algorithm. The 3DES algorithm is based on the DES algorithm. It is a packet encryption algorithm designed by combining packets. This method expands the key length from 56 bits to 112 bits, so the efficiency of encryption and decryption is very low. However, with the development of symmetric encryption technology, DES and 3DES are gradually replaced by the new encryption algorithm AES (Advanced Encryption Standard).

As a new generation of data encryption standard, AES combines security, performance, efficiency, simplicity and flexibility. These features make AES the preferred choice in security-related applications [8]. In recent years, experts and scholars have done a lot of research on the efficiency and security of AES algorithm. Literature [9] provides a safer and more economical encryption mechanism, which randomizes the key of AES algorithm and hides the key data into the encrypted digital image by using the basic concepts of cryptography and digital watermarking. Then the pixels are repositioned to break the correlation between them. Literature [10] based on chaos theory, the AES algorithm S-box is replaced by another S-box. The new S-box has low correlation and shows significant performance improvement with increasing complexity. Document [11] proposes a method to provide different security services, such as authentication, authorization, and confidentiality. This method is used for different data security and privacy protection issues in cloud computing environments. The algorithm AES-128 is used to improve the security and confidentiality of data. In this method, data is encrypted using AES and then uploaded to the cloud. Literature [12] implements parallel execution of AES algorithm in the supercomputer that name is Shen Wei Tai Hu Light. Literature [13] uses biometric generation keys to improve the security of the AES algorithm. The 128 bit key generated by iris feature is combined with the 128 bit key randomly generated to form a 128 bit hybrid key. Although it improves the security of the algorithm, it increases the cost of application and is

difficult to popularize. Literature [14] implements AES-256 encryption and decryption algorithm on hardware. When encrypting, the round-key is generated in parallel to avoid encrypting period, which reduces the delay of each round of encrypting. Literature [15] implements the hardware implementation of AES algorithm on Xilinx Virtex-7 FPGA. The hardware design method is based entirely on pre-computed lookup tables (LUTs), which reduces the complexity of the architecture and provides high throughput and low latency. Literature [16] increases the number of rounds (Nr) of the encryption and decryption process of AES algorithm to 16. Although the system is more secure, the execution efficiency of the algorithm decreases with the increase of the number of rounds. However, none of these improvements involve the management of AES keys. In practical applications, the granularity of database encryption is different, and the number of secret keys produced is also different. Only by ensuring the secure storage of the data secret key can the encrypted data not be cracked.

RSA, an asymmetric encryption algorithm, has high security and is widely used in user data encryption and digital signature [17]. However, the security of this algorithm depends on the decomposition of large numbers, which makes RSA's fastest operation speed much lower than DES algorithm and other symmetric encryption algorithms. Therefore, its low encryption and decryption efficiency is not suitable for encrypting large amounts of data.

The combination of symmetric encryption algorithm and asymmetric encryption algorithm can make up for the shortcomings of each other. Using symmetric encryption algorithm to encrypt and decrypt information can improve the efficiency of encryption and decryption, and using asymmetric encryption algorithm to exchange secret keys. In this way, the secret key security of symmetric encryption algorithm can be guaranteed [18].

In the literature [19], a hybrid application of local file encryption system based on 3DES and RSA algorithm is proposed, which provides better protection for personal privacy and important files. Although this method effectively solves the problem of symmetric secret key management, it is not suitable for encrypting and decrypting large quantities of data because of the low efficiency of the implementation of the 3DES algorithm. In the literature [20], a hybrid DES and SHA-1 encryption system based on VC++ environment is developed to ensure the concealment and integrity of data. On the one hand, the system uses triple DES and RSA algorithm to encrypt and hide data, on the other hand, SHA-1 algorithm is used to verify the integrity of data. In the literature [21], a new hybrid image encryption algorithm based on AES (CS-AES) is developed by using the RNG and S-Box generation algorithms designed. But this algorithm is still a disguised symmetric encryption algorithm based on AES. In the literature [22], combining the characteristics of symmetric encryption and asymmetric encryption, a hybrid encryption system based on RSA and Blowfish algorithm is proposed and implemented on the device of FPGA. In the literature [23], a hybrid algorithm based on DES and RSA is proposed for text data. But this algorithm uses symmetric encryption algorithm DES, although the encryption efficiency is high, but the algorithm security is not high.

Because medical information in the medical information system cloud database needs to be shared between untrusted entities, protecting patient privacy is one of the attributes that medical information systems must satisfy [24]. Aiming at the problem of data security in cloud database of medical information management system, a hybrid encryption algorithm based on AES and RSA is proposed by analyzing the characteristics of cloud database and medical data. This algorithm combines the characteristics of AES encryption algorithm (high encryption and decryption efficiency, high reliability, flexible secret key and grouping) and RSA algorithm (high security, easy implementation) to ensure the security of medical data in cloud database. The AES algorithm is improved to make it more suitable for medical information management system. The algorithm is successfully applied to the medical information management system, and the feasibility of the method is verified.

2.2. AES ALGORITHM

AES algorithm is a block iterative encryption algorithm. Plaintext is divided into several 128 bit plaintext blocks. According to the length of the secret key, it can be divided into three types: AES-128, AES-192 and AES-256, which require 10, 12 and 14 rounds of operation respectively. The lengths of keys corresponding to the three types are 128, 192 and 256 bits. The keys need to be extended to generate 11, 13 and 15 sets of sub-keys. The key corresponding to each round operation is called the round key. It fully combines security, performance, efficiency and flexibility, and can effectively resist existing attack methods.

The encryption process of AES algorithm is to add a 128 bit plaintext block to the byte matrix named State. All operations of the algorithm are performed on State. Taking AES-128 as an example, the number of operation rounds is 10. The operations involved include:

1. **Key Expansion:** generating round-key and cooperating with State to encrypt and decrypt.
2. **Add Round Key:** The State matrix and 128-bit initial secret key output the new State matrix by using the XOR function.
3. **The First Nine Rounds (Round 1-9)**: The first to ninth round perform the following four operations:
    a) SubBytes -each byte in the State is replaced with the substitution table (S-Box). To replace a byte, first convert the byte into two hexadecimal numbers, and then look for the S-Box. These two numbers indicate the rows and columns of the substitution table. In S-Box, the two hexadecimal numbers at the intersection of rows and columns are new bytes.
    b) ShiftRows -each row in the State is cyclically shifted according to a different offset. The unit of offset is byte, the first line remains unchanged, the second line moves one byte to the left, the third line moves two bytes to the left, and the fourth line moves four bytes to the left
    c) MixColumns - this conversion runs on the status column by column. Each byte in the State column is mapped to a new value, which is obtained by a function transformation of 4 bytes in the column.
    d) AddRoundKey -each round of the corresponding round key and State perform XOR function operations.
4. **Final Round (Rounds 10)**: Compared with the first nine rounds, the MixColumms was excluded.

The AES algorithm decryption operation is the inverse of the encryption operation, and the round operations are performed in the reverse order.

3. P-AES ALGORITHM

This paper proposes a P-AES algorithm based on AES-128 algorithm.

3.1. IMPROVEMENT OF AES ALGORITHM ARCHITECTURE

A large amount of medical data is stored in the database of the hospital information system, and the data sizes vary greatly. For example, prescriptions, medical orders, medical records and other information. The data length of the medical order information is relatively small, usually no more than 20 bytes. However, the length of medical record information, such as outpatient electronic medical record home page, outpatient health examination record, inpatient medical record home page, admission record and so on, is tens of thousands of bytes. If these information is encrypted

and decrypted, it will take less time in the case of fewer records. However, when the number of information items exceeds a certain number, it will take a lot of time to process the information. In the actual working environment, the system response time will be too long, which is intolerable for hospital staff.

The architecture of standard AES algorithm is serial structure. In this structure, the State packets that need to be encrypted are introduced serially, and the next packet can be executed only when the current packet processing is completed. This method is suitable for plaintext information with small data length. As described in Section 2.2, each plaintext packet requires 10 rounds operations to complete the encryption operation. With fewer plaintext packets, the computer can quickly encrypt the plaintext. However, some data in the medical database is very large, such as surgical records.

If 20 packets are generated after the plaintext conversion and the serial architecture is still used, 200 rounds of rotation are required. According to Table 1, it takes about 0.26ms for 20 rounds of operation. 200 rounds of operation takes about 2.46ms. In theory, it is 10 times the encryption time compared to the time of 2 packet encryption. Therefore, this paper considers changing the serial structure to a parallel structure, loading 4 State packets at a time, and the round-robin operation is also connected in parallel. Compared with the serial structure, in theory, the serial structure encrypts the time of one packet, and the parallel structure encrypts four packets, and the efficiency of the algorithm will be four times that of the serial structure.

Table 1. Cycle time comparison

Rounds	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Average Time**
20	0.27ms	0.30ms	0.26ms	0.25ms	0.28ms	0.24ms	0.24ms	0.26ms
200	2.5ms	2.48ms	2.43ms	2.82ms	2.46ms	2.26ms	2.26ms	2.46ms

In addition, by looking up the literature, it is determined that the current 128-bit AES algorithm can only implement shortcut attacks on the reduced version of 6 rounds or less, and more than 6 rounds can defend against known shortcut attacks. Through experimental analysis, the number of AES algorithm rounds is set to 7 rounds.

Based on the standard AES algorithm architecture, this paper changes the serial structure to a parallel structure. State plaintext packets that need to be encrypted are introduced in parallel, and round-robin operations are performed in parallel. Assuming that the number of State packets is n, the encryption operation is shown in Figure 1.

Step 1: Plaintext is re-encoded into n State packets;

Step 2: It is judged whether n is ≥ 4, and if it is less than 4, it is sequentially executed in accordance with the standard AES encryption method. Otherwise, go to step 3;

Step 3: The system opens up four threads, and passes the group State1, State2, State3, and State4 as parameters to the Round () method of the thread;

Step 4: Monitor each thread, and if any thread finishes processing, a state group is taken out from the State groups in order and sent to the thread, and arrange the output State packets in order;

Step 5: The encrypted State plaintext packet is re-encoded and the ciphertext is output.

The decryption process of AES is the inverse process of the encryption process, while the decryption of the P-AES algorithm is the inverse operation of the above process. When the algorithm encrypts and decrypts, it can not only allocate redundant threads to affect the efficiency when the state group is small, but also can coordinate the processing of packets by allocating

multiple threads in the case of too many State packets, thereby improving the execution efficiency of the algorithm. Speed up the time of encryption and decryption.

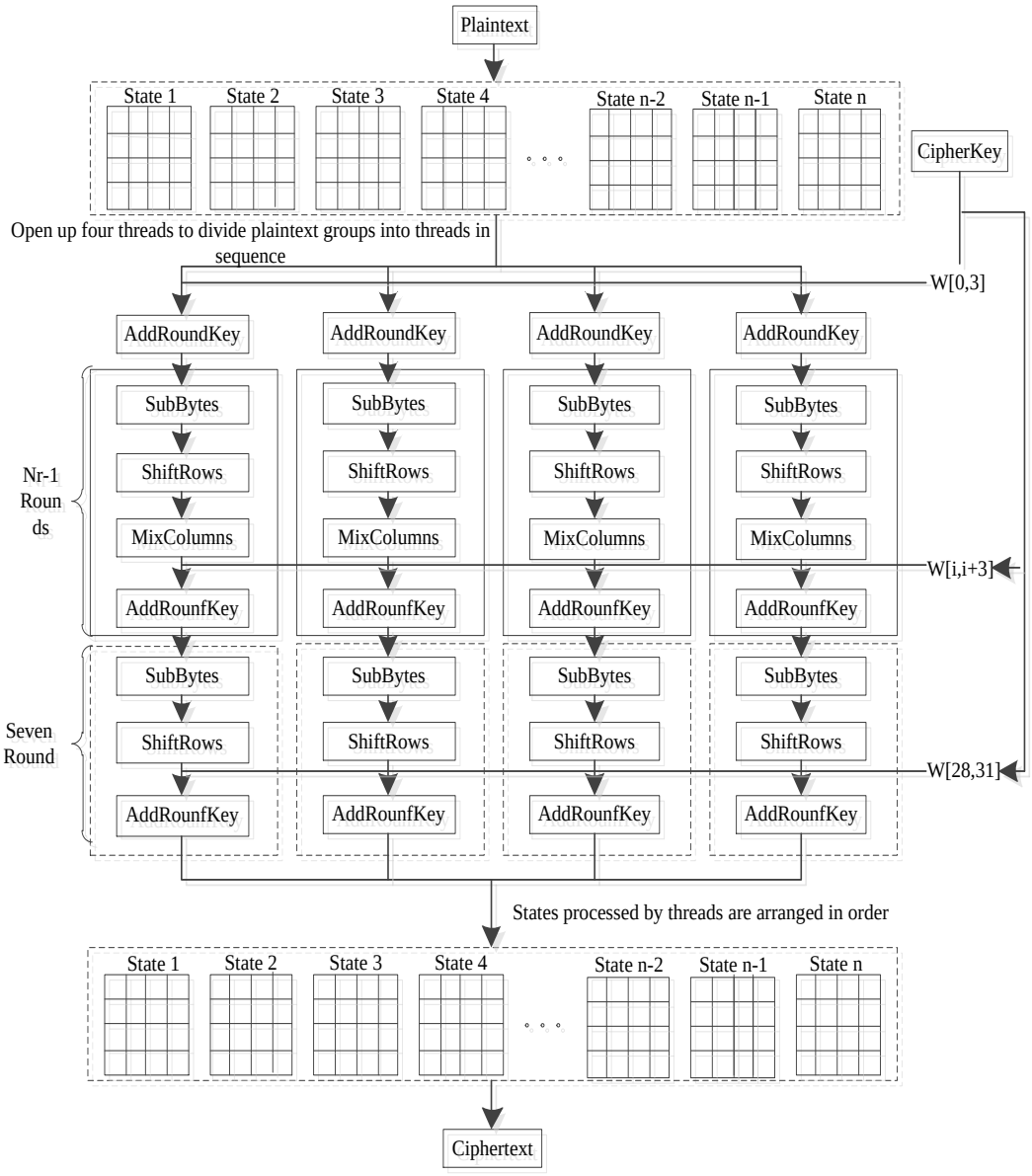

Figure 1. Improved encryption algorithm process

3.2. OPTIMIZATION OF MIXCOLUMNS AND INVMIXCOLUMNS

MixColumns /InvMixColumns is a linear transformation of each column in the State. In the Mix Columns transformation, the column of State is interpreted as a polynomial in the Galois $GF\left(2^8\right)$, multiplication with a given polynomial under modular polynomial x^4+1. If the column input of State is $a(x)$, the output is $b(x)$, we can get $b(x)=a(x)\bullet c(x) mod\left(x^4+1\right)$, where $c(x)$ a reversible polynomial of the modular polynomial is x^4+1. In the standard AES algorithm,

$c(x) = (03)x^3 + (01)x^2 + (01)x + (02)$ in the process of encryption, $c(x) = (0B)x^3 + (0D)x^2 + (09)x + (0E)$ in the process of decryption.

The two transformations are optimized. bi as the i-th element of the Mix Columns transformation output, $b_i^{'}$ as the i-th element of the inverse Mix Columns transformation output, ai is the i-th element of the two transformations,0≤i≤4. The matrix expression (1) of the Mix Columns transformation can be simplified to the expression (2).

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \otimes \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 \cdot (a_0 \oplus a_1) \oplus a_2 \oplus a_3 \oplus a_1 \\ 02 \cdot (a_1 \oplus a_2) \oplus a_0 \oplus a_3 \oplus a_2 \\ 02 \cdot (a_2 \oplus a_3) \oplus a_0 \oplus a_1 \oplus a_3 \\ 02 \cdot (a_0 \oplus a_3) \oplus a_1 \oplus a_2 \oplus a_0 \end{bmatrix} = \begin{bmatrix} 02 \cdot (a_0 \oplus a_1) \\ 02 \cdot (a_1 \oplus a_2) \\ 02 \cdot (a_2 \oplus a_3) \\ 02 \cdot (a_0 \oplus a_3) \end{bmatrix} \oplus \begin{bmatrix} a_2 \oplus a_3 \oplus a_1 \\ a_0 \oplus a_3 \oplus a_2 \\ a_0 \oplus a_1 \oplus a_3 \\ a_1 \oplus a_2 \oplus a_0 \end{bmatrix} \tag{2}$$

The {02} multiplication on Galois $GF(2^8)$ is implemented by the Xtime operation. Because the Xtime operation is very complex, the two transformations take more time. Before optimization, each element needs to be obtained by two times of Xtime multiplications and four times of XOR additions. After optimization, it can be seen from expression (2) that the Mix Columns transformation can reduce the Xtime operation once. This change not only improves the efficiency of the Mix Columns transformation but also reduces the difficulty of code implementation.

The coefficient matrix of the inverse Mix Columns transformation is much more complicated than the Mix Columns transformation. Factorization of the coefficient matrix of the inverse Mix Columns transformation. By factor decomposition, the expression of the inverse Mix Columns transformation finally appears the coefficient matrix of Mix Columns transformation. These two transformations implement code reuse to improve operational efficiency.

The expression (3) represents the inverse Mix Columns transformation:

$$\begin{bmatrix} b_0^{'} \\ b_1^{'} \\ b_2^{'} \\ b_3^{'} \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \otimes \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{3}$$

Observe that the AES decryption algorithm needs to use 0E, 0B, 0D, and 09 constants for Xtime multiplication. The operation process of $b_0^{'}$:

$$\begin{aligned} b_0^{'} &= 0E \cdot a_0 \oplus 0B \cdot a_1 \oplus 0D \cdot a_2 \oplus 09 \cdot a_3 \\ &= 02 \cdot 02 \cdot 02 \cdot (a_0 \oplus a_1 \oplus a_2 \oplus a_3) \oplus 02 \cdot 02 \cdot (a_0 \oplus a_2) \oplus 02 \cdot (a_0 \oplus a_1) \oplus a_1 \oplus a_2 \oplus a_3 \end{aligned} \tag{4}$$

The expression (4) requires six times of Xtime multiplication operations and ten times of XOR addition operations. The computational process is complex and consumes system resources. By factoring, we can get the relationship of the coefficient matrix of the inverse Mix Columns transformation in (5).

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \otimes \begin{bmatrix} 05 & 00 & 04 & 00 \\ 00 & 05 & 00 & 04 \\ 04 & 00 & 05 & 00 \\ 00 & 04 & 00 & 05 \end{bmatrix} \tag{5}$$

Bring (5) to (3), and by the nature of the matrix, get (6):

$$\begin{bmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \otimes \begin{bmatrix} 05 & 00 & 04 & 00 \\ 00 & 05 & 00 & 04 \\ 04 & 00 & 05 & 00 \\ 00 & 04 & 00 & 05 \end{bmatrix} \otimes \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \otimes \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \tag{6}$$

The expression (6) shows that inverse Mix Columns transformation can pre-process the input column in advance. Then work with the coefficient matrix of the Mix Columns transformation to get the result of the inverse Mix Columns transformation. The preprocessing module t resolves to get the expression (7):

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 05 & 00 & 04 & 00 \\ 00 & 05 & 00 & 04 \\ 04 & 00 & 05 & 00 \\ 00 & 04 & 00 & 05 \end{bmatrix} \otimes \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 02 \cdot 02 \cdot (a_0 \oplus a_2) \\ 02 \cdot 02 \cdot (a_1 \oplus a_3) \\ 02 \cdot 02 \cdot (a_0 \oplus a_2) \\ 02 \cdot 02 \cdot (a_1 \oplus a_3) \end{bmatrix} \oplus \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{7}$$

Compared with the previous improvements, only three times of Xtime multiplication and six times of XOR addition are needed.

3.3. IMPROVEMENT OF KEYEXPANSION

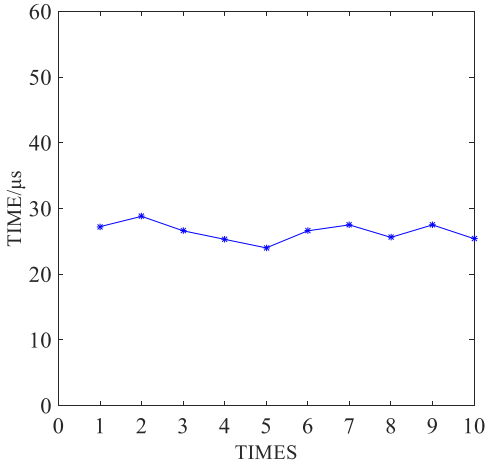

Figure 2. Extending the secret key takes time

Standard AES algorithm, when encrypting and decrypting, must first expand the secret key. Processing n pieces of data, secret key expansion operation performed n times. In the medical information management system, many data tables have storage items of millions or even tens of millions. If the data is encrypted and decrypted, although the same key is used, it needs to perform millions of key expansion operations. This will not only repeat the operation, but also reduce the efficiency of the system. Figure 2 shows the time taken to perform the key expansion operation 10 times, averaging 27.23μs each time. However, when operating a million-level table, it takes about 27 seconds to process millions of data.

Combined with the actual application scenario, this paper made an improvement to the AES algorithm. The field using the same key is only executed once the secret key expansion operation during encryption and decryption. The standard AES algorithm performs secret key expansion operations inside the algorithm. In this paper, key extension operation is implemented outside AES algorithm. Operation steps:

Step 1: Perform key extension based on the initial key to get the extended key;

Step 2: Extended secret key is passed into AES encryption algorithm or decryption algorithm as a parameter;

Step 3: Get ciphertext or plaintext data.

In this method, the system only needs to perform a secret key expansion operation on the first piece of data, and the result can be applied to the encryption and decryption operations of other data.

The algorithm is designed for the characteristics of medical data in the database. In a cloud computing environment, the P-AES algorithm is used to encrypt data in a cloud database. Because of the parallel architecture, it is more suitable for processing data with longer data length than AES-128. In the cloud, data is shared and data queries are large. By changing the architecture of AES-128 algorithm, optimizing the operation process of column mixing and inverse column hybrid transform and changing the way of secret key expansion, the purpose of improving the encryption and decryption efficiency in cloud data is achieved. However, the P-AES algorithm has certain limitations and can only be used to encrypt and decrypt text data. In addition, only 128-bit key can be used to encrypt and decrypt data. Code connection address: https://gitee.com/XuWoYiShiXiaoYan/P-AES.git.

4. THE HYBRID P-AES AND RSA ENCRYPTION ALGORITHM

4.1. THE HYBRID ENCRYPTION ALGORITHM

As described in Section 2.1, the AES algorithm is suitable for encrypting and decrypting large amounts of data. However, the characteristics of symmetric encryption algorithms (encryption and decryption use the same key), resulting in AES algorithm key management is very inefficient. Although the RSA algorithm is not suitable for encrypting and decrypting large amounts of data, it is more convenient to manage keys. The algorithm is highly secure.

In order to ensure the confidentiality of data in the hospital cloud database, this paper proposes a hybrid method combining RSA algorithm and P- AES algorithm. The P-AES algorithm is used to encrypt and decrypt the attribute values of the field, and the P-AES key is protected by the RSA algorithm. In the system, distribute a pair of keys (public and private) for each system user. Use KU1, KU2, ... , KUn to represent the public keys of User1, User2, ... , Usern; The private keys of User1, User2, ... , Usern are represented by KR1, KR2, ... , KRn. Different system users have unused roles, and the corresponding roles have different permissions. Each role has different operational permissions on the encrypted field.

4.2. DATA ENCRYPTION PROCESS

In the cloud computing environment, the sensitive data stored in the cloud data of the hospital is encrypted by using the P-AES algorithm, and becomes an illegible form that cannot be read. In the case where data is obtained by hackers or cloud service providers using illegal means, it does not reveal patient privacy. At the same time, the RSA algorithm is used to protect the secret key of the P-AES algorithm and securely manage the key. If Userl, User2, ... , Usern correspond to the same role, the role has the operation authority for the table M. The plaintext q is an attribute value of the encrypted field Mi ($i \leq n$) in the table M. K represents the secret key for encrypting Mi using the P-AES encryption algorithm. The user Useri ($i \leq n$) encrypts the plaintext q. KUl, KU2, ... , KUn are the public keys of Userl, User2, ... , Usern. The encryption process for plaintext q and the protection process of secret key K are described as follows:

Step 1: There are records in Table M, indicates an encrypted record of the key K in which the Mi exists in the data key table. Execute Step 2- Step 4 if it exists, otherwise execute Step 3- Step 6.

Step 2: Useri decrypts RSAEncrypt (K, KUi) using its own private key KRi to obtain the encrypted key K of plaintext q.

Step 3: K is the encryption key of the plaintext q. The plaintext q is encrypted by using the P-AES encryption algorithm to obtain the ciphertext form AESEncrypt (q, K) of q.

Step 4: AESEncrypt (q,K) is stored in the table M as the attribute value of Mi.

Step 5: Encrypt the data key K using the RSA algorithm, the key is Useri's public key KUi, and obtain the ciphertext form RSAEncrypt(K, KUi) of the data key K. Record RSAEncrypt(K, KUi) into the data key table of the database.

Step 6: all users under the corresponding role of Useri use their own public key to encrypt and store the data key. For example, the ciphertext form of the data key K corresponding to User2 is RSAEncrypt (K, KU2).

4.3. DATA DECRYPTION PROCESS

In a cloud environment, when data in a cloud database is encrypted using P-AES, if you want to invoke information, you must first decrypt the information. First, the key of the P-AES is decrypted by the private key of the cloud database user. The decrypted key and the P-AES decryption algorithm are used to decrypt the data. The decryption process of the ciphertext state of plaintext q is as follows:

Step 1: Useri decrypts RSAEncrypt (K, KUi) using its own private key KRi to obtain the encrypted key K of plaintext q.

Step 2: Take out the attribute value AESEncrypt (q, K) of the Mi field in the table M, and use the data secret key K to decrypt it by calling the decryption algorithm AESDecrypt ( ) to obtain the plaintext q.

4.4. USER PRIVATE KEY PROTECTION

This article uses the user's password to protect the user's private key. Take User Useri as an example. The encryption process is as follows:

Step 1: The user Useri password is encrypted by the MD5 algorithm to obtain the MD5 encrypted value S of the password;

Step 2: Using S as the encryption key, use the P-AES encryption algorithm to obtain the Useri encrypted private key AESEncrypt (KRi, S) and store it in the database.

The process of decrypting the private key is as follows:

Step 1: The user Useri password is encrypted by the MD5 algorithm to obtain the MD5 encrypted value S of the password;

Step 2: Using S as the encryption key, the P-AES decryption algorithm is used to obtain Useri's private key AESDecrypt (AESEncrypt (KRi, S), S).

5. EXPERIMENTAL RESULTS AND SYSTEM IMPLEMENTATION

5.1. SYSTEM IMPLEMENTATION

The successful application of this research in medical information system effectively protects the security of medical data and thus protects patient privacy. Figure 3 is a screenshot of the patient prescription query interface. The operating environment of the system is: processor Intel Core i5-4460 3.20GHz, 8GB of memory, 1TB of hard disk, Windows 10 x64 operating system, programming language is C#. In the system, the staff sees the plaintext information, and the related fields in the database are ciphertext storage. As shown in Figure 4, the blue mark record represents the information queried in the PRESCRIPTION of Figure 3. The encrypted information of the field sZD corresponds to the information in Figure 3 Diagnose.

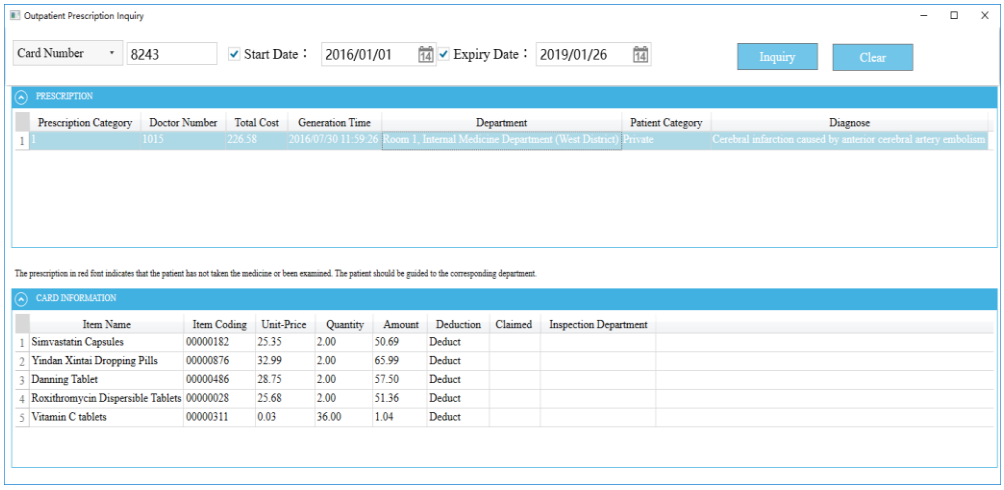

Figure 3. Patient prescription record query

nDDtrNO	dDMDate	sDBID	sDStatus	nCardId	nYQId	nTotalPrice	sZD
94	2016-07-30 11:33:43	(Null)	3	8228	2	68.4570000	2tWSoZ+OlDgPeksyyc4JdFQYTS2auxLiJl9xRNPTBz8=
94	2016-07-30 11:40:29	3614	0	8239	(Null)	25.0000000	4baHj54ZN0S9gacuPBsee0wkF5GgOBl22K7Teth064B=
1015	2016-07-30 11:48:01	2266	0	8242	(Null)	220.0000000	v3+YlbB049u0ipqE4FexyXRzjyUXz1sSaPF5znd9DQ=
1015	2016-07-30 11:59:26	(Null)	3	8243	2	226.5800000	2a80f803c2b3A9aCD57f799f6e525FQ387a5tJe16H95QWe774+d16/321a==334B47dDcc1Ve9da3Pd1X928d24f3/3A04NLS5fa0WGJa45e1/4cb35==4481cd2e03
1015	2016-07-30 12:04:14	(Null)	3	8245	2	50.6920000	cmlNjY1ZHnJL2lg1wQ/1SZUvOOOkgeHcNXkCpIBXhsQ=
102	2016-07-30 12:21:14	(Null)	0	8230	2	52.3940000	zQ1cSUKdMynhP+3vEK7psg==
102	2016-07-30 12:21:14	2268	0	8230	(Null)	60.0000000	HLd3B60AFHUwxMZm/lusmRbzmlABTFgJU6t3SDxZ26U=
15	2016-07-30 14:40:45	2270	0	8247	(Null)	80.0000000	3tsj01zgFP93O0V3VoU3Y+MXeo96m7+97wYuY4JKh6s=
29	2016-07-30 14:55:52	2271	0	8246	(Null)	60.0000000	lyKDhTKZngTL3hL+ro5NOUEmevk6ZclbN17LK5UNQCs=
15	2016-07-30 14:57:36	3622	1	8247	(Null)	18.0000000	xA45GmmE+NKGZvBaHN7i3A==
15	2016-07-30 15:09:34	(Null)	0	8247	2	109.2500000	xA45GmmE+NKGZvBaHN7i3A==
29	2016-07-30 15:14:58	2272	0	8248	(Null)	390.0000000	1TfahOw/nkoN9wlY6GrczjY6DkGx/fXCKRU0ruCP0Pw=
94	2016-07-30 15:18:08	2273	0	8249	(Null)	80.0000000	OQSm6K5E4KH0d+wKFtqa36owWVou46RDw/A2Z6fySY=
94	2016-07-30 15:19:23	2274	4	8249	(Null)	29.0000000	OQSm6K5E4KH0d+wKFtqa36owWVou46RDw/A2Z6fySY=
94	2016-07-30 15:19:23	2275	4	8249	(Null)	7.0000000	OQSm6K5E4KH0d+wKFtqa36owWVou46RDw/A2Z6fySY=
94	2016-07-30 15:20:18	3623	1	8249	(Null)	18.0000000	OQSm6K5E4KH0d+wKFtqa30j9b1sC3ky4BBch04T4tA0=
94	2016-07-30 15:21:22	3624	1	8249	(Null)	6.0000000	OQSm6K5E4KH0d+wKFtqa30j9b1sC3ky4BBch04T4tA0=
94	2016-07-30 15:23:21	2276	0	8249	(Null)	42.0000000	OQSm6K5E4KH0d+wKFtqa36owWVou46RDw/A2Z6fySY=
29	2016-07-30 15:24:59	2277	0	8246	(Null)	60.0000000	lyKDhTKZngTL3hL+ro5NOUEmevk6ZclbN17LK5UNQCs=

Figure 4. Prescription table information in the database

5.2. EXPERIMENTAL RESULTS

This paper uses the medical data in the database of the hospital information management system to carry out relevant experiments.

5.2.1. FEASIBILITY ANALYSIS OF ENCRYPTION ALGORITHM AND ENCRYPTION LEVEL

The drug name "Tianmasu Injection" was continuously encrypted and decrypted 10 times using four different algorithms to obtain an average time. On this basis, repeat 10 operations. Figure 5 shows, the four algorithm encryption takes between 30ms and 40ms. The 3DES algorithm is obviously higher than the other three algorithms. However, the encryption of the AES algorithm takes less time than the Blowfish algorithm and the DES algorithm. Figure 6 shows, the AES algorithm takes the least amount of decryption time. So choose AES algorithm as the data encryption and decryption algorithm.

In Figure 7 and Figure 8, AES1 indicates the AES algorithm using the MySQL database without external coding. AES2 is implemented by external encoding, but uses the AES algorithm of the MySQL database. AES3 represents an implementation using an externally encoded AES algorithm.

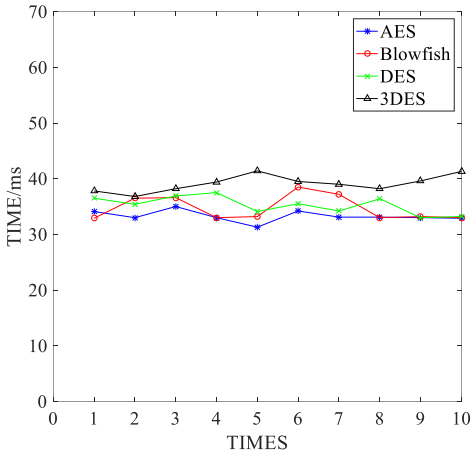

Figure 5. Encryption time comparison

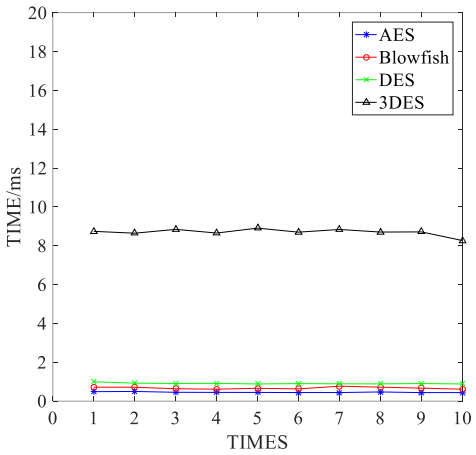

Figure 6. Decryption time comparison

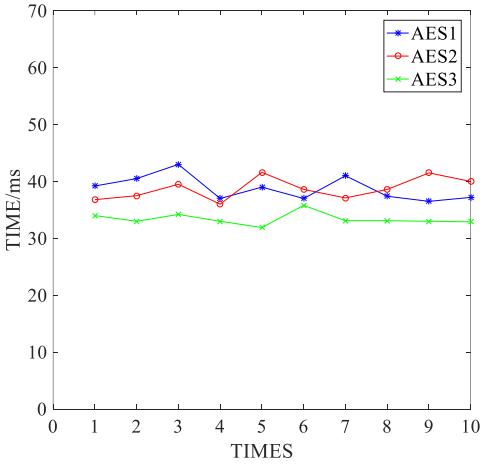

Figure 7. AES encryption time comparison

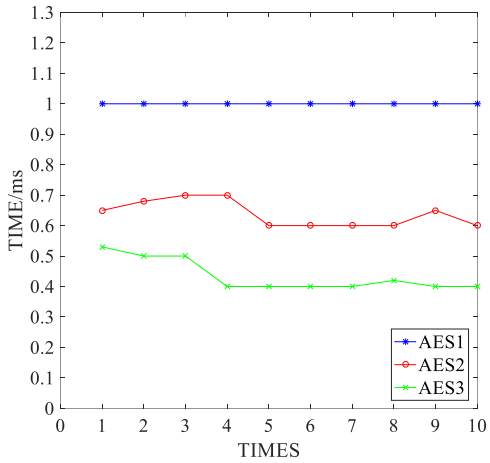

Figure 8. AES decryption time comparison

In the encryption and decryption experiment, the drug name "Tianmasu injection" was continuously encrypted and decrypted 10 times, and the average time was obtained. On this basis, repeat 10 operations. Figure 7 shows, AES3 takes between 30ms and 35ms when encrypting, and AES1 and AES2 encryption takes between 35ms and 45ms. Comparison of AES3 is better than AES1 and AES2. In the decryption experiment, because the time viewed in Navicat for MySQL can only be accurate to milliseconds, the decryption time is about 1ms, so the decryption time of AES1 is 1ms. Figure 8 clearly shows that AES3 is much less time consuming than AES1 and AES2. The AES algorithm is implemented by external coding, and the data is encrypted and decrypted by this algorithm.

5.2.2. IMPROVEMENT BEFORE AND AFTER ENCRYPTION AND DECRYPTION TEST COMPARISON

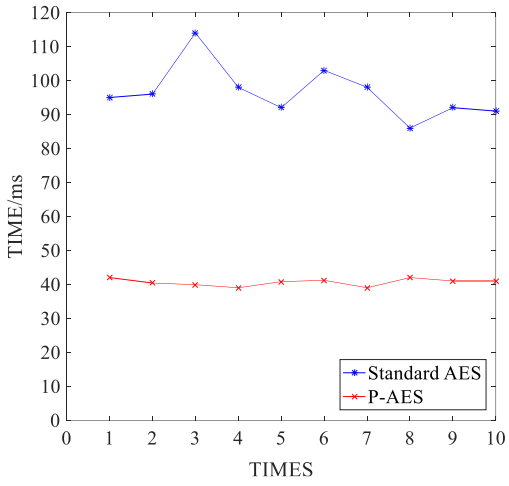

Figure 9. Short data encryption comparison

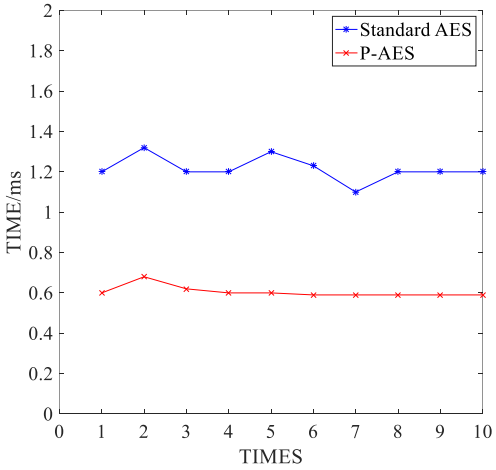

Figure 10. Short data decryption comparison

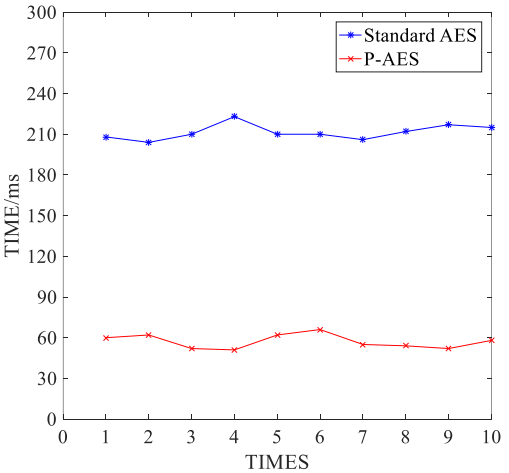

Figure 11. Long data encryption comparison

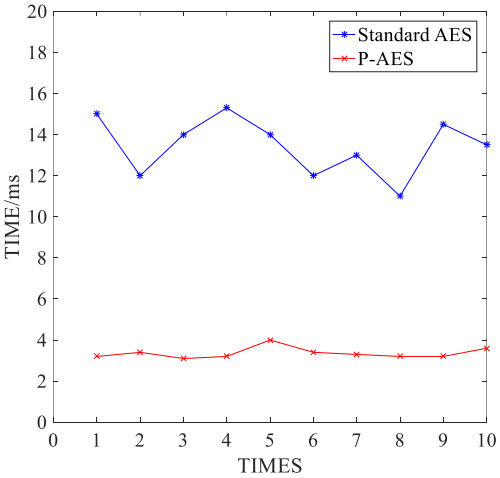

Figure 12. Long data decryption comparison

The test selects the standard AES algorithm and compares it with the P-AES algorithm. The algorithm is applied to the hospital information system management system. Experiment with short data and long data separately. Figure 9 and Figure 10 show the encryption and decryption test of short data (e.g., drug name) in the database. The encryption and decryption of a drug name is continued 10 times, and the average time is obtained. On this basis, repeat 10 operations. Figure 11 and Figure 12 show the encryption and decryption test on the long data (e.g., the

surgical anesthesia record) in the database, and the data length is about 190 KB. The anatomical record of a certain operation was continuously encrypted and decrypted 10 times, and the average time was obtained. On this basis, repeat 10 operations. It can be obtained that the P-AES algorithm is superior to the standard AES algorithm. The P-AES algorithm takes less time and is more stable. Especially when dealing with longer data, the advantages are more obvious.

5.2.3. Confusion and Diffusion Test

In the algorithm of this paper, the number of rounds of encryption is reduced, and it is necessary to verify whether it affects confusion and diffusivity through experiments. Test the diffusivity and encrypt 128 bit strings. In the same key, the plaintext changes one bit at a time. Observe the number of bits in the ciphertext change. Figure 13 shows that the standard algorithm ciphertext change digits range from 66 ± 7, and the P-AES algorithm ciphertext change digits range from 66 ± 7. The confusing test, the plaintext is unchanged, and the key is changed one at a time. Observe the impact on the ciphertext. Figure 14 shows that the standard algorithm ciphertext change digits range from 65 ± 6, and the P-AES algorithm ciphertext change digits range from 64 ± 4. After testing, the number of cipher text changes is around 66. It shows that the P-AES algorithm does not affect the diffusion and confusion of the algorithm.

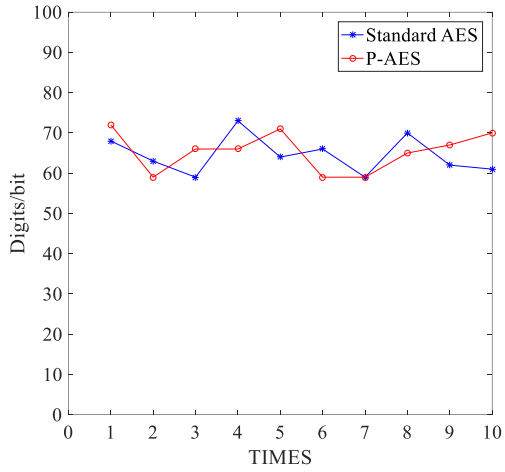

Figure 13. Diffusivity contrast

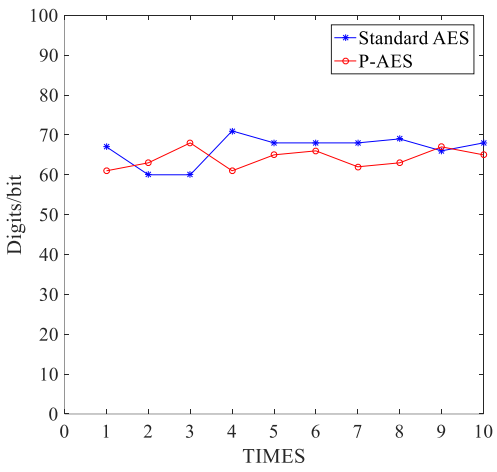

Figure 14. Confusing contrast

5.2.4. The Effect of Hybrid Algorithm on Data Query Efficiency

Whether the hybrid encryption algorithm is suitable for medical management system needs to be compared with that in the case of unencrypted. Time-consuming trial of data query based on the medical order table of the hospital information management system. The table has 1.4 million rows of data, the size of the encrypted table is 298MB, and the size of the unencrypted table is 216MB. Compare the query time of both encrypted and unencrypted methods under the same query conditions. As seen from Figure 15, the encryption has a certain impact on the time of data query. However, it is not much different from the time spent unencrypted.

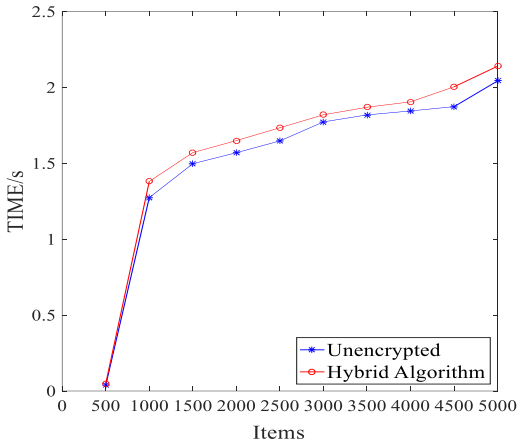

Figure 15. Comparison of average query time of projects with different data volumes

6. CONCLUSION

The data in this paper is based on medical data from Yongcheng Chinese Medicine Hospital. This paper verifies that the AES algorithm and the database outer layer encryption method have higher encryption and decryption efficiency through relevant experiments. According to the characteristics of medical data, the P-AES algorithm is proposed. It has been proved by relevant experiments that it has higher encryption and decryption efficiency than the original algorithm and is more suitable for processing long data. Finally, based on this, combined with the RSA algorithm, a hybrid encryption algorithm is proposed and successfully applied in the medical information management system. Through query comparison, although data encryption will affect the efficiency of the query, it is within the user's acceptance. Therefore, the hybrid encryption algorithm can provide security protection for medical data stored in cloud database and protect patients' privacy.

However, the P-AES algorithm also has certain limitations, and currently only text data can be encrypted. Cannot be used to encrypt data such as pictures and images. Moreover, the algorithm is based on AES-128 and can only encrypt data using a 128-bit key. There is still much room for improvement in the P-AES algorithm.

ACKNOWLEDGEMENTS

Thanks to the laboratory for providing the environment for my research practice. Thanks to my mentor and other members of the lab for their help in my paper research. Thanks to Yongcheng Central Hospital for supporting our work. It was because of their help that this article was successfully completed.

REFERENCES

[1] Wang Cong, Wang Qian & Ren Kui, (2011) “Ensuring data storage security in Cloud Computing”, IEEE International Conference on Parallel Distributed & Grid Computing.

[2] Salma & T. J., (2013) “A flexible distributed storage integrity auditing mechanism in Cloud Computing”, IEEE International Conference on Information Communication & Embedded Systems.

[3] S.Suganya, (2014) “Enhancing security for storage services in cloud computing”, IEEE Current Trends in Engineering and Technology, Vol. 3, No. 6, pp283-287.

[4] Kaufman & L. M., (2009) “Data Security in the world of cloud computing”, IEEE Security & Privacy, Vol. 4, No. 7, pp61-64.

[5] Takabi H, Joshi J. B. D. & Ahn G, (2010) "Security and privacy challenges in cloud computing Environment", IEEE Security & Privacy, Vol. 6, No. 8, pp24-31.

[6] Poh, G. S., Chin, J. J. & Yau, W. C., (2017) "Searchable symmetric encryption", ACM Computing Surveys, Vol. 50, No. 3, pp1-37.

[7] Qiu Weixing, Xiao Kezhi & Li Fang, (2011) "A kind of method of extension of the DES key," Computer Engineering, Vol. 5, No. 37, pp167-168.

[8] T. Good & M. Benaissa, (2007) "Pipelined AES on FPGA with support for feedback modes (in a multi-channel environment)", IET INFORMATION SECURITY, Vol. 1, No. 1, pp1-10.

[9] Mondal, S. & Maitra, S., (2014) "Data security-modified aes algorithm and its applications", ACM SIGARCH Computer Architecture News, Vol. 2, No. 42, pp1-8.

[10] Elbadawy & A. M., (2010) "A new chaos Advanced Encryption Standard (AES) algorithm for data security", IEEE International Conference on Signals & Electronic Systems, pp7-10.

[11] Babitha M. P. & Babu K. R. R., (2017) "Secure cloud storage using AES encryption", IEEE International Conference on Automatic Control & Dynamic Optimization Techniques.

[12] Chen, Y, & Li K, (2017) "Implementation and Optimization of AES Algorithm on the Sunway TaihuLight", IEEE International Conference on Parallel & Distributed Computing. Pp256-261.

[13] Priya, S. S. S., & Karthigaikumar, P., (2015) "Generation of 128-Bit Blended Key for AES Algorithm", CSI Emerging ICT for Bridging the Future , Vol. 2, No. 49, pp431-439.

[14] Mohurle, M., & V. V. Panchbhai, (2017) "Review on realization of AES encryption and decryption with power and area optimization", IEEE International Conference on Power Electronics, Vol. 2.

[15] N. S. Sai Srinivas & Monhammed Akramuddin, (2016) "FPGA Based Hardware Implementation of AES Rijndael Algorithm for Encryption and Decryption", IEEE International Conference on Electrical, Electronics and Optimization Techniques (ICEEOT-2016).

[16] Puneet Kumar & Shashi B. Rana, (2016) "Development of modified aes algorithm for data security", Optik - International Journal for Light and Electron Optics, Vol. 4, No. 127, pp2341-2345.

[17] Rivest, R. L., Shamir, A., & Adleman, L., (1978) "A method for obtaining digital signatures and public-key cryptosystems", Communications of the Acm, Vol. 2, No. 21, pp120-126.

[18] Anane Nadjia & Anane Mohamed, (2015) "AES IP for hybrid cryptosystem RSA-AES", IEEE International Multi-conference on Systems. No. 12.

[19] HU Zhen, (2012) "Triple DES and RSA-based file encryption system", Computer and Modernization, No. 9, pp101-105.

[20] Jian Zhang & Xuling Jin, (2012) "Encryption System Design Based on DES and SHA-1", IEEE International Symposium on Distributed Computing & Applications to Business, pp317-320.

[21] Ünal Çavuşoğlu & Sezgin Kaçar, (2018) "A novel hybrid encryption algorithm based on chaos and S-AES algorithm", Nonlinear Dynamics, Vol. 92, No. 4, pp1745-1759.

[22] Viney Pal Bansal & Sandeep Singh, (2016) "A hybrid data encryption technique using RSA and Blowfish for cloud computing on FPGAs", IEEE International Conference on Recent Advances in Engineering & Computational Sciences.

[23] Smita Chourasia & Kedar Nath Singh, (2016) "An Efficient Hybrid Encryption Technique Based on DES and RSA for Textual Data", Information Systems Design and Intelligent Applications, pp73-80.

[24] Anand Sharma & Vibha Ojha, (2010) "Implementation Of Cryptography For Privacy Preserving Data Mining", International Journal of Database Management Systems, Vol. 2, No. 3,pp57-65.

AUTHORS

Fenghua Zhang is a student at Henan University of Science and Technology, Master's degree.

Yanming Chen is a student at Henan University of Science and Technology, Master's degree.

Weiming Meng is a student at Henan University of Science and Technology, Master's degree.