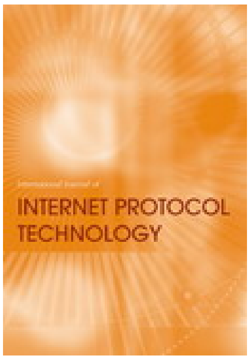
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Internet Protocol Technology

ISSN online: 1743-8217 - ISSN print: 1743-8209
https://www.inderscience.com/ijipt

A cross encryption scheme for data security storage in cloud computing environment

Haiyan Kang, Jie Deng

DOI: 10.1504/IJIPT.2023.10054903

Article History:

Received:	11 February 2021
Last revised:	13 June 2021
Accepted:	13 June 2021
Published online:	23 March 2023

A cross encryption scheme for data security storage in cloud computing environment

Haiyan Kang*

School of Information Management,
Beijing Information Science and Technology University,
Beijing 100192, China
Email: kanghaiyan@126.com
*Corresponding author

Jie Deng

Computer School,
Beijing Information Science and Technology University,
Beijing 100192, China
Email: bistu_dengjie@163.com

Abstract: Cloud computing is one of the popular technologies in the development of information technology. Cloud computing not only provides users with high-performance computing, but also meets the needs of large-scale data storage. However, because the storage service provided by cloud computing is completely transparent to users, users cannot understand whether their data is safe in the cloud computing environment. The resulting distrust has brought great obstacles to the development of cloud computing. Therefore, this paper first describes the basic knowledge and system architecture of cloud storage, and analyses the development status of cloud storage. Secondly, in order to ensure the storage security of user data in the cloud computing environment, this paper studies the data encryption algorithm, and proposes a cross encryption scheme of data security storage in the cloud computing environment. Finally, the scheme is compared with the traditional hybrid encryption method. The experimental results show that the scheme has the advantages of good encryption and decryption effect, fast execution speed and high security. It is an ideal scheme for data security storage in cloud computing environment.

Keywords: cloud computing; data encryption; DES; RSA; cross encryption.

Reference to this paper should be made as follows: Kang, H. and Deng, J. (2023) 'A cross encryption scheme for data security storage in cloud computing environment', *Int. J. Internet Protocol Technology*, Vol. 16, No. 1, pp.1–10.

Biographical notes: Haiyan Kang is a senior member of the China Computer Federation (No. E200028533M), ACM Membership (No.9495204), and member of privacy protection committee of China Confidentiality Association. He received his PhD in Computer Application Technology from Beijing Institute of Technology, China in 2005. His research interest fields include information system security, privacy preserving, and natural language processing (NLP). He is currently working as a professor at Department of Information Security, School of Information and Management, Beijing Information Science and Technology University, Beijing, China.

Jie Deng received a bachelor's degree in Internet of Things Engineering from Xingtai University. She is currently studying for a master's degree in computer technology at Beijing Information Science and Technology University. Her research interests include privacy protection in the field of blockchain and data security encryption.

This paper is a revised and expanded version of a paper entitled 'Hybrid Encryption Scheme for Secure Storage of Smart Grid Data' presented at 'Smart Grid and Internet of Things - 4th EAI International Conference', TaiChung, Taiwan PRC, 5–6 December 2020.

1 Introduction

In recent years, with the continuous development of Internet of Things, mobile computing and big data storage, more and more large internet companies and scientific research institutions have made more in-depth research on cloud computing. Cloud computing, with its own characteristics of ultra fast computing capacity, super large storage capacity and on-demand provision, is also considered as the framework of the core technology of the next generation computer network (Hoefer and Karagiannis, 2011; Qiu et al., 2013; Qiang, 2019). When massive data is put into the cloud computing environment, more and more people will pay attention to its data storage security (Feng et al., 2015; Li et al., 2017). Because the storage service provided by cloud computing is completely transparent to users, the general users cannot understand the specific situation of their data stored in the cloud, resulting in a sense of distrust, which has brought great obstacles to the development of cloud computing. Therefore, the research of data security storage based on cloud computing is of great significance. At present, the primary problem of cloud storage is to take effective measures to protect the security of users' privacy data. Because the user's information processing, analysis and storage are not in the local, but in the cloud, it is necessary to use high-intensity means to encrypt the information, so as to ensure that the information is safe even when it is stolen. It is a common scheme to protect the files and data stored in the cloud by storage encryption. However, the traditional encryption methods have some defects. For example, DES (Tuchman, 1979), 3DES (Gao et al., 2006) and AES (Jain et al., 2019), which belong to symmetric encryption, have the same secret key as the decryption key. In the cloud environment, the difficulty of secret key management is greatly increased. However, RSA (Rivest et al., 1978), ECC (Seroussi, 1999) and DSA (Wang et al., 2003), which belong to asymmetric encryption, take a long time to encrypt and decrypt, and cannot meet the requirements of efficient large-scale data storage. At the same time, because the cloud platform has the characteristics of massive data and files, it is difficult for traditional encryption methods to make full use of the advantages brought by the cloud platform resources to complete the encryption efficiently. Therefore, in order to adapt to the characteristics of the cloud environment, an efficient, reliable and easy to implement encryption scheme is needed to encrypt the data and files stored in the cloud platform. In this regard, a lot of research has been done on data encryption methods based on cloud computing at home and abroad, and various data storage encryption schemes in cloud environment have been proposed. For example, Sun (2013) proposed an improved DES algorithm. Without changing the time complexity of the traditional DES algorithm, the algorithm adopts the strategy of lengthening the key length to improve its security, and better solves the security problem of open source cloud computing platform Hadoop. Yellamma et al. (2014) proposed a RSA public key cryptosystem to protect the security of data stored in the cloud. Taking advantage of the feature that RSA algorithm can only use brute force cracking at present, two 1024 bit large prime numbers are used as encryption key prime numbers to ensure the security of data. However, in order to prevent brute force cracking and improve security, RSA will continue to increase the number of large primes used to produce secret key calculation, which will lead to the decline of encryption efficiency, especially for large file encryption. Due to various shortcomings of a single encryption algorithm, hybrid encryption algorithm has become the key research direction of cloud computing data security in recent years. For example, Mahalle and Shahade (2014) and Khanezaei and Hanapi (2015) proposed a hybrid encryption method of symmetric encryption and asymmetric encryption to protect cloud platform data, which has better encryption efficiency while ensuring encryption security. Kanna and Vasudevan (2016) proposed a new identity based hybrid encryption (RSA with ECC) to enhance the security of outsourced data.

In this method, the sender uses a hybrid algorithm to encrypt sensitive data, and then the agent reencrypts the key words and identity to enhance the security of encrypted data. Zhao (2019) proposed a parallel AES and RSA hybrid encryption algorithm improved by MapReduce in cloud computing environment. The algorithm uses RSA algorithm to manage the key and AES algorithm to encrypt the plaintext data, which improves the encryption efficiency in the cloud computing environment as a whole. Li et al. (2019) designs a corresponding protocol for the new service model based on hybrid encryption algorithms, which combines AES symmetric encryption and Paillier homomorphic encryption. And analysis results show the scheme can not only protect both the location privacy of the user and the data privacy of the data service provider, but also can ensure good performances, that is, it can reduce the computation, communication and storage costs of the date service provider and the query user.

Based on the data storage service model in cloud computing environment and the encryption algorithm of data encryption technology, this paper proposes a cross encryption scheme of data security storage in cloud computing environment. Based on the traditional DES and RSA algorithm, this paper first analyses the advantages and disadvantages of DES, and combines the advantages of Triple Data Encryption Algorithm (TDEA), improves DES algorithm, and proposes a N-DES encryption algorithm. Then, this paper makes a detailed study on the method of judging prime number which affects the operation speed of RSA algorithm. On the basis of not affecting the security of RSA, this paper improves the original method of prime number judgment, and proposes a RSA Algorithm Based on Optimised Prime Number Judgement (RSA_OPJ). Finally, this paper combines the N-DES encryption algorithm and RSA_OPJ encryption algorithm to form a cross encryption scheme based on N-DES and RSA_OPJ, which can effectively ensure the security of user data in the cloud.

Section 1 introduces the background of cloud computing and the research status of data security protection of cloud storage at home and abroad. Section 2 introduces the basic theory and security analysis of cloud computing. Section 3 introduces the algorithm design, performance comparison experiment and experimental results analysis of the cross encryption scheme for data security storage. A discussion of future work and summarisation are presented in Section 4.

2 Cloud computing related basic theory and security analysis

2.1 *Basic concepts of cloud computing*

2.1.1 *Cloud computing technology*

Cloud computing (Feng et al., 2011) is a combination of traditional computing modes such as distributed computing, utility computing, grid storage and virtualisation, and the development of network technology. It is a new network service mode. Cloud computing includes virtualisation technology, data storage technology, data management technology and other key technologies with their own unique characteristics. Cloud computing has four deployment models (Yi et al., 2012): private cloud, public cloud, community cloud, and hybrid cloud.

2.1.2 Cloud storage technology

Cloud storage (Li, 2010) is a data access service based on cloud computing, which can be easily understood as a cloud computing system with super large storage space. It integrates a large number of different types of storage devices in the network through the corresponding software to work together to provide data storage and access services for users. Its core is data storage and management. With the development of cloud storage services, more and more enterprises and individuals enjoy its efficient, fast, low-cost services, but its security issues also deserve people's attention. For the end users who use cloud storage, cloud storage is not only a hardware, but a huge system composed of storage devices, network devices, application software, servers, terminal customers and so on. Generally speaking, the system structure of cloud storage consists of storage layer, basic management layer, application interface layer and access layer, as shown in Figure 1.

1 *Storage layer.* The most basic part of cloud storage. It can contain a variety of types of devices, which are distributed in different areas and connected together through the network, and can realise the centralised management and status monitoring of massive data.

2 *Basic management layer.* The most core and complex part of cloud storage. Using distributed storage technology and integrated management technology, not only can realise the collaborative work among multiple devices, provide efficient data access performance, but also undertake the tasks of data encryption, disaster recovery, backup, etc., ensuring the stability and security of cloud storage itself.

3 *Application interface layer.* The most flexible part of cloud storage. Different cloud storage providers develop different service interfaces according to business requirements to provide diversified services to end users, such as video monitoring application platform, network hard disk application platform, etc.

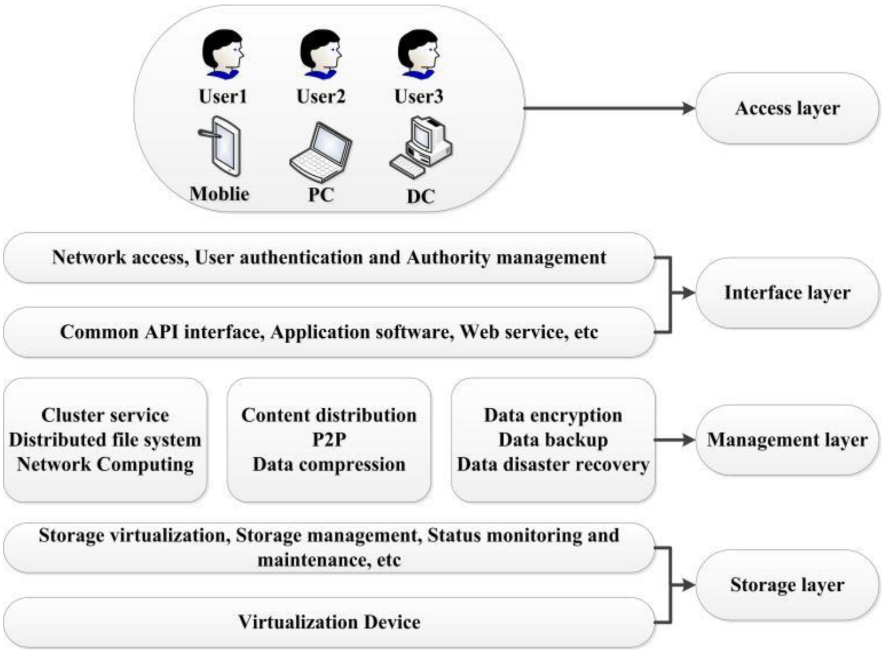
4 *Access layer.* Any legitimate user can log in to the cloud service platform through the standard application interface to share the services provided by cloud storage.

2.2 Data security storage analysis of cloud computing

While cloud computing brings great convenience to people, the centralised storage of data may endanger the data security of users. In the cloud computing services that have been implemented, the problem of data security has been worrying, so that it has become a huge challenge in the popularisation of cloud computing.

Research shows that most users are mainly concerned about the security of cloud computing. If enterprises want to use cloud computing services to reduce the cost and complexity of internet technology, they need to ensure that this process will not bring any potential data security problems. Thus, security is one of the most important obstacles to the development of cloud computing (Shi et al., 2012).

Figure 1 Cloud storage system structure



In the cloud computing environment, data storage and operation are provided in the form of services, which has inherent characteristics in data security. First of all, the storage and security of user data is entirely the responsibility of cloud computing providers. Therefore, digital data is transparent to providers. Secondly, in the cloud computing mode, the user's data is stored in the internet server, which increases the security problem of data transmission.

At present, cloud computing is based on the existing distributed network. Every computer on the network can be considered as a node. If there is no reliable security protection, in theory, every node can access other nodes through certain means. The storage security of Cloud Computing mainly involves data transmission, isolation, recovery, long-term survival and so on.

3 Cross encryption scheme for data security storage in cloud computing environment

3.1 Analysis of data security storage in cloud computing environment

The data security storage service in cloud computing environment needs to consider efficiency while ensuring data security. Therefore, the design of cloud computing secure storage needs to encrypt the user's data. According to the requirements of data security storage, the user data stored in the cloud is secure enough, that is, the user data is encrypted by the cloud computing server, and there is no ordered information for non users.

In the process of cloud computing applications, to provide good data storage and transmission services, it is necessary to provide effective protection in the process of data upload and download. Taking data exchange as an example, using the services provided by cloud computing providers to store the data to be exchanged, there is a risk of data leakage, so data encryption algorithm has become the first choice to solve the problem of data security storage.

Assuming that the cloud computing provider is trusted, this paper studies the problem of data security storage in the cloud computing environment, that is, the problem of data security exchange between users and cloud processes. From the perspective of the security and processing performance of data encryption in cloud computing applications, this paper discusses how to achieve a data storage method suitable for cloud computing with both security and high performance.

At present, the commonly used data encryption algorithms can be divided into symmetric encryption algorithm and asymmetric encryption algorithm. Symmetric encryption algorithm is widely used and mature. Because of its fast encryption and decryption speed, it is widely used in large data transmission. In symmetric encryption algorithm, both sides use the same key to encrypt and decrypt the data. Its advantages are open algorithm, fast encryption speed and high efficiency. The disadvantage is that both sides use the same key, so the security cannot be guaranteed. Asymmetric encryption algorithm divides the traditional key into encryption key and decryption key to control the encryption and decryption respectively, and ensures the security of the key in terms of computational complexity. Its advantage is that the key system is flexible, but the problem is that it has a large amount of computation. The reliability of encryption mechanism mainly depends on the difficulty of decryption, including symmetric key encryption system and asymmetric key encryption system. The security of asymmetric key is high, but the speed of encryption and decryption is slow.

Symmetric encryption algorithm has the inherent problem that it is difficult to manage the key by using the same key, and the cost is high, so it is difficult to use in distributed network system. The asymmetric encryption algorithm is not suitable for the encryption and decryption of large amount of data because of the large amount of computation. Therefore, this paper proposes a cross idea of symmetric encryption and asymmetric encryption to solve the problem of data security storage in cloud computing.

3.2 Cross encryption scheme

3.2.1 N-DES encryption algorithm

DES algorithm is also called data encryption standard. It is a symmetric encryption algorithm developed by IBM in 1972. It was determined as the Federal Information Processing Standard (FIPS) by the National Bureau of standards of the federal government of the USA in 1976, and then spread widely in the world. Up to now, it still plays a very important role in the international information security arena.

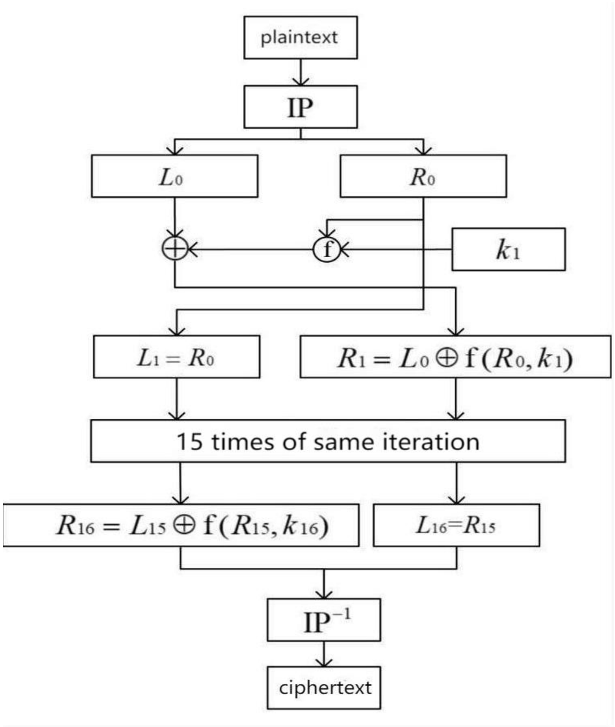
DES algorithm groups the plaintext by 64 bits, and the length of the key involved in the calculation is fixed to 64 bits (56 effective bits). The encryption process mainly includes three parts: initial permutation, 16-round cycle iteration and inverse initial permutation. And the sub-keys involved in the calculation in 16 rounds of iteration are extended from 56 bit keys.

As shown in Figure 2, the 64 bit input plaintext is divided into L0 and R0 parts by initial permutation, and then 16 rounds of the same iterative operation are carried out. Finally, the output ciphertext of 64 bit is obtained by inverse initial permutation. In each iteration, there is an XOR operation and an F function operation.

DES has many disadvantages, such as low data transmission rate, not suitable for long-term data protection, and vulnerable to differential key cracking. Therefore, scholars at home and abroad have made many attempts to improve DES algorithm. In this context, they have proposed more influential Triple DES algorithm.

Triple DES algorithm: Because the key length of traditional DES algorithm is short and easy to be cracked, in order to make up for this deficiency, researchers have proposed a Triple DES Encryption Algorithm (TDEA), that is, the key length of DES is increased by three times, and three different keys are used for triple encryption and decryption. The encryption process is as follows: first encrypt with the first key $k1$, then decrypt with the second key $k2$, and finally encrypt again with the third key $k3$, that is, $C = Ek3(DK2(Ek1M))$. The decryption is in reverse order, that is, $M = Dk1(\mathrm{E}K2(Dk3C))$. The core of TDEA is to use *k*1, *k*2, *k*3 to encrypt plaintext for many times, and the key length is three times of DES.

Figure 2 DES algorithm flow chart

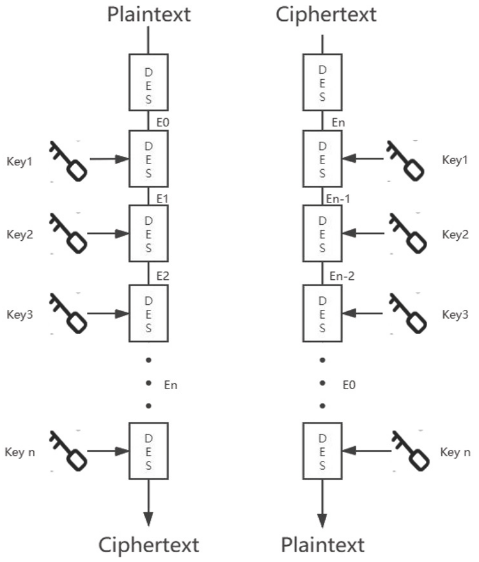

plaintext
IP
L_0
R_0
f
k_1
$L_1 = R_0$
$R_1 = L_0 \oplus f(R_0, k_1)$
15 times of same iteration
$R_{16} = L_{15} \oplus f(R_{15}, k_{16})$
$L_{16}=R_{15}$
IP^{-1}
ciphertext

Although this method increases the length of the key, improves the security strength of the algorithm, and effectively avoids brute force cracking, its calculation time is increased by *n*–1 times, so the operation efficiency is very low. In addition, although the key bits in TDEA are 168 bits, the threat of brute force cracking cannot be avoided for the current computer computing power.

N-DES algorithm: Inspired by 3-DES, this paper proposes N-DES algorithm, which the number of keys is user-defined. The N in N-DES means that the number of keys is uncertain. Manual input of the length of the key during encryption, to achieve the effect of one secret at a time. Increasing the number of bits in the key makes it exponentially more difficult to crack, and the attacker does not know the length of the key, making the exhausting time even longer. Since *N* is determined by users themselves, users can dynamically adjust the size of *N* according to their own hardware configuration and the size of the encrypted text to meet the diversified needs of users. The specific implementation process of N-DES algorithm is as follows.

There are two kinds of keys, the default key and the random key: the default key is only 64 bits, and the random key is (*n*–1) x 64 bits (to be exact (*n*–1) x 56 bits, because there are 8 check bits in each key). During encryption, the user inputs the number of random keys *m*, and the program encrypts (*m* + 1) times (because there is a default key). At the same time, the program generates a key file to save the key. When decryption occurs, the program reads the fixed key and the key in the key file for decryption. The encryption and decryption process of N-DES is shown in Figure 3.

Figure 3 Encryption and decryption process of N-DES



Algorithm 1. N-DES Encryption Algorithm

Input: Plaintext ***M***

Output: Encrypted ciphertext ***C***,

1. ***Choose the size of N***: Choose the number of random keys and name it *m*.
2. ***Random key generation***: Generate (*m*+1) keys randomly (*m* random keys and 1 default key).
3. ***Generation of file***: The program generates a key management file to store the generated keys
4. ***Plaintext encryption***: After inputting plaintext, encrypting with key1 first, and then encrypting with key2 again, and so on, until key *m*+1 is used to encrypt to generate ciphertext ***C.***
5. Output encrypted ciphertext ***C***.

Theoretically, the number of keys is infinite. Due to the constraints of computing time, it cannot be set too large, which will lead to a long wait when processing big data, let alone when we increase the length of the key. And the preservation of external key files is also a consideration.

3.2.2 RSA algorithm based on improved prime number decision

RSA (Rivest Shamir Adleman) is an algorithm based on large number decomposition proposed in 1977. Because large number decomposition is a recognised mathematical problem, RSA has high security, but its computing speed is much slower than DES. Although the rapid update of computer hardware makes the performance of computer break through the limit, it still takes a lot of time to break the large number decomposition. The encryption process for RSA is as follows:

Step 1: Message receiver B generates a key pair according to the rules, where the encryption key is PK and the decryption key is SK.

Step 2: Message receiver B sends the public key PK to message sender A and keeps the private key SK secret.

Step 3: If message sender A wants to send message M to B, it will encrypt the message using B's public key PK, which is represented as C = EPK(M), where C is the ciphertext and E is the encryption algorithm.

Step 4: After receiving the ciphertext C, B decrypts it with its own private key, Sk, and obtains the plaintext M = DSK(C), where D is the decryption algorithm.

Since the core algorithm of RSA is the modular power operation of large prime numbers, that is, large number self multiplication module. In order to improve the efficiency of RSA algorithm, it is necessary to solve the problem of operation speed of module power operation in RSA. The core complexity of modular power operation depends on the modular operation, which includes division operation. For a computer, a division operation requires several addition, subtraction and multiplication operations, which is quite time-consuming. Therefore, assuming that RSA algorithm can reduce or even avoid the operation of modulus taking, the performance of RSA algorithm will be significantly improved. Based on this, on the premise of ensuring the security of RSA algorithm, this paper makes a detailed study on the method of judging prime number which affects the operation speed of RSA algorithm module power, and carefully compares the advantages and disadvantages of deterministic and probabilistic prime number judgment algorithms. Then, this paper uses Montgomery fast power algorithm (Qin et al., 2002) to optimise the probability property number judgment algorithm (Lehman algorithm), and proposes an Optimised Prime Number Judgment Algorithm Based on Lehman Algorithm (OPJBLA). Finally, this paper applies OPJBLA to RSA algorithm to form an RSA Algorithm Based on Optimised Prime number Judgement (RSA_OPJ).

Because of the low efficiency and high complexity of the deterministic prime judgment algorithm, it is not suitable for the modular exponentiation of RSA algorithm, so this paper uses the probabilistic prime judgment algorithm to improve the modular exponentiation of RSA algorithm. There are three mainstream algorithms, Miller Rabin algorithm, Solovey Strassen algorithm (Zhao et al., 2018) and Lehman algorithm (Fu et al., 2011). And the Lehman algorithm has higher probability to judge prime number. Therefore, this paper chooses Lehman algorithm to improve.

Lehman algorithm is a new integer decomposition quantum algorithm, which makes use of multiple quantum Fourier transform and variable substitution to make the probability amplitude of non-target states except |0 > states become zero, so as to improve the probability of success of the algorithm. Compared with Shor integer decomposition quantum algorithm, the algorithm has a higher probability of success and is no longer dependent on the order r of the selected element in ZN. The computational complexity of the algorithm is polynomial time. The number of quantum logic gates required to run the algorithm is O(L3). In this paper, Montgomery fast power algorithm, which can greatly reduce modular power operation, is introduced to optimise Lehman algorithm to form an Optimised Prime number Judgment Algorithm Based on Lehman Algorithm (OPJBLA). The specific process is shown in algorithm 2.

Algorithm 2. An Optimised Prime number Judgment Algorithm based on Lehman Algorithm

Input: large number **A**, **B**, modulus ***N***, Lehman algorithm

Output: fast modular multiplication results of large numbers **A** and **B**

- *Data input*: input large numbers **A**, **B** and modulus ***N***
- *Base selection*: select a positive integer *R* which is coprime with ***N*** as the cardinal number. At the same time, when *R* is 2*k*, ***N*** should meet the following requirements: 2*k*-1≤*N*≤2*k* and *GCD*(*R*, *N*)=1
- *Montgomery fast power multiplication*: use Montgomery fast power algorithm to simplify Lehman algorithm and carry out modular multiplication on large numbers **A** and **B**, namely Montgomery(**A**, **B**, **N**)=ABR-1(mod *N*)
- Output the fast modular multiplication results of large numbers **A** and **B**

The main advantage of OPJBLA using Montgomery fast power algorithm is to transform division into shift operation, which not only simplifies the calculation process, but also improves the efficiency of large number power multiplication.

In order to improve the judging efficiency of OPJBLA applied to RSA algorithm, in the initial stage of prime number generation, all even numbers and numbers divisible by 5 are directly eliminated, and 53 small prime numbers are selected to form a filter array for in-depth filtering, and then OPJBLA is applied to the module power operation of RSA algorithm for rapid screening. All the screening methods complement each other to form a RSA Algorithm Based on Optimised Prime Number Judgment (RSA_OPJ). The specific improvement steps of RSA_OPJ are shown in algorithm 3.

Algorithm 3. RSA Algorithm Based on Optimised Prime Number Judgment

Input: plaintext **M**, random large array **N**, Lehman algorithm

Output: encrypted ciphertext ***C***, decrypted plaintext ***M***

- *Large array generation and screening*: generates a large array ***N*** randomly except even numbers and numbers divisible by 5. Then, select 53 small primes and use the remainder method to filter large array ***N***
- *Generate large prime numbers p and q*: combine the steps above, OPJBLA and Lehman algorithm optimised by Montgomery fast power algorithm to generate two large prime numbers *p* and *q*
- *Encryption process*: input plaintext ***M***, generate RSA key with two large prime numbers *p* and *q* to encrypt plaintext and generate ciphertext ***C***, finally output encrypted ciphertext ***C***
- *Decryption process*: input ciphertext ***C***, generate RSA key with two prime numbers *p* and *q* to decrypt ciphertext and generate plaintext ***M***, finally output decrypted plaintext ***M***

3.2.3 Cross encryption scheme based on N-DES and RSA_OPJ

Because the encryption and decryption process of symmetric encryption algorithm (such as DES) is very fast, and the encryption efficiency is very high, it is very suitable for data encryption in the cloud environment with fast update frequency and large amount of data. But because the key is easy to be stolen in the process of transmission, the security is not high. The encryption and decryption of asymmetric encryption algorithm (such as RSA) is very slow, the encryption efficiency is very low, and it is not suitable for considerable data encryption in the cloud environment. However, due to the difficulty of cracking and the key is not afraid of being stolen, the security is very high. Therefore, in order to solve this problem, this paper adopts a cross encryption scheme combining symmetric encryption and asymmetric encryption, that is, N-DES and RSA_OPJ are used to encrypt the data in the cloud environment. The specific process is shown in Figure 4.

In the process of encryption of data plaintext, this paper proposes a cross encryption algorithm to divide original data into sub-data block by a certain data block partitioning strategy.

The block size intersecting changes the sub-data blocks, according to the encryption efficiency and security of symmetric encryption and asymmetric encryption. Similarly, symmetric encryption is more efficient for large data blocks, while it is low efficient for small data blocks. The asymmetric encryption with low efficiency is used for small data blocks to improve the security of data encryption and make up for the defects of asymmetric encryption in the efficiency of performance. The specific process is shown in Figure 5 and the detailed encryption process is as follows:

Step 1: User provides plaintext M to be encrypted.

Step 2: Select different encryption parameters L and K according to the requirements. L is the size of the encryption unit, and K is Encrypted data ratio.

Step 3: The plaintext M divided into text blocks according to the size of L, and suppose $F = \{f1, f2, f3\ldots fn\}$, the size of each block is L.

Step 4: Each sub-data block $fi(i\in 1,2\ldots n)$ is divided into two parts according to the encryption data ratio k, denuded as $fi1$ and $fi2$. And the data size ratio of the two parts before and after is K.

Step 5: All $fi1$ and $fi2$ are encrypted using DES and RSA respectively. The number of the encrypted data is $2n$.

Step 6: Merge all the encrypted data parts obtained in Step 5 to form the final ciphertext C.

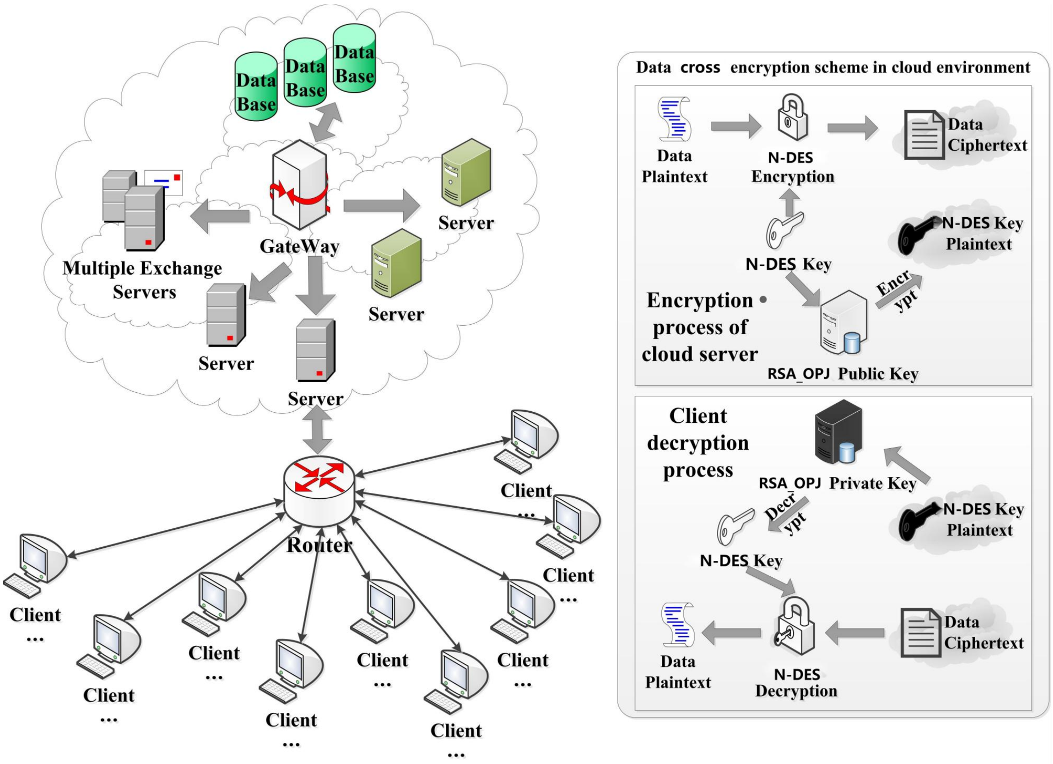
The decryption process of the file is the inverse process of the encryption process. The detailed decryption steps are as followings:

Step 1: Enter the ciphertext C for the data to be decrypted.

Step 2: Obtain the encryption unit size L and the encryption data ratio K in the encryption process as the decryption parameters.

Step 3: The ciphertext C is divided into the following parts according to the encryption unit size L and the encryption data ratio: C = {C11, C12, C21,C22...Cn1, Cn2}

Figure 4 Schematic diagram of cross encryption scheme based on N-DES and RSA_OPJ in cloud environment

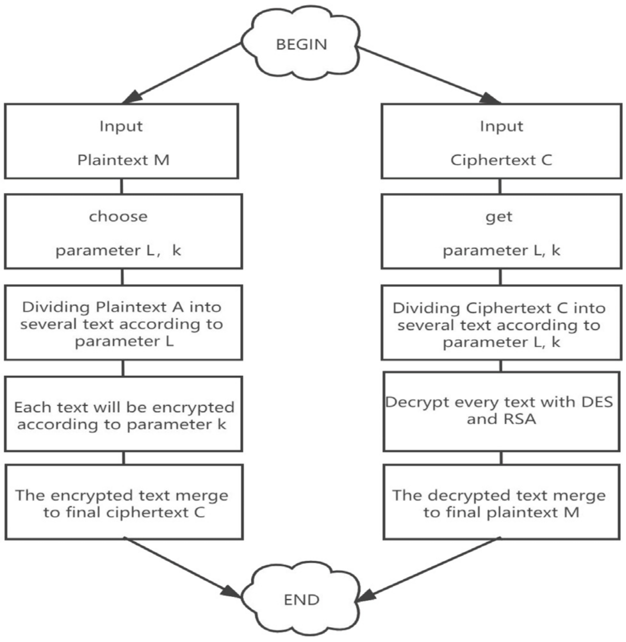


Step 4: AES algorithm is used to decrypt all Cn1, and RSA algorithm is used to decrypt all Cn2, and 2N plaintext parts of data are obtained.

Step 5: Merge the plaintext generated in Step 4 to get the whole plaintext M.

Because the cross encryption algorithm is a linear mixture of DES and RSA with some strategy. In order to crack the ciphertext, not only the size of the specific encryption unit of the data and the encryption ratio of the data need to be known, but also the DES algorithm and RSA algorithm need to be deciphered respectively. Compared with the simple DES and RSA encryption algorithm, the difficulty of ciphertext deciphering is obviously increased.

Figure 5 The flow chart of cross encryption on data plaintext


BEGIN
Input
Plaintext M
choose
parameter L, k
Dividing Plaintext A into several text according to parameter L
Each text will be encrypted according to parameter k
The encrypted text merge to final ciphertext C
Input
Ciphertext C
get
parameter L, k
Dividing Ciphertext C into several text according to parameter L, k
Decrypt every text with DES and RSA
The decrypted text merge to final plaintext M
END

3.3 *Verification and analysis of cross encryption scheme*

3.3.1 *Experimental environment setting*

In order to verify that the scheme can effectively encrypt the user data in the cloud computing environment, this paper uses the text information of a real patient radiotherapy plan to carry out the effect experiment. The experimental environment settings are given below, as shown in Table 1.

Table 1 Experimental environment parameters

Name	*Parameter*
CPU	Intel Core i5
Memory capacity	24 GB
Hard disk capacity	1 T
Operating system	Linux CentOS6.4
Development software	Myeclipse
Development platform	Hadoop
Cloud server	Personal Alibaba cloud server

3.3.2 *Performance experiment and result analysis of cross scheme*

Table 2 shows the comparison of the time spent on encrypting a small amount of user data (250 B) when the two encryption algorithms are running separately. There are ten groups of experiments, each group runs 50 times, and the average encryption time is taken. And the value of *N* in this experiment is set to 3.

Table 2 Performance comparison of two encryption schemes

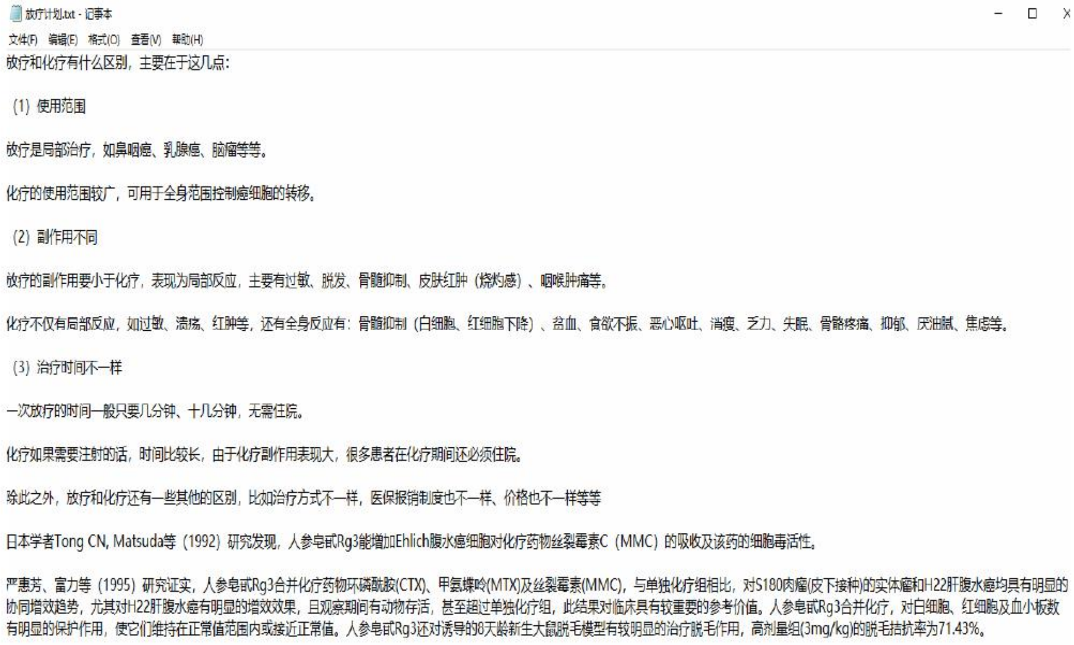
Operation time	*Comparison scheme (DES and RSA)*	*Cross scheme (N-DES (N=3)and RSA_OPJ)*
First time	4864 ms	4528 ms
Second time	4896 ms	4555 ms
Third time	4793 ms	4481 ms
Fourth time	4887 ms	4506 ms
Fifth time	4822 ms	4512 ms
Sixth time	4901 ms	4503 ms
Seventh time	4912 ms	4497 ms
Eighth time	4893 ms	4497 ms
Ninth time	4789 ms	4536 ms
Tenth time	4906 ms	4516 ms

As can be seen from Table 2, when encrypting short messages, the time difference between the two encryption schemes remains at the level of about 400 ms. Human beings can hardly perceive this subtle time gap, but it is only one encryption operation. If the encryption times exceed a certain number, the time-consuming gap will become considerable. For example, a web page user uses static data encryption, and the result after encryption is the same, that is, each encryption uses the same key. Therefore, as long as a malicious user intercepts the encrypted message and simulates the form submission information, it can cheat the encryption system to directly invade. Obviously, this static encryption method is not feasible. Even the RSA algorithm using public key cryptosystem has the same result. This risk can be avoided only if the data encryption algorithm uses a different key for each encryption. Therefore, in normal life, it is reasonable and safe to use different keys for each encryption. In addition, if the super large amount of user data (more than 1 PB) is encrypted, the time gap required will be very obvious. Therefore, the encryption efficiency of the cross encryption scheme based on N-DES and RSA_OPJ has obvious advantages over the traditional hybrid encryption scheme based on DES and RSA.

3.3.3 *Verification and analysis of cross scheme*

In order to verify that the scheme can effectively encrypt the user data in the cloud computing environment, this paper uses the text information of a real patient radiotherapy plan to carry out the effect experiment, and then the differences between local data and cloud data are compared and analysed. In this paper, the real radiotherapy plan data of a patient is encrypted with a cross encryption scheme and stored in the cloud. Figure 6 shows part of the original content of the stored data, and Figure 7 shows part of the ciphertext data content viewed in the cloud background.

Figure 6 Raw data of a patient’s radiotherapy plan



放疗和化疗有什么区别，主要在于这几点：

（1）使用范围

放疗是局部治疗，如鼻咽癌、乳腺癌、脑瘤等等。

化疗的使用范围较广，可用于全身范围控制癌细胞的转移。

（2）副作用不同

放疗的副作用要小于化疗，表现为局部反应，主要有过敏、脱发、骨髓抑制、皮肤红肿（烧灼感）、咽喉肿痛等。

化疗不仅有局部反应，如过敏、溃疡、红肿等，还有全身反应有：骨髓抑制（白细胞、红细胞下降）、贫血、食欲不振、恶心呕吐、消瘦、乏力、失眠、骨骼疼痛、抑郁、厌油腻、焦虑等。

（3）治疗时间不一样

一次放疗的时间一般只要几分钟、十几分钟，无需住院。

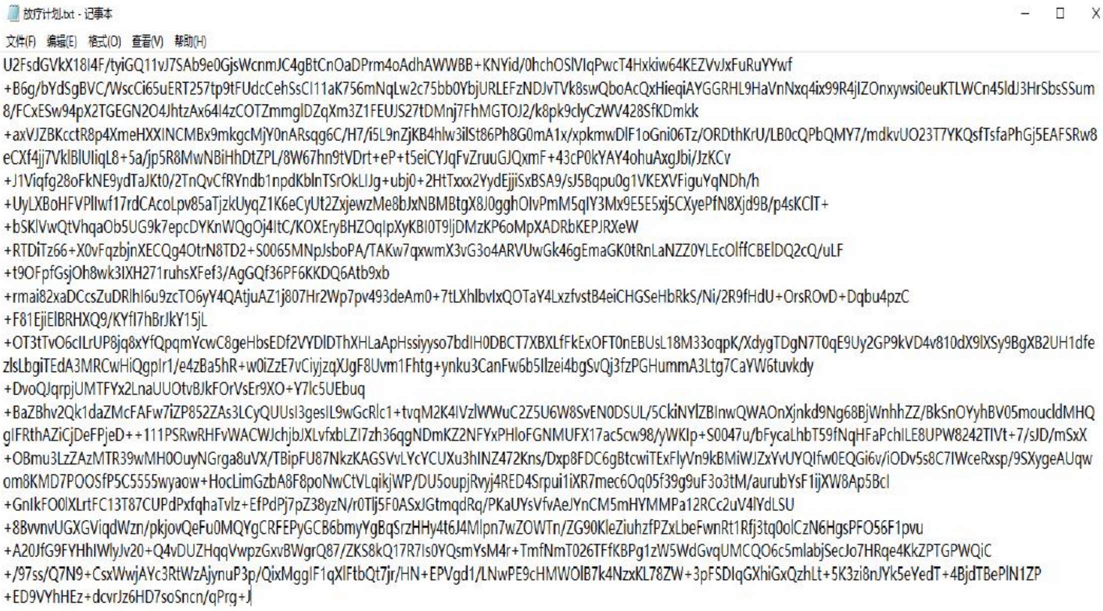
化疗如果需要注射的话，时间比较长，由于化疗副作用表现大，很多患者在化疗期间还必须住院。

除此之外，放疗和化疗还有一些其他的区别，比如治疗方式不一样，医保报销制度也不一样、价格也不一样等等

日本学者Tong CN, Matsuda等（1992）研究发现，人参皂甙Rg3能增加Ehlich腹水癌细胞对化疗药物丝裂霉素C（MMC）的吸收及该药的细胞毒活性。

严惠芳、富力等（1995）研究证实，人参皂甙Rg3合并化疗药物环磷酰胺(CTX)、甲氨蝶呤(MTX)及丝裂霉素(MMC)，与单独化疗组相比，对S180肉瘤(皮下接种)的实体瘤和H22肝腹水癌均具有明显的协同增效趋势，尤其对H22肝腹水癌有明显的增效效果，且观察期间有动物存活，甚至超过单独化疗组，此结果对临床具有较重要的参考价值。人参皂甙Rg3合并化疗，对白细胞、红细胞及血小板数有明显的保护作用，使它们维持在正常值范围内或接近正常值。人参皂甙Rg3还对诱导的8天龄新生大鼠脱毛模型有较明显的治疗脱毛作用，高剂量组(3mg/kg)的脱毛拮抗率为71.43%。

Figure 7 Encrypted data stored in Alibaba cloud



```
U2FsdGVkX18l4F/tyiGQ11vJ7SAb9e0GjsWcnmJC4gBtCnOaDPrm4oAdhAWWBB+KNYid/0hchOSlVIqPwcT4Hxkiw64KEZVvJxFuRuYYwf
+B6g/bYdSgBVC/WscCi65uERT257tp9tFUdcCehSsCl11aK756mNqLw2c75bb0YbjURLEFzNDJvTVk8swQboAcQxHieqiAYGGRHL9HaVnNxq4ix99R4jIZOnxywsi0euKTLWCn45ldJ3HrSbsSSum
8/FCxESw94pX2TGEGN2O4JhtzAx64l4zCOTZmmglDZqXm3Z1FEUJS27tDMnj7FhMGTOJ2/k8pk9clyCzWV428SfKDmkk
+axVJZBKcctR8p4XmeHXXINCMBx9mkgcMjY0nARsqg6C/H7/i5L9nZjKB4hlw3ilSt86Ph8G0mA1x/xpkmwDlF1oGni06Tz/ORDthKrU/LB0cQPbQMY7/mdkvUO23T7YKQsfTsfaPhGj5EAFSRw8
eCXf4jj7VklBlUliqL8+5a/jp5R8MwNBiHhDtZPL/8W67hn9tVDrt+eP+t5eiCYJqFvZruuGJQxmF+43cP0kYAY4ohuAxgJbi/JzKCv
+J1Viqfg28oFkNE9ydTaJKt0/2TnQvCfRYndb1npdKblnTSrOkLIJg+ubj0+2HtTxxx2YydEjjiSxBSA9/sJ5Bqpu0g1VKEXVFiguYqNDh/h
+UyLXBoHFVPllwf17rdCAcoLpv85aTjzkUyqZ1K6eCyUt2ZxjewzMe8bJxNBMBtgX8J0gghOlvPmM5qlY3Mx9E5E5xj5CXyePfN8Xjd9B/p4sKClT+
+bSKlVwQtVhqaOb5UG9k7epcDYKnWQgOj4ltC/KOXEryBHZOqIpXyKBI0T9ljDMzKP6oMpXADRbKEPJRXeW
+RTDiTz66+X0vFqzbjnXECQg4OtrN8TD2+S0065MNpJsboPA/TAKw7qxwmX3vG3o4ARVUwGk46gEmaGK0tRnLaNZZ0YLEcOlffCBElDQ2cQ/uLF
+t9OFpfGsjOh8wk3IXH271ruhsXFef3/AgGQf36PF6KKDQ6Atb9xb
+rmai82xaDCcsZuDRlhI6u9zcTO6yY4QAtjuAZ1j807Hr2Wp7pv493deAm0+7tLXhlbvIxQOTaY4LxzfvstB4eiCHGSeHbRkS/Ni/2R9fHdU+OrsROvD+Dqbu4pzC
+F81EjiElBRHXQ9/KYfI7hBrJkY15jL
+OT3tTvO6cILrUP8jq8xYfQpqmYcwC8geHbsEDf2VYDlDThXHLaApHssiyyso7bdIH0DBCT7XBXLfFkExOFT0nEBUsL18M33oqpK/XdygTDgN7T0qE9Uy2GP9kVD4v810dX9lXSy9BgXB2UH1dfe
zlsLbgiTEdA3MRCwHiQgplr1/e4zBa5hR+w0iZzE7vCiyjzqXJgF8Uvm1Fhtg+ynku3CanFw6b5Ilzei4bgSvQj3fzPGHummA3Ltg7CaYW6tuvkdy
+DvoQJqrpjUMTFYx2LnaUUOtvBJkFOrVsEr9XO+Y7lc5UEbuq
+BaZBhv2Qk1daZMcFAFw7iZP852ZAs3LCyQUUsI3gesIL9wGcRlc1+tvqM2K4IVzlWWuC2Z5U6W8SvEN0DSUL/5CkiNYlZBlnwQWAOnXjnkd9Ng68BjWnhhZZ/BkSnOYyhBV05moucldMHQ
glFRthAZiCjDeFPjeD++111PSRwRHFvWACWJchjbJXLvfxbLZI7zh36qgNDmKZ2NFYxPHloFGNMUFX17ac5cw98/yWKIp+S0047u/bFycaLhbT59fNqHFaPchILE8UPW8242TIVt+7/sJD/mSxX
+OBmu3LzZAzMTR39wMH0OuyNGrga8uVX/TBipFU87NkzKAGSVvLYcYCUXu3hINZ472Kns/Dxp8FDC6gBtcwiTExFlyVn9kBMiWJZxYvUYQIfw0EQGi6v/iODv5s8C7IWceRxsp/9SXygeAUqw
om8KMD7POOSfP5C5555wyaow+HocLimGzbA8F8poNwCtVLqikjWP/DU5oupjRvyj4RED4Srpui1iXR7mec6Oq05f39g9uF3o3tM/aurubYsF1ijXW8Ap5BcI
+GnIkFO0lXLrtFC13T87CUPdPxfqhaTvlz+EfPdPj7pZ38yzN/r0Tlj5F0ASxJGtmqdRq/PKaUYsVfvAeJYnCM5mHYMMPa12RCc2uV4lYdLSU
+8BvvnvUGXGViqdWzn/pkjovQeFu0MQYgCRFEPyGCB6bmyYgBqSrzHHy4t6J4Mlpn7wZOWTn/ZG90KleZiuhzfPZxLbeFwnRt1Rfj3tq0olCzN6HgsPFO56F1pvu
+A20JfG9FYHhlWlyJv20+Q4vDUZHqqVwpzGxvBWgrQ87/ZKS8kQ17R7Is0YQsmYsM4r+TmfNmT026TFfKBPg1zW5WdGvqUMCQO6c5mlabjSecJo7HRqe4KkZPTGPWQiC
+/97ss/Q7N9+CsxWwjAYc3RtWzAjynuP3p/QixMggIF1qXlFtbQt7jr/HN+EPVgd1/LNwPE9cHMWOlB7k4NzxKL78ZW+3pFSDIqGXhiGxQzhLt+5K3zi8nJYk5eYedT+4BjdTBePlN1ZP
+ED9VYhHEz+dcvrJz6HD7soSncn/qPrg+J
```

It can be seen from Figures 6 and 7 that the cross encryption scheme based on N-DES and RSA_OPJ proposed in this paper ensures the security of data transmission process. That is to say, before the data is transmitted, it goes through the process of identity authentication of both sides, and the data to be exchanged is encrypted, so as to ensure the security of the data in the communication process. In addition, the scheme can ensure the security of stored data. Because the user generated data is encrypted by cross encryption to generate ciphertext, the key and ciphertext are encrypted by RSA_OPJ algorithm, even if the malicious user obtains the ciphertext in the transmission process, it cannot be cracked because there is no private key of the receiver and the split ratio of the sub-block is unknown. Therefore, it can be considered that the stored data has high security.

To sum up, the cross encryption scheme based on N-DES and RSA_OPJ proposed in this paper can meet the requirements of data storage security in the development of cloud computing, and has high operation efficiency.

4 Conclusions and prospects

With the in-depth development of cloud computing in the future, the data scale of users is also expanding, and the data in cloud database is also showing a huge increase. The development of cloud computing technology provides a new direction for the storage and processing of massive data, but it also brings the security problems of data storage. In order to solve the problem of data security storage in cloud environment, this paper uses symmetric encryption and asymmetric encryption to encrypt user data, proposes two improved algorithms and a cross encryption scheme, and then gives full play to the advantages of the improved algorithm. That is, on the premise of ensuring the security of the key, the symmetric and asymmetric encryption algorithm is used to encrypt the user's data. At the same time, the asymmetric algorithm RSA_OPJ has the characteristics of high security but slow speed, which is not suitable for large-scale data encryption, and encrypts the private key which has very small amount of data. On the premise that the data security reaches the asymmetric encryption algorithm, the efficiency of the algorithm is close to the encryption speed of the symmetric encryption algorithm, which solves the problem of user data security storage in the current cloud computing environment to a certain extent. The cross encryption scheme can be applied not only to secure storage in the cloud computing environment, but also to many fields such as blockchain, database encryption, e-commerce, and identity authentication. Of course, the cross encryption scheme proposed in this paper can improve the security of data storage, but the data is only plain text, and does not cover the encryption and decryption of pictures, audio and video. In the future, we will continue to study and improve from the above aspects.

Acknowledgements

This work is partially supported by Humanities and social sciences research project of the Ministry of Education (No. 20YJAZH046), Teaching reform project of Bistu (No. 2019JGZD04), and recommended by 4th EAI International Conference on Smart Grid and Internet of Things.

References

Feng, C.S, Qin, Z.G. and Yuan, D. (2015) 'Cloud data secure storage technology', *Chinese Journal of Computers*, Vol. 38, No. 1, pp.150–163.

Feng, D.G., Zhang, M., Zhang, Y. et al. (2011) 'Study on cloud computing security', *Journal of Software*, Vol. 22, No. 1, pp.71–83.

Fu, X.Q., Bao, W.S., Zhou, C. et al. (2011) 'Integer factorization quantum algorithm with high probability', *Acta Electronica Sinica*, Vol. 39, No. 1, pp.35–39.

Gao, N.N., Li, Z.C. and Wang, Q. (2006) 'A reconfigurable architecture for high speed implementation of DES, 3DES and AES', *Acta Electronica Sinica*, Vol. 34, No. 8, pp.1386–1390.

Hoefer, C.N. and Karagiannis, G. (2011) 'Cloud computing services: taxonomy and comparison', *Journal of Internet Services & Applications*, Vol. 2, No. 2,pp.81–94.

Jain, N., Ajnar, D.S. and Jain, P.K. (2019) 'Optimization of advanced encryption standard algorithm (AES) on field programmable gate array (FPGA)', *International Conference on Communication and Electronics Systems*, India.

Kanna, G.P. and Vasudevan, V. (2016) 'Enhancing the security of user data using the keyword encryption and hybrid cryptographic algorithm in cloud', *International Conference on Electrical*, USA.

Khanezaei, N. and Hanapi, Z.M. (2015) 'A framework based on RSA and AES encryption algorithms for cloud computing services', *Systems, Process & Control*, USA.

Li, F., Gong, Z.Y., Lei, F.F. et al. (2019) 'Summary of fast prime generation methods', *Journal of Cryptologic Research*, Vol. 6, No. 4, pp.463–476.

Li, J., Yao, W., Zhang, Y. et al. (2017) 'Flexible and fine-grained attribute-based data storage in cloud computing', *IEEE Transactions on Services Computing*, Vol. 5, No. 1, p.1.

Li, X.J. (2010) 'Research on data storage system based on cloud computing', *Silicon Valley*, Vol. 1, No. 19, pp.73–74.

Mahalle, V.S. and Shahade, A.K. (2014) 'Enhancing the data security in Cloud by implementing hybrid (RSA & AES) encryption algorithm', *International Conference on Power*, USA.

Qiang, L. (2019) 'Research on the architecture and key technologies of cloud computing', *Proceedings of 3rd International Conference on Mechatronics Engineering and Information Technology*, Wuhan, China.

Qin, X.D., Xin, Y.W. and Lu, G.Z. (2002) 'Research and optimization of Miller Rabin algorithm', *Computer Engineering*, Vol. 28, No. 10, pp.55–57.

Rivest, R.L., Shamir, A. and Adleman, L. (1978) 'A method for obtaining digital signatures and public-key cryptosystems', *Communications of the ACM*, Vol. 21, No. 2, pp.120–126.

Seroussi, G. (1999) 'Elliptic curve cryptography', *Information Theory and Networking Workshop*, USA.

Shi, J., Li, H. and Zhou, L.D. (2012) 'Research on secure data storage based on cloud computing', *Journal of Nanjing Normal University*, Vol. 35, No. 3, pp.138–142.

Sun, S.S. (2013) *Research on cloud computing data security based on Hadoop*, Wuhan University of Technology, Wuhan, China.

Tuchman, W. (1979) 'Hellman presents no shortcut solutions to thedes', *IEEE Spectrum*, Vol. 16, No. 7, pp.40–41.

Wang, S.P., Wang, Y.M. and Zhang, Y.L. (2003) 'A confirmer signature scheme based on DSA and RSA', *Journal of Software*, Vol. 14, No. 3, pp.588–593.

Yellamma, P., Narasimham, C. and Sreenivas, V. (2014) 'Data security in cloud using RSA', *4th International Conference on Computing*, USA.

Yi, Z.D., Duan, Y.Z., Xiao, C.W. et al. (2012) 'Cloud computing security: concept, status quo and key technologies', *Proceedings of the 27th National Conference on Computer Security*, China.

Zhao, J.C. (2019) *Data storage security technology and application based on cloud platform*, Nanjing University of Posts and Telecommunications, Nanjing, China.

Zhao, Y.W., Liu, F.F., Jiang, L.J. et al. (2018) 'Research on large integer multiplication Schnhage-Strassen algorithm's multi-core parallelization', *Journal of Software*, Vol. 29, No. 12, pp.3604–3613.