Research on Integrity Protection of Data for Multi-server in the Cloud Storage

Guangjun Song[1(✉)], Dandan Lu[1], and Ming Li[2]

[1] School of Mathematics, Physics and Information Science, Zhejiang Ocean University, Zhoushan 316022, Zhejiang, People's Republic of China
song_gj@126.com

[2] College of Computer and Control Engineering, Qiqihar University, Qiqihar 161006, Heilongjiang, People's Republic of China
lcrb406@163.com

Abstract. Based on existing cloud storage services and remote file synchronization algorithm analysis, a secure cloud storage integration solution is proposed. Its design makes the realization of a cloud-storage-based personal encryption file synchronization and backup system possible. Users can simultaneously manage multiple cloud storage accounts, so that they can synchronize multiple folders and backup at any time. The system can correctly synchronize and backup personal data according to users' needs. Taking advantage of the MD5 algorithm to make encrypted backup file safer, the new mode can prevent the illegal change and disclosure of personal files after synchronization, thus integrity protection of data is achieved, and it becomes easier to manage cloud storage accounts with different servers. Experiments prove the validity and reliability of the system.

Keywords: Cloud storage · MD5 · Cloud sync and backup
Integrity protection

1 Introduction

Cloud computing can conveniently provide users with available resources such as network storage, applications, services and so on. Of them, cloud storage technology offers users a certain capacity of storage space, so that users can upload their data files to the cloud, and they can check, download or sync these files on other terminals or mobile terminals [1, 2]. In recent years, Cloud storage technology has achieved rapid development. To make it easier for users to manage cloud storage accounts of different servers, some cloud storage management platforms have emerged, for example, CarotDAV, Otixo, MultCloud, ZipShare, etc. These cloud storage management softwares provide convenience for users to take full advantage of cloud storage resources. However, data security of cloud storage system has always been a most concerned problem for cloud storage users [3, 4]. Though the cloud storage technology of transfer encryption and storage encryption or other measures have been taken by SSL and AES, etc., data loss, illegal change and other safety problems still exist in cloud storage [5, 6].

G. Sun and S. Liu (Eds.): ADHIP 2017, LNICST 219, pp. 520–528, 2018.
https://doi.org/10.1007/978-3-319-73317-3_61

These problems have not been satisfactorily resolved by the above-mentioned cloud storage management platforms. Therefore, how can security backup files optionally as required is one of the most pressing problem for cloud storage management platforms. At present, researchers have proposed a variety of solutions to integrity verification of the backup files in cloud [7–10]. However, these fail to meet the actual security needs.

Based on the deficiencies of existing personal cloud storage services, a cloud storage integration solution is presented. That is, users can freely add any personal cloud storage accounts to the system, and can also simultaneously manage several cloud storage accounts of different servers. A backup model combined with compression and MD5 encryption is proposed, which adopts technologies like single or bidirectional selective backup, timing and cycle synchronization. MD5 algorithms are used to generate digital fingerprints of files to identify their tiny changes. Thus, data integrity protection is achieved, the problem of files storage security in cloud computing is better solved.

The paper is organized as follows: The structure and function design of the system are discussed in Sect. 2. The Technology of encryption and Process for backup files are put forward in Sect. 3. The results of simulation and analysis are presented in Sect. 4. Section 5 includes conclusion and further study.

2 System Design

The cloud-storage-based personal encryption file synchronization and backup system is located between open programming interfaces of cloud storage and data access layer.

2.1 System Function Design

The system contains an account database and a client that can run on multiple platforms, mainly including the following functions.

(1) File synchronization: to analyze similarities and differences in local folders and cloud ones, and keep files in the two folders consistent without missing any files.
(2) Account management: users control their multiple cloud storage accounts. They can login, unbind and complete additions and deletions operation of multiple cloud storage services, and get account information of that service. The integration of cloud storage services is achieved by the management of multiple cloud storage accounts.
(3) File backup: to create backup files by compression and MD5 encryption, and simultaneously record this backup time point that could be used as a search point to restore files when needed. The current version of folders is stored in cloud, and the version of the backup file can be retrieved and decrypted through encryption keys so that the model can solve the problem of no backup restore points. By generating digital fingerprints through applying the MD5 algorithm to all files, you can detect whether any changes have been made to this version of the backup files.

(4) Log: the complete log records and content change records are established, including time, file names and type of change.
(5) Common settings: network and system settings, including network proxy settings, bandwidth settings, HTTPS secure transmission mode start using and whether startup.

2.2 System Process Flow

The system implementation process is as shown in Fig. 1. Firstly, users select a cloud storage services platform, login account management module, and create a cloud storage client. Then the system and network parameters of cloud storage client are set in common settings module, and API (Application Programming Interface) control module is called through file synchronization module and simultaneously update local files and cloud files so as to achieve automatic storage and synchronization on multiple cloud storage service platforms. Secondly, file backup module periodically carries on compression, encryption and retrieval for local folder to be backed up, meanwhile checks whether the backup file has been changed. Finally, the performance of API control module is real-time recorded by log module, thereby realizing the security protection for backup files stored in the cloud.

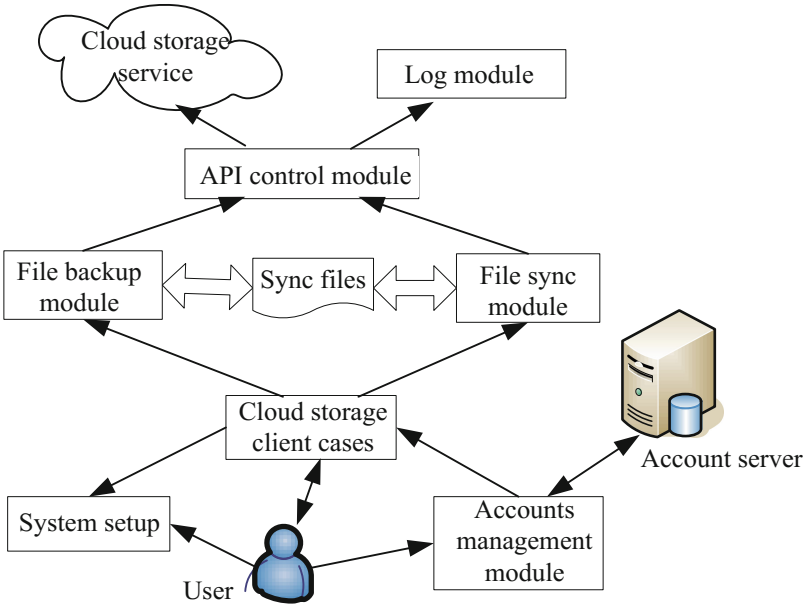

Fig. 1. The system implementation process

3 Encryption Technology and Process

Nowadays, as information security is regarded as an important problem by the society, Cryptography is paying more and more attention. Cryptography is always companioned with Hash functions, which is a kernel of modern Cryptography.

3.1 MD5 Encryption

The full name of MD5 is Message-Digest Algorithm 5 [11], evolving from multiple algorithms, such as, MD2, MD3 and MD4.

The one-way hash function is also called the Hash function [12], which is the core of the MD5 algorithm. Hash function is an important branch of cryptography. It is a non-reversible and one-way cryptosystem which transforms the input of arbitrary length into fixed-length output. Let M be a message with arbitrary length, h = H(M) means the evaluation of M with a one-way hash function, a fixed length value of h is obtained, where h is called the hash value of M. If M is divided into L packets, it is expressed as M_0, M_1, … M_{L-1}, the length of each packet is m bit, and if the length of the last packet is not enough, it needs to be padded with zeros. A compression function f is reused in the algorithm. It has two inputs, one is the n-bit output h_{i-1} of the previous round and the other is the m-bit input packet M_{i-1}, the output of the compression function f is n bit h_i, and it is the next round of input too. At the beginning of the algorithm, you need to specify an initial input value of n bits IV. The output value with the fixed length in the last round is the final hash value of the entire message. The whole algorithm can be expressed as follows:

$$h_0 = IV \tag{1}$$

$$h_i = f(h_{i-1}, M_{i-1}),\ 1 \le i \le L \tag{2}$$

$$H(M) = h_L \tag{3}$$

As Hash is a one-way function, that is we can very easily calculate H from M, while it is difficult to calculate M from the known H. Therefore, it can be only used to encrypt data, yet there is no way to decrypt the encrypted data. MD5 is one of the most widely used Hash algorithm currently, which can convert an arbitrary length byte into a fixed-length string of 128 large integer (message digest), namely H = hash(M), where H is called M's hash value. It is typically applied in two aspects, one is to encrypt user's password by taking advantage of its irreversibility, so as to maintain the security of the system, and the other is to verify the integrity of information. Namely, MD5 takes the entire file as a large text message, generating a unique MD5 message digest. In the process of this document transmission, as long as the contents of the file occur any form of change, and message digest will also change through MD5 computing of that document, thus it can be determined that the received file is not the original one. This design mainly uses the latter. Read reference for a complete description of the MD5 [11].

3.2 System Implementation Steps

Assume that a user has an account of N ($N \geq 1$) cloud storage service platform, and file backup module in the system adopts MD5 encryption method to encrypt files and generate Message-Digest, uses zip file encryption method to compress files, and uses a MySQL as an account database. System implementation process is as follows:

Step 1. Among N cloud storage service platforms, users can choose any one to login account management module by using his account information in the cloud storage services platform. A cloud storage client is created, and the account information selected in the cloud storage services platforms is stored in the account database.
Step 2. Users make use of common setting module to set the parameters of system and network of the cloud storage client. After the completion of setting the parameters of system and network of the cloud storage client, the system calls the file synchronization module.
Step 3. The methods and directories to be synchronized are selected from both local folders and the cloud in files synchronization module, the local files and cloud files are synchronous updated by the API control module, as shown in Fig. 2:

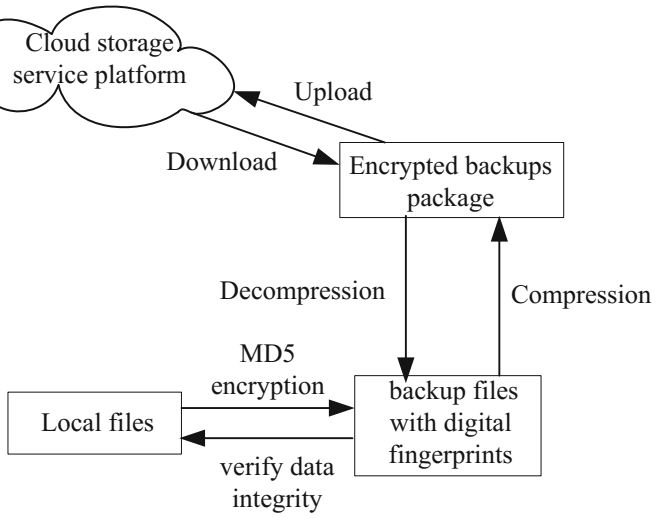

Fig. 2. Files backup flowchart

Step 4. The local files folder to be synchronized in step (3) are periodically carried on compression, encrypted and accessed, and checked the integrity of the files by file backup module.
Step 5. The performances of API control module from step (1) to (4) are real-time recorded by log module.

4 System Tests and Analysis

The correct BibTeX entries for the Lecture Notes in Computer Science volumes can System test platform for PC, and the cloud storage service API of Kanbox is accessed through the campus network. Testing machine configuration is: Intel (R) Core (TM) i7 CPU, 2.80 GHz frequency, 4.00 GB RAM, Windows 7 Home operating system.

4.1 Test on Backup Files

First, the backup efficiency of commonly used files is tested with the same type but different sizes. According to practical experience of backup files, generally speaking, the text file is the most frequently used file type in everyday. For example, the size of txt file is generally ranged 100 KB to 100 MB. Since backup time for less than 1M of file is mostly within 200 ms, so we select a series of files gradually increased from 1M to 100 MB. Those txt files make up data sets that will be used for system performance tests. Figure 3 shows the time of backup system spending on the same type of with different sizes.

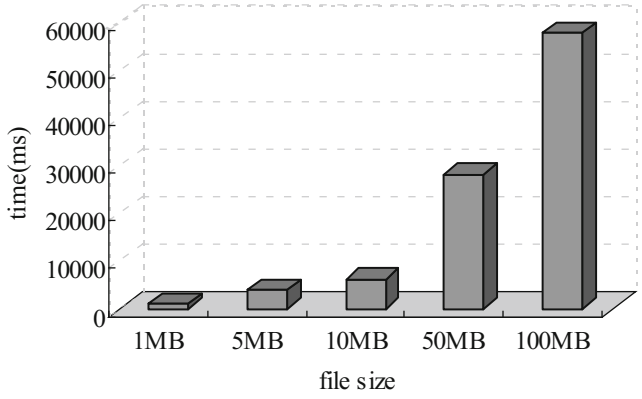

Fig. 3. Backup efficiency of text files

As we can see from Fig. 3, as the file size increases, the longer time it takes to backup system files. Mainly because of larger size, it takes more time for compress and upload larger files, but the backup time of frequently used files between 100 KB–100 MB is within tolerable range.

We distinguish backup files of different types and sizes. Frequently used file types are txt, doc, ppt, pdf, mp3, jpg, etc., A series of files of different types, whose range of size gradually increased from 1M to100 MB, are selected to make data sets that will be used for system performance tests. Specific file sizes and types are as shown in Table 1.

Test file sets that are selected in Table 1 have certain representativeness. Files larger than 1 GB (rmvb files and other multimedia files) are less used at present, so they

Table 1. Test sets of different types of files

File size (M)	File types				
1	.txt	.ppt	.doc	.jpg	.rar
5	.txt	.mp3	.doc	.jpg	.pdf
10	.txt	.pdf	.doc	.jpg	.wmv
50	.txt	.ppt	.doc	.jpg	.rmvb
100	.txt	.ppt	.doc	.jpg	.rar

will not be included in system tests. Figure 4 shows minimum, maximum and average time that system takes to backup files of different sizes and different types.

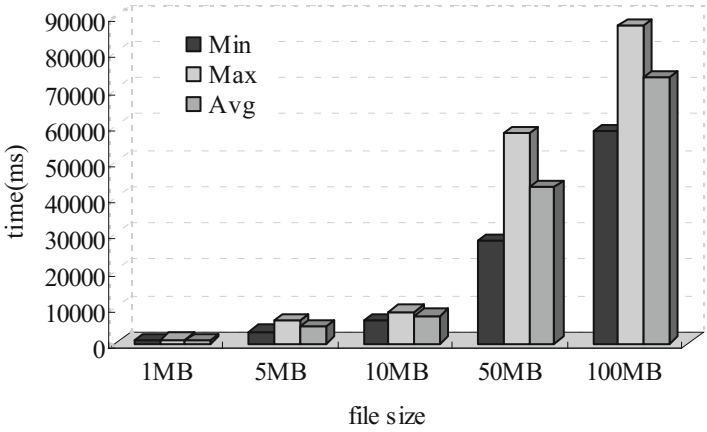

Fig. 4. Backup efficiency of different types of files

Experimental results show that with the increase of file sizes, file system backup time of different types of files presents the overall trend of gradually increase. To backup the same size files in different types, the time it takes fluctuation within a fixed range. That is mainly because it takes different time to compress different file types. Through the analysis of test results, we can see that system backup function is normal.

4.2 Synchronization Performance Comparison on Storage Services

We use resource monitor for network monitoring on Kanbox. The time comparison of certain operation is completed through Kanbox client and the system client respectively, then the performance of the system can be evaluated.

First of all, compare the first round of synchronization, namely initialization, including upload and download. Select 100 synchronize folders, and each one is about 60 MB and contains multi-level files and folders. Followed by adding and deleting file operation, select 100 operating objects, and all of which are about 20 MB files in size.

Finally, select 100 operating objects, all of which are 5 MB files in size to test update operation (Table 2). The average time spent on each operation test is gotten, comparison results are shown as in Fig. 5.

Table 2. Comparison of the performance of different operations

Client	Initialization	Add file	Delete file	Update file
Design model	45346 ms	14124 ms	975 ms	2618 ms
Kanbox	52000 ms	13000 ms	1000 ms	2000 ms

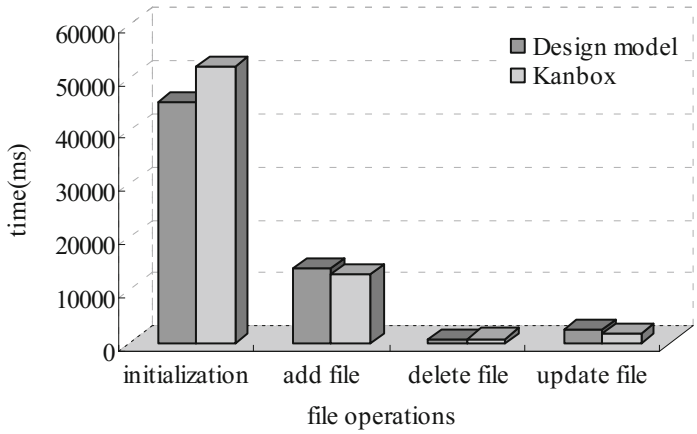

Fig. 5. Comparison of synchronization performance

The test results show that basic operation time of this system and Kanbox are much the same, but the system has a large performance advantage in initialization, namely the first round of synchronized strategy.

5 Conclusions

According to currently existing problems of cloud storage management products, the paper proposed a secure cloud storage integration solution. Its design makes the realization of a cloud storage-based personal encryption file with digital fingerprints synchronization and backup system possible. The system can correctly synchronize and backup personal data as needed, and ensure the safety of files. Through an open API control for cloud storage services, it allows users to simultaneously manage multiple cloud storage accounts and carry on multiple folder reliable synchronization and backup at anytime. The new backup mode uses MD5 algorithm to generate digital fingerprints for backup files, which can effectively prevent changes and tampering of personal files, thus integrity protection of the backup files are achieved. Experimental

results show that the system enables users to achieve multiple cloud storage account integration simultaneously and the purpose of synchronizing storage and security management. However, this system synchronization strategy for optimization of conflict processing, mobile terminal services and other issues still need to be improved, which is also an improvement goal of the next step.

Acknowledgments. This work is supported by the National Nature Science Foundation of Zhejiang Province, China, No. LY16F020014, the National Nature Science Foundation of Heilongjiang Province, China, No. F201204 and the Education Department of Heilongjiang Province, China, No. 12531756.

References

1. Zeng, W., Zhao, Y., Ou, K., et al.: Research on cloud storage architecture and key technologies. In: 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, pp. 1044–1048. ACM, New York (2009)
2. Mell, P., Grance, T.: The NIST definition of cloud computing. J. Natl. Inst. Stand. Technol. **53**, 50–57 (2009)
3. Li, H., Sun, W.H., Li, F.H., et al.: Secure and privacy-preserving data storage service in public cloud. J. Comput. Res. Dev. **51**, 1397–1409 (2014). (in Chinese)
4. Xue, M., Xue, W., Shu, J.W., et al.: A secure storage system over cloud storage environment. J. Comput. **38**, 987–998 (2015). (in Chinese)
5. Julisch, K., Hall, M.: Security and control in the cloud. Inf. Secur. J.: Glob. Perspect. **19**, 299–309 (2010)
6. Feng, D.G., Zhang, M., Zhang, Y., Xu, Z.: Study on cloud computing security. J. Softw. **22**, 71–83 (2011). (in Chinese)
7. Fu, Y.Y., Zhang, M., Chen, K.Q., et al.: Proofs of data possession of multiple copies. J. Comput. Res. Dev. **5l**, 1410–1416 (2014). (in Chinese)
8. Wang, H.Q.: Identity-based distributed provable data possession in multicloud storage. IEEE Trans. Serv. Comput. **8**, 328–340 (2015)
9. Liu, C., Yang, C., Zhang, X.Y., et al.: External integrity verification for outsourced big data in cloud and IoT: a big picture. Future Gener. Comput. Syst. **49**, 58–67 (2015)
10. Wang, H.F., Li, Z.H., Zhang, X., et al.: A self-adaptive audit method of data integrity in the cloud storage. J. Comput. Res. Dev. **54**, 172–183 (2017). (in Chinese)
11. Rivest, R.: The MD5 Message-Digest Algorithm. RFC1321, April 1992
12. FIPS PUB 180-1. Secure Hash Standard, April 1995