A Mobile Terminal Security Strategy Based On the Cloud Storage

Wang Hui, Tang Junyong
School of Computer Science and Engineering
Xi'an Technological University, Xi'an 710032, China
Email: 277019826@qq.com

Abstract. With the emergence of mass storage systems and the development of the Distributed File System, Cloud storage system has become the focus of the industry. The cloud storage services on mobile terminal have been putted on the agenda based on the rapid development of intelligent mobile terminal. Based on the analysis of the architecture of HDFS and Dynamo, a mobile Terminal Security strategy is presented in this paper. The database technology and t he dynamic consistent hashing algorithm are adopted to deal with different target groups. According to the storage costs of nodes, data would be integrated scheduling by the storage system. Make full use of the advantages of AES(Advanced Encryption Standard) and RSA. A solution that combines AES and RSA encryption algorithm is proposed to implement the mobile terminal cloud storage security. Through the theoretical analysis and the simulation results, the cloud storage strategy proposed in this paper can make the cloud system achieve load balance. Moreover, multi-copy mechanism can improve the overall efficiency of the system.

Keywords: Cloud Storage, HDFS, Dynamo, Dynamic Consistent Hashing Algorithm, AES, RSA, Multi - Copy Mechanism

1. Introduction

With the rapid development of the Internet of things, more and more people are used to using mobile devices such as smart phones to surf the Internet, chat, browse news, shopping entertainment, and view all kinds of information. Traditional mobile cloud storage systems have lower storage density, the overall storage efficiency is low too. Traditional cloud storage systems do not adapt well to different application environment sand do not guarantee the integrity and confidentiality of cloud data. The cloud storage service does not guarantee that the data and operation of mobile users will not be lost, damaged, leaked, or illegally exploited by malicious or non-malicious. So it' s very dangerous for sensitive data to be stored directly in the cloud. Simple encryption techniques have key management issues and can't support complex requirements such as query, parallel modification, and fine-grained authorization. As a result, a mobile cloud storage security technology solution is proposed in this paper, which enables reliable and secure cloud storage.

First, the distributed file system (the hadoop distributed file system, HDFS) and Dynamo would be compared in this paper, and then the dynamic consistency hash algorithm is introduced to realize the processing of data in different size. According to the storage cost of each storage node, select the optimal storage node to implement the access of mobile cloud storage. The relational database is used for storing indexes in small object files, and the class Dynamo system model is used to handle large object files.

The cloud storage system will choose the closest copy when the mobile terminal makes a request. This method can effectively improve the storage efficiency of the cloud system and ensure the load balance of the system. On the basis of implementing the mobile cloud storage, we make full use of advantages of AES and RSA algorithms, a cloud storage security scheme for mobile is proposed in this paper. The solution combines AES and RSA encryption algorithms to improve the shortcomings of the cloud storage system. The reliability model of the cloud storage system is also proposed in the paper. Finally, a series of simulation experiments show that the proposed cloud storage security technology scheme is a reliable scheme with higher security.

2. System Architecture of the Mobile Cloud Storag

Cloud Storage is developed on the basis of clustering techniques and embedded virtualized technologies, which is an extension of cloud computing. Grid technology, cluster technology and distributed system are used in cloud storage, which coordinated all different types of storage devices in the network. All these technologies and devices can be cooperated with cloud storage to provide the required data storage capabilities and related business visit. Cloud storage is not a single storage device. The nature of cloud storage is not storage. The essence of cloud storage is providing services. Different ways are taken to deal with different sizes of objects. In this way, system architecture of the mobile cloud storage is designed . System architecture of the mobile cloud storage is shown in figure 1.

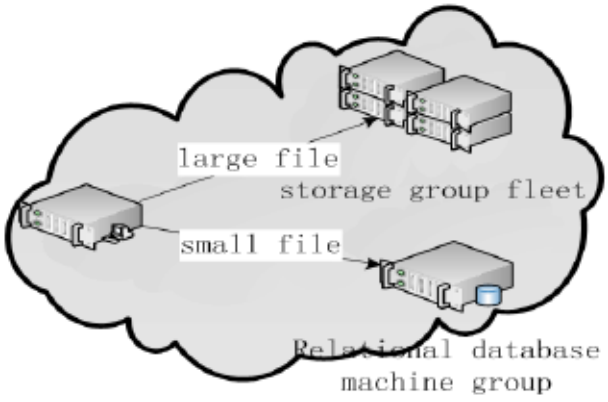

Figure.1 System Architecture of the Mobile Cloud Storage

Users access the network through a mobile terminal. The lowest load in the dispatcher group is selected by the mobile terminal and then communicates with it. Depending on the size of the file to determine whether control is handed over to the relational database machine group or to the storage group. Small files are handed over to the relational database machine group, and large files are processed by the storage group machine group. The copy mechanism was introduced to improve the reliability of the cloud storage system, multiple-copy mechanism can effectively improve system efficiency. When a mobile terminal makes a request, the closest copy can be chosen.

The primary copy would be selected first in traditional cloud storage system. The request is made to the standby copy only when the master copy is wrong. This process affects the speed of the traditional cloud storage system without considering the location of the copy.

Different storage policies and backup solutions are described in this article, which are used in the relational database machine group and the storage group machine group.

3. Storage Policies and Backup Plans

HDFS and Dynamo are reliable solutions that are commonly used in the cloud storage system. HDFS is a distributed file system that is suitable for running on common hardware. HDFS has good fault tolerance and can be used for inexpensive hardware. HDFS provides data access mechanisms with high throughput that can be widely applied to large data sets. Distributed file systems are developed on the infrastructure of the Apache Nutch search engine and apply to batch processing for data storage. HDFS emphasizes data throughput rather than response time for accessing data. The program in HDFS has a lot of data sets. File size of the HDFS is typically gigabyte to terabyte. As a result, terabytes of large files can be supported in HDFS through higher aggregated data bandwidth. And hundreds of nodal devices can be contained in a cluster, which allowing the terabytes of large files to be supported in it.

Dynamo is storage platform of amazon, and the key-value pair is used to store data in the key-value database schema. Dynamo has better availability and the higher extensibility. In Dynamo, the data are segmented according to the hash algorithm used in distributed file systems. And then all these segmented data are stored in separate nodes. The corresponding node is searched according to the hash value of the key, so that the read operation is realized in Dynamo. The consistency hash algorithm is used by Dynamo. At that time, it’ s not the exact hash value, but a range of hash values. When the hash value of the key is in this range, it will be searched clockwise along the loop, and the first node encountered is what we need. The consistency hash algorithm is improved by Dynamo, and in the ring， a set of devices are acted as a node rather than only one device is acted as a node. The synchronization mechanism is used to achieve the consistency of the data.

In HDFS the numbers of the copies are set to be 3. Whether the data would be stored in the node or not depends on the capacity of the node. The greater the capacity of the node, the greater the probability that the data will be stored in this node. So, when the capacity of the node is quite different, the nodes with large storage capacity in the system would be overloaded. The copy mechanism proposed in this paper can achieve load balancing, and the reliability and availability of cloud storage are also effectively improved. The system replication policy includes dynamic replica policies and static replica policies. The static copy strategy refers to the numbers of copies. The placement is fixed from the start to the data failure. The dynamic copy strategy is a strategy that system can adjust the numbers of copies in real time and their location, depending on performance requirements, load, and so on. The copy strategies for small files and large files are described as follows.

3.1 The copy strategy for a small file

Files that do not exceed 10MB are defined as small files. The SQL Server relational database is used as the copy strategy for small files. After receiving the file from the mobile terminal, the dispatcher will judge its size first, and then, once the small file is identified, it will be handed over to the relational database machine group. The correlation properties of the file are stored in the database table. The optimal node with the lightest load is dynamically selected by the database machine group. The lightest load database server in the machine group is selected to store the file, and keep a copy to ensure the reliability of the data. The IP address of the database server will be stored in the primary server. The IP address of the database server is retrieved and got from the primary server and then interacts with the database server. The storage processing pattern for small files is shown in figure 2.

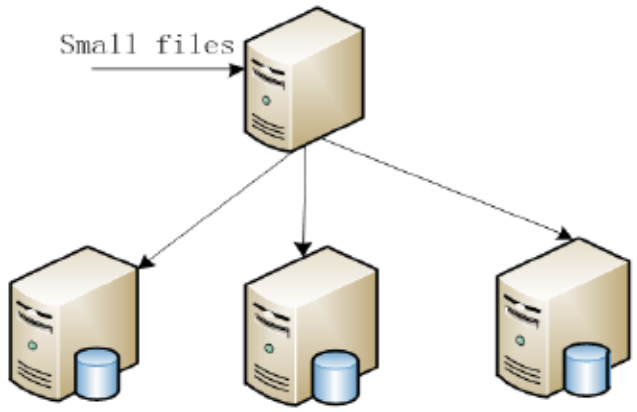

Figure.2 The Storage Processing Pattern for Small Files

The corresponding relationship is stored in the data table in the primary server, and the corresponding relationship is the file name and the server IP address. The data table is shown in table I.

Table 1 Data Tables in the Primary Server

Field	Typ	Length	Note
ID	int		Serial number
fname	varchar	255	Filename
IP	varchar	15	IP address

File names, file sizes, and content are stored in the database server. The file information table is shown in table 2.

Table 2 The File Information table

Field	Typ	Length	Note
ID	int		Serial number
fname	varchar	255	Filename
fsize	int		File size
creat	datetime		creation time
context	mediumtext		Content

3.2 The Copy strategy for large files

Files larger than 10MB are called large files. The storage group is used as a copy strategy for large files. The system architecture of the storage group is fully connected. The system architecture is shown in figure 3. The PC is used as a storage medium in the storage group. However, the reliability of the PC is not high, and it will even fail when the data is stored. Therefore, a copy is required to ensure that the data is reliable.

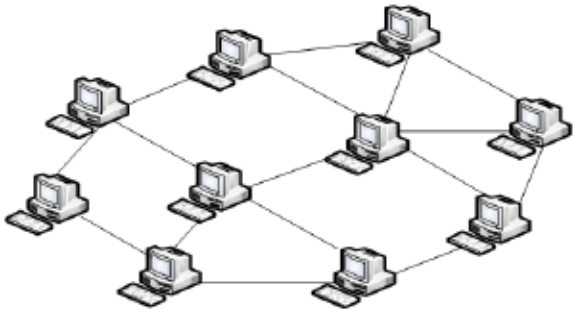

Figure.3 The System Architecture Diagram of the Storage Group

In the storage group system, all information about adjacent nodes are stored in every PC. The needed storage nodes can be found quickly through querying the information stored in the nodes.

The structure of storage space in the storage group is ring, and at the same time, the method of the unified addressing is adopted. In the storage group, the difference in performance of the PC can be offset by the virtual contiguous storage space. First, the hash Algorithm message-digest Algorithm 5 is used to implement system address conversion. The actual physical address is processed and converted to 32-bit information string through the MD5 algorithm, And then these information string are stored in the virtual continuous address. Thus, the differences in performance between devices will be offset. The loop storage structure of the storage group is shown in figure 4.

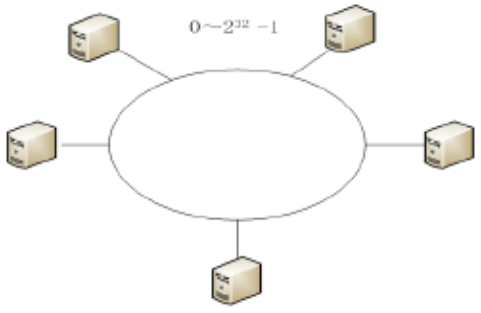

Figure.4 The Loop Storage Structure of the Storage Group

The converted address is mapped to the virtual storage space loop of the storage group through the MD5 algorithm. The device is found in the clockwise direction, and then the data is stored in the first PC mapped. The data is backed up to two adjacent PC. The larger the amount of data in the system, the more uniform the spatial distribution will be. The data are stored when the routing of the corresponding PC and adjacent PC are updated. The routing information table is shown in table 3.

Table 3 Routing Table

Field	Typ	Length	Note
ID	int		Serial number
fname	varchar	255	Filename
fsize	int		File size
IP	varchar	15	IP address

The IP address of the PC device where the file replica is located is stored in the IP field In the routing information table. The IP field is the routing information for the adjacent PC. Once a node fails, all the information stored in the node are backed up and the routing information of the adjacent node is modified in time. According to the principle of the consistency hash algorithm, the storage space of the new PC device will be mapped to the new virtual address space when a new device needs to be added to the storage group. The existing space on the ring will not be changed, and this method can be very effective in avoiding the vibration of the address space. Meanwhile, the routing information on the adjacent PC are updated. The process of adding a PC is as shown in figure 5.

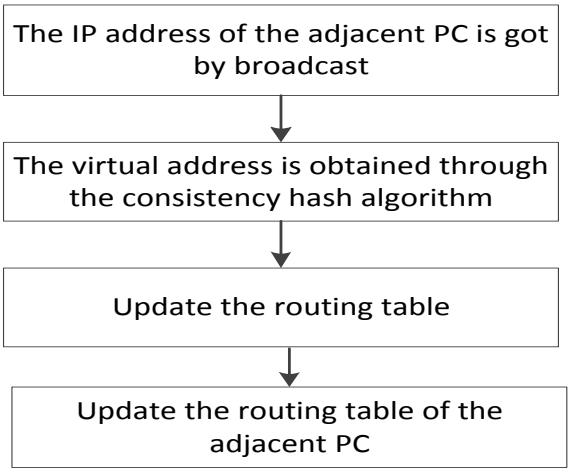

Figure.5 The Process of Adding A PC

The process of exiting the storage group is essentially the same as the process of joining. In the clockwise direction, there are three PC in the storage group, X, Y, and Z. If Y applied to get out of the loop， the information that was stored in the Y would be copied from X to Z, and the information that was stored in the Y would be copied from Z to X, and then, the data in Y-master backup is copied to the first device from z of the loop.

4. Security Design of the Cloud Storage

Cloud storage is a hot topic in industry and academia in recent years, and the security problems of the cloud storage would have been under scrutiny. The AES(Advanced Encryption Standard) algorithm is simple and the encryption of AES is fast. However, AES has problems with key allocation and confidentiality management. There is no need for secret allocation of keys in asymmetric encryption algorithms, and at the same time the security of the keys is easier. In addition, user authentication and digital signatures can be achieved through the RSA algorithm. To make full use of the advantages of the AES and RSA algorithms, a solution that combines AES and RSA encryption algorithms for mobile terminal cloud storage security design is proposed in the paper.

4.1 Encryption and decryption design for Mobile terminal

After the data is encrypted through AES, then the encryption key is encrypted by RSA. The encrypted key message are binded to the encrypted data. The message will then be stored in each node of the HDFS. This approach can improve the storage efficiency of the mobile cloud storage system and also solve the key distribution of single key cryptography. When the data on HDFS is read and downloaded,

the AES key is extracted from the cryptograph, then the decryption is obtained through the user's private key, finally, the document is declassified and plain text is obtained. The process is as follows.

1) During data encryption uploads, users log in to the cloud storage system, Sending data requests to HDFS and encrypting the transfer. At the same time, a 128-bit AES encryption key is generated by the client random key generator.

2) On the mobile side, the data that needs to be transmitted is encrypted with the AES key, and the cipher text would be got.

3) The encryption KEY of the file is encrypted through the 128-bit RSA public KEY, and then the key cipher is obtained.

4) The key cipher are bound to the file cipher, In accordance with the file cipher, the file is stored in the HDFS file system with the corresponding tag bit and data length identification.

5) When the data is downloaded from an HDFS system in the cloud, the data are decrypted and downloaded. After the data are obtained, which are transferred from the HDFS system to the mobile end. The first bit of data is judged first by the system, and if it is zero means that the data is in plain text, the data are restored after removing the tag bits. On the contrary, if it is 1 means that the data is a cipher, it should be decrypted.

6) First, extract a 128-byte AES key cipher from the data, AES plaintext key are got by decrypted the user's personal RSA private key.

7) The Cipher part of the stored file cipher is decrypted by AES through the AES key, and then the stored file plaintext is got

4.2 The data storage format of the cloud storage system

The data for the cloud storage system includes two storage formats, which are plaintext storage and crypto text storage. The storage format is shown in figure 6

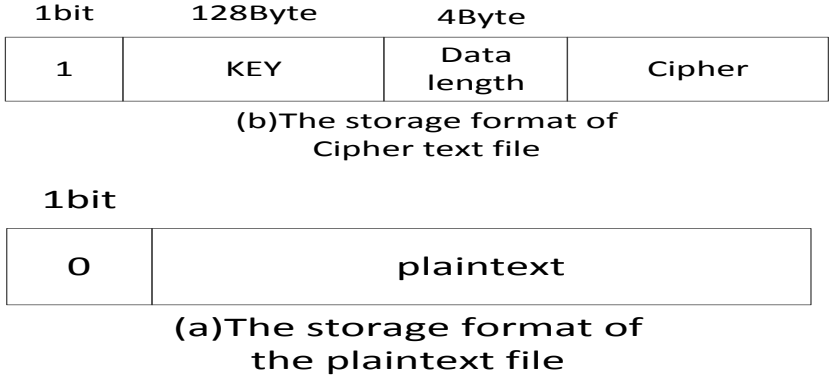

Figure.6 The Storage Format of the File

If the first digit in the storage format is 0, which means that the data is stored in plain text, on the contrary, if it is 1 means that the data is a cipher. The 128 bytes need to be added before the cipher text to store the AES cipher key encoded by RSA when the data is stored in cipher text. The field for the valid

length of the data is 4 bytes, and the field of Cipher represents the encrypted text by AES.

Each 1024-byte byte stream is encrypted with AES and is converted into a 1040-byte cipher stream. So the length of the Cipher section grows relative to the original plaintext data.

4.3 Security analysis

The security design of the mobile cloud storage is implemented in a combination of the AES algorithm and the RSA algorithm. The security is analyzed separately for the AES and RSA algorithms. The security of AES is analyzed through exhaustive attack, the differential attack, and interpolation attack when the AES key don't be known.

1) Exhaustive attack: The average complexity of the exhaustive key is 2k-1 AES encryption, in which K is the length of the key. For the 128-bit key in this scheme, 2127 times of AES encryption are required and the calculation is very large, and obviously this method of attack is invalid.

2) Differential attack: The wide trajectory strategy adopted by the AES algorithm can effectively resist differential attacks. The prediction probability of the difference trajectory is between 2 and 150 after four rounds of transformation, it's between 2 and 300 after the eight transformations. So, enough times can be identified to make all the differential trajectory less than 1/2n-1, n is the number of blocks. This makes the difference attack fail.

3) Interpolation attack: F256 domain in AES algorithm, expansion is shown as blow:

63+8FX127+B5X191+01X223+F4X239+25X247+F9X251+09X253+05X254

Because the expansion is complex, the attack is also invalid.

Through this analysis, The AES algorithm is better immune to known attacks the unknown AES key, From the analysis above, we can learn that the AES algorithm is better immune to known attacks in case of not knowing the AES key. Also, the user's files in HDFS are stored in a certain size, and the security of the system can be further enhanced. Therefore, the main issue of security is the security of the AES file encryption key. How to manage and store file encryption keys is the key to determining the security of the solution.

In the design scenario presented in this article, Technology of one-timepad is used for file transfer storage. Each data stored has a different AES key, and the AES key is transparent to the user. In addition, the AES key for each file is encrypted by using the RSA algorithm. The encrypted AES key is bound to the file cipher and then stored in HDFS. The user must take care of his RSA private key throughout the process. The above encryption is done on the mobile side, which implements the file's cryptographic transfer and cryptographic storage.

And then the security of RSA is analyzed in detail. The security of RSA depends on the large integer factorization. The difficulty of attacking the RSA system is the difficulty of the large integer factorization. The Schroeppel algorithm is a better factorization algorithm, and which is often used to analyze the problems of the large integer factorization. The number of operations required in decomposing the factor of decimal number n with different length by using Schroeppel algorithm. The number of decomposing operations is shown in table IV, in which the factor of decimal number n with different length is decomposed by using Schroeppel algorithm.

Table 4 The Number of Operations of Decomposition Factor By Using the Schroeppel

digits of Decimal number n	50	100	200	300	400
the number of operation	1.3×10^{10}	2.4×10^{15}	1.1×10^{23}	1.4×10^{29}	2.6×10^{34}

The longer the length of n is, the more difficult the factorization is in the RSA algorithm. For every ten bits of binary that are added, the time of decomposition is going to be doubled. And then the harder it is to decode the password, the more the strength of the encryption will be. A key length of 512, 1024, 2048 bit are often selected in RSA.

In the cloud storage security technology solution designed in this paper, the 16-byte AES key is the object of RSA encryption, the system will be highly secure once the key of 2048 bit is selected.

Assuming that the reliability of the cloud storage system is A. The time of encryption through different encryption algorithms is At, the encryption time At is reversed first， and after the normalization processing， Aj is got from At. The transfer rate of a file with the same size after the normalization processing is Ak， n is the copynumber. The reliability model of the system is:

$$A=[1-(1-A_j)^n][1-(1-A_k)^n] \tag{1}$$

It can be concluded through the analysis of the reliability model, when the value of Aj and Ak are more closer to 1, and the number of n is more larger, the cloud storage system will be more reliable and with higher security.

5. Experiment and Result Analysis

The Hadoop cluster built in this article consists of one namenode server and three datanode servers. The client submits the data through the namenode server. The configuration of the four datanode servers is: Intel dual core CPUG630@2700MHz; network environment for NetLink BCM5784 Gigabit Ethernet; The version of Hadoop is 1.0.3; The version of Linux is ubuntu 11.10; And JDK 1.6.0 _17 is used. The configuration of Client is Pentium/(R) Dual-core CPU E5200@2.5GHz. The mobile terminal is Huawei y635-cl00, Qualcomm Snapdragon CPU and 1 GB memory are used. A certain number of storage nodes are simulated by using CloudSim simulation. The data response tests are performed on file upload, file copy and file movement, for large files and small files respectively.

System is tested by using a smart mobile terminal. Experimental results demonstrate that the page is properly displayed, and the response time of the login page is basically completed within two seconds. The percentage of the response time for the transaction is shown in figure 7.

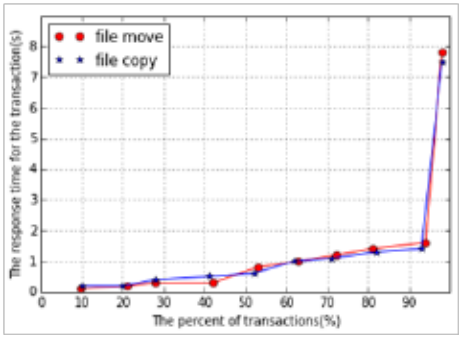

Figure.7 The Percentage of the Response Time for the Transaction

Through the analysis of figure 7, it can be known that 94 percent of transactions in a mobile cloud storage system can be implemented quickly within two seconds. The experimental results show that the system responds fast. The average response time of the transaction is obtained from the diagram.

Although one of the transaction response time is longer, but the response time for most other transactions is acceptable. When this happens, it is thought that the performance of the mobile cloud storage system is better.

After the encryption and decryption mechanism is introduced in the security of the mobile cloud storage, the security of the cloud storage is improved effectively.

There are two questions to be considered: The impact of encryption and decryption on file speed; The impact of encryption and decryption on the performance of the client host.

In the mobile cloud storage security technology proposed in this article, the method of encrypting and decrypting the file on the mobile end is used. The length of the file will change after the file is encrypted into a cipher file. According to the analysis of the file storage format in 4.2. The header of each cipher file needs to be added 128 bytes to store the AES secret key. In addition, when the AES file is encrypted, each 1024 bytes text will be encrypted and then changed to 1040 bytes cipher. In conclusion, after encrypting， the length of the cipher file is about 1.56 percent more than the file. The namenode and datanode in HDFS may be caused an additional cost of about 1.56 percent after encrypting.

For clientnode in HDFS, the time spent on file encryption and decryption is increased, and the performance is reduced eventually.

The effect of file encryption and decryption on the whole file transfer rate is mainly in two aspects: The time required to encrypt and decrypt the transmission file by using AES; The time spent on encrypting and decrypting the AES key by using RSA.

The experimental data are listed in table 5, which includes the time spent on encrypting the different sizes or different types files by using AES and the time spent on transmitting the file in HDFS.

Table 5 Time Comparison on AES Encryption and Decryption

File size (M)	File type	AES encryption (ms)	HDS upload (ms)	AES decryption (ms)	HDS download (ms)
3.07	pdf	1050	2685	370	2800
3.22	MP3	1178	2600	478	2830
23.8	mkv	3238	5930	2648	6290
25.8	doc	3140	5260	2163	6400
166.518	rmvb	23830	46400	16500	42460

The time that AES KEY is encrypted by using RSA are also tested. By using RSA the 128bit AES key was encrypted for an average time of 499ms and decrypted for an average time of 32ms. It can be concluded from the above test data, the time spent on encryption or decryption by using AES is regardless of the file type. The time comparison on AES encryption and decryption is shown in figure 8.

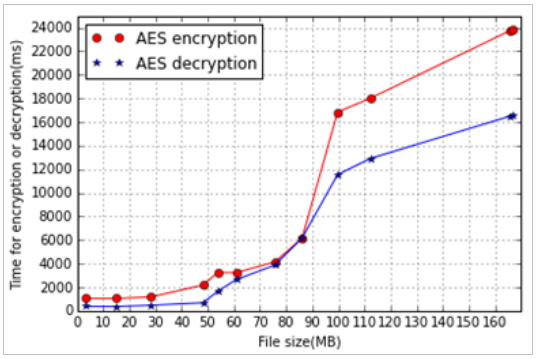

Figure.8 Time Comparison on AES Encryption and Decryption

The same size files are encrypted by different AES, RSA, or RSA + AES algorithms and then the encryption time is different, as shown in table X.

Table 6 Time Comparison for Different Algorithm Encryption

File size (M)	AES encryption (ms)	RSA encryption (ms)	AES+RSA encryption (ms)
3.04	1048	753	770
23.15	3230	2889	2901
167.58	23820	23598	23612

In the solution proposed in this paper， the time of AES key encrypted or decrypted by using RSA is relatively short. File transferring have little impact on total time loss and user experience. It may takes a relatively long time to encrypt files by using the AES, which cause a significant additional time overhead for HDFS. However, the encryption time that AES combined with RSA for encrypting the file was not significantly increased compared to RSA. Besides the impact on overall transmission rates, the impact of encryption and decryption on mobile performance is also important. The 167.58 MB file in table VI is the test case. Table 7 and table 8 are the test result.

Table 7 The Performance of the Mobile End Upload the Data

type of test	utilization rate of Mobile CPU (%)	transmission rate (Mbps)	utilization rate of Mobile /transmission rate
Raw data	16.436	3.57020	4.603663
After the encryption	38.432	2.36015	16.283711

Table 8 The Performance of the Mobile End Download the Data

type of test	utilization rate of Mobile CPU (%)	transmission rate (Mbps)	utilization rate of Mobile /transmission rate
Raw data	14.681	3.90135	3.763056
After the encryption	35.221	2.76147	12.754439

The ratio of CPU occupancy to upload speed is shown in table 7, which the data on the mobile side are tested before the encryption and after the encryption respectively. The ratio of CPU occupancy to download speed is shown in table 8, which the data on the mobile side are tested before the decryption and after the decryption respectively. It can be known from the table 7 and the table 8, if the encryption and decryption mechanism are used for HDFS transmission, then the CPU utilization will be increased by an average of 22% ~ 25% and the overall file transfer rate will be reduced by 30% ~ 35%. As we can see, when the encryption and the decryption mechanism are used, more than three times the performance loss can be caused on the mobile end side.

Although the encryption mechanism and decryption mechanism will cause some performance loss to the mobile end, the confidentiality of the data can be guaranteed. So it is acceptable from the perspective of user data security. Lots of time are spent during encryption or decryption, which can cause a drop in transmission rates. Two points of improvement are proposed for this situation:

1) The user can choose whether or not to encrypt the file. Important files are usually in the form of text or images, which are generally small and can choose to be encrypted. However, some larger files, such as video, audio, etc., users can choose whether to encrypt or not. The less important files are stored in plain text, which can improve the access efficiency of the files.

2) For cloud storage users, the file transfer and stored procedures are transparent. Therefore, the transport encryption buffer can be set up on the client for large file transferring. After the transfer request is submitted by the user, the file's decryption and transfer operations are implemented in the background. After the transfer is completed, only the prompt message can be given on the mobile end, which can improve the user experience.

Using the reliability model formula of the cloud storage system proposed in 4.3, combine the time required for processing the same size of file in table 6, when a different algorithm AES, RSA, or RSA + AES is used, the encryption time required for encrypting file, after that the encryption time is reversed, Aj then be got after the encryption time is normalized. In the same way, after normalizing, transfer rate Ak is got. If both Aj and Ak are closer to 1, and the number of copies in the cloud storage system is larger, then the reliability of the system is higher. According to the above analysis, the data in one hour is sampled continuously, combined with the data in table 6 and table 7, the reliability contrast diagram for the cloud storage system is shown in figure 9.

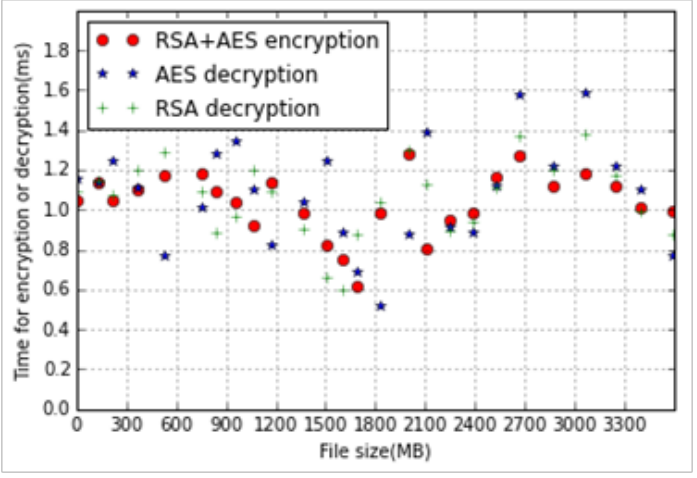

Figure.9 The Reliability Contrast Diagram

It can be know the reliability of the system through different algorithm by comparing the data in figure 9. By using the RSA encryption, the system's reliability values are almost maintained at 1. The reliability of the system is relatively high. That is, it has little impact on file transfer and user experience by using RSA encryption. It may takes a relatively long time to encrypt files by using the AES. And the reliability of the system is very jitter. It is shown that the system reliability is low with AES encryption. However, the encryption time that AES combined with RSA for encrypting the file was not significantly increased compared to RSA. The value of the system reliability is consistent with the use of RSA, which can be maintained around 1. It is concluded that the system is relatively reliable by using AES and RSA encryption.

Through the simulation experiment, it is verified that the mobile cloud storage system has a good user experience. It is also verified that the multi-copy mechanism of the cloud storage system can effectively improve the efficiency of the cloud storage system. When a mobile terminal makes a request, the closest copy is selected and then the time can be saved effectively.

In the solution presented in this paper, the encryption and decryption performed by the mobile side has the following characteristics: transport security and storage security of the user data are guaranteed. The mobile side finish the encryption before calculating the checksum, so the encryption will not break the HDFS data integrity check mechanism. In the entire distributed file storage system, the encryption and decryption are scattered to the various mobile devices. While this will cause some performance damage to the mobile, there is no additional performance penalty for namenode and datanode.

The solution enables the entire distributed file system to be protected by data privacy, and there is no significant performance penalty for multi-user, large access, and file access. In the current version of HDFS, the mobile user identity is given by the host operating system. The user authentication mechanism for the HDFS mobile is also very flawed. In the scheme proposed in this paper, the RSA algorithm and its public key library are introduced, which can create the prerequisite for solving the kind of problem.

6. Conclusion

Cloud storage security technologies for mobile terminals proposed in this paper, different storage policies are used for file in different size. Considering the storage efficiency of mobile, the load balancing effect of cloud storage system is improved, and the stability and extensibility of cloud storage system is improved.

In addition, in order to make the cloud storage system has higher reliability. According to the characteristics of HDFS data input output and integrity checking, the AES algorithm is used to encrypt the files uploaded by the user on the client side of HDFS. This ensures that the confidentiality of mobile user data in the cloud storage system. By using the RSA algorithm, the security of the AES key is guaranteed, and the issue of distribution and management of the AES single key password can be resolved. The two storage formats of the cloud file are designed to implement the user's own choice, reduce the number of copies, and ultimately improve the storage efficiency of the mobile cloud storage system. Finally, the reliability of the mobile cloud storage security technology scheme is verified through a series of simulation experiments.

The mobile cloud storage security technology scheme proposed In this paper has better security and reliability. But there are still many problems that have not been solved, and further research is needed.

In order to improve the user experience by setting up the encryption buffer. At the same time, PKI technology can be used to implement CA authentication and digital signatures for HDFS users to further enhance HDFS security.

Foundation Item

The Industrial research project of Science and Technology Department of Shaanxi Province(Grant No. 2016KTZDGY4-09)

References

[1] IDZIOREK J, TANNIAN M, JACOBSON D. Attribution of Fraudulent Resource Consumption in the Cloud [J]. 2012 IEEE Fifth International Conference on Cloud Computing, 2012: 99-106.

[2] TSAI T, THEERA -AMPORNPUNT N, BAGCHI S. A study of soft error consequences in hard disk drives [J].IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012), 2012:1-8.

[3] Schmuck F B, Haskin R L.GPFS: A shared-disk file system for large computing clusters[C]//Proceedings of the Conference on File and Storage Technologies, January 28-30, 2002:231-244.

[4] Namjoshi J, Gupte A. Service oriented architecture for cloud based travel reservation software as a service[C]// Proceedingsof the 2009 IEEE International Conference on Cloud Computing(CLOUD’09), Bangalore, India, Sep 21-25, 2009.Los Alamitos, CA, USA: IEEE Computer Society, 2009:147-150.

[5] Goth G. Virtualization: Old technology offers huge new potential [J].IEEE Distributed Systems Online, 2007,8(2).

[6] BOWERS K D, JUELS A, OPREA A. Proofs of retrievability : theory and implementation [J]. Proceedings of the 2009 ACM workshop on Cloud computing security (CCSW′ 09), 2009:43.

[7] ARMBRUST M, STOICA I, ZAHARIA M, et al. A view of cloud computing [C]. Communications of the ACM , 2010, 53(4):50.

[8] MELL P, GRANCE T. NIST Special Publication 800 -145:The NIST Definition of Cloud Computing [J]. National Institute of Standards and Technology, 2011.

[9] KARAME G O, CAPKUN S, MAURER U. Privacy -preserving outsourcing of brute-force key searches[J]. Proceedings of the 3rd ACM workshop on Cloud computing security workshop (CCSW′ 11): 2011:101.

[10] SCHIFFMAN J, MOYER T, VIJAYAKUMAR H, et al. Seeding clouds with trust anchors [J]. Proceedings of the2010 ACM workshop on Cloud computing security workshop (CCSW′ 10), 2010: 43