User-Controlled Encrypted Data Sharing Model in Cloud Storage

Yuezhong Wu[1,2], Shuhong Chen[3,5](✉), Guojun Wang[3],
and Changyun Li[2,4]

[1] School of Information Science and Engineering, Central South University,
Changsha 410083, China
yuezhong.wu@163.com

[2] School of Computer Science, Hunan University of Technology,
Zhuzhou 412007, China
lcy469@163.com

[3] School of Computer Science and Educational Software,
Guangzhou University, Guangzhou 510006, China
shuhongchen@gzhu.edu.cn, csgjwang@gmail.com

[4] Intelligent Information Perception and Processing Technology,
Hunan Province Key Laboratory, Zhuzhou 412007, China

[5] School of Computer and Communication, Hunan Institute of Engineering,
Xiangtan 411104, China

Abstract. Cloud storage services provide us convenience for storing and sharing vast amounts of data by its low cost, high scalability and other advantages while it brings out security risks as well. A user-controlled encrypted data sharing model in cloud storage (UESMCS) is put forward hereby. It preprocesses user data to ensure the confidentiality and integrity based on triple encryption scheme of CP-ABE ciphertext access control mechanism and integrity verification. Thus, the reliability and safety for data sharing can be achieved provided the trustworthy third party being brought in. The experimental results show that UESMCS ensures data security in cloud storage services platform and enhances the operational performance for data sharing. The security sharing mechanism perfectly fits the actual cloud storage environment.

Keywords: Cloud storage · Data confidentiality · Ciphertext access control

1 Introduction

A substantial number of people, in their learning, working and living store, share their information through an open network. Cloud storage services, a new form of network application model, emerged and gathered numerous different types of storage devices through the application of software co-functioning to realize external data storage and business access services through using clustering applications, grid technology and distributed file systems and other functions, ensures data security and saves storage space effectively [1–3]. Users can store their data in remote cloud storage stored centers, access on-demand and user-friendly for enterprises to save costs, improve availability and reliability. However, corporate users lost a fundamental physical

G. Sun and S. Liu (Eds.): ADHIP 2017, LNICST 219, pp. 12–20, 2018.
https://doi.org/10.1007/978-3-319-73317-3_2

control for their data stored in the cloud, which will doubt their confidentiality and integrity of the data, and inevitably raises its concerns about data security and privacy aspects. There are two points about the reason: First, the cloud service providers are facing a wide range of internal and external attacks following malicious enemies deleting or destroying user data. Second, the cloud service providers may be dishonest, they may seek to save their reputation or interests while trying to hide the information of theft or destruction of the data stored in the cloud. Thus, based on the complexity of the dynamic and open cloud storage environment and other features, users rely entirely on untrusted cloud storage providers and other data storage and management factors, how to securely share data in the open cloud storage environments is a problem need to be solved for cloud storage applications.

To have these problems worked out, and to guarantee a safe cloud storage service of data sharing for general users or business users, a secure storage for cloud sharing model is proposed based on CP-ABE technology. It is functioning actively by user and systems with triple encryption to secure user-controlled access for the data in cloud storage.

The main contributions of this paper are:

(1) User-controlled encrypted data. Based on symmetric encryption, CP-ABE and MD5 technology, it is triple encrypted and integrities checking for the data. It promises access permissions of the encrypted data by user-controlled, ensuring the security of data stored and shared in the cloud storage.

(2) Trusted third party. The introduction of a trusted certification authority as a third party authorized purposed to store key information, monitor and audit user access data to achieve security data sharing.

The remainder of this paper is organised as follows: Sect. 2 introduces the terminology and the related work. Symbol description is in Sect. 3, and we also introduce the proposed the data sharing model and application scene. In Sect. 4, we detail the security encryption and algorithm design. In Sect. 5, we present the results and analysis for the experiment. Finally, we conclude in Sect. 6, and briefly touch on the future work.

2 Related Work

Ciphertext access control mechanism is a cloud storage data security approach, which uses the data encryption keys, and achieves the access control target through the control key access permissions. It is an important solution for protecting the privacy of user data in the untrusted server-side scene. CP-ABE uses a set of attributes to represent a user, generates user's private keys in accordance with their properties set, and associates with the ciphertext and the access control policy. The user can decrypt the ciphertext only when the user's private key attributes meet the ciphertext access control policies. It is a suitable ciphertext access control mechanism in cloud storage environments, encrypted data for user groups satisfied certain conditions, and does not encrypted by determining the user groups individually. The authors proposed CP-ABE mechanisms, which are flexible to satisfy the requirements for customizing access policy by the data owner in cloud storage environment [4, 5]; Jung et al. proposed a multi-authorities mechanism for preserving privacy data in cloud storage environments with CP-ABE access control program, which uses globally unique identifier for the user to prevent users conspiracy [6]; The authors proposed the CP-ABE programs for multi-authority in cloud storage to solve the key escrow problem [7, 8]. In this paper, the authors adopt CP-ABE access control policies to encrypt plaintext file encryption key, improving encryption efficiency, while add a trusted third party to solve the key escrow problem.

After obtaining the ciphertext, it also needs to be considered to provide users with data integrity verification. The techniques in this research field include: hash functions, public key cryptography, digital signatures, Merkle hash trees, and so on. The authors proposed some more efficient data integrity verification methods, making use of these methods, the client will be able to verify the integrity of the data being damaged only through exchanging minimal data with the cloud platform [9, 10]; the authors proposed a data integrity dynamic authentication service, which processed blocks and generated verification labels before storing the data, then stored the processed data into the cloud server, and verified the integrity of the data by selecting the method of random sampling [11, 12]. In this paper, the authors adopt MD5 data integrity verification program, which can verify whether data integrity suffered damage through a series of simple digest value.

3 Data Sharing Model

3.1 Secure Sharing Model

Based on the network application scene storing and sharing unstructured documents in the cloud, the authors proposed a user-controlled encrypted data sharing model in cloud storage. There are three-layer architecture in this model: the cloud user layer, the system service layer and the cloud storage layer. Respectively including: the cloud client, the authentication servers (AS), the system servers (SS) and the cloud server provider (CSP), as shown in Fig. 1.

(1) The cloud client is made up by the document owner and user. By operating the application directly, the clients upload document, retrieve document and other resource sharing services. It interacts with the AS and SS. The cloud client has the following functions: ①Creating index for a plaintext uploaded by the user, and encrypting the index keywords; ②Encrypting the plaintext and the key respectively according to encryption keys and user access policy set by the user; ③Packaging ciphertext, and uploading them to the SS; ④Getting the ciphertext and decrypting from the SS, and getting the detection and audits for the encytpted data from the AS.

(2) The authentication servers (AS) is used as a trusted third-party. It interacts with the cloud client, stores user information and encryption policy, and provides key services to help users complete the encryption and decryption; It reviews and monitors access relevant data from the SS; It verifies MD5 digest value of documents generated by the cloud client and SS.

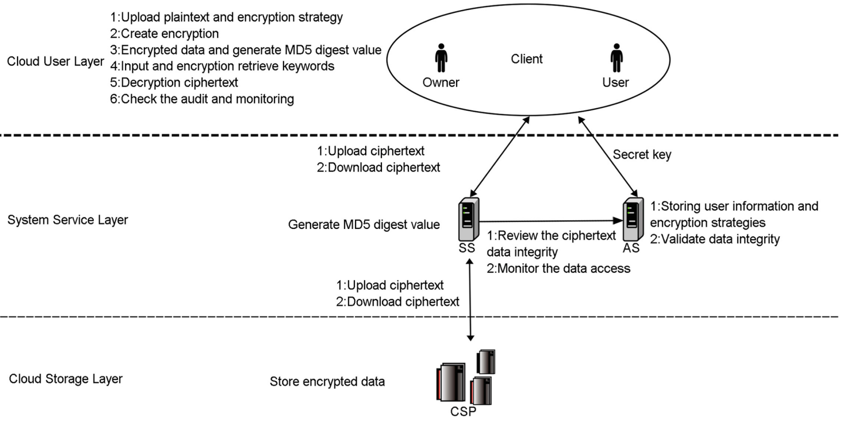
Cloud User Layer
1:Upload plaintext and encryption strategy
2:Create encryption
3:Encrypted data and generate MD5 digest value
4:Input and encryption retrieve keywords
5:Decryption ciphertext
6:Check the audit and monitoring
Client
Owner
User
1:Upload ciphertext
2:Download ciphertext
Secret key
System Service Layer
Generate MD5 digest value
SS
1:Review the ciphertext data integrity
2:Monitor the data access
AS
1:Storing user information and encryption strategies
2:Validate data integrity
1:Upload ciphertext
2:Download ciphertext
Cloud Storage Layer
Store encrypted data
CSP

Fig. 1. User-controlled encrypted data sharing model in cloud storage

(3) The system servers (SS) supports the interaction between the cloud client and the cloud server provider. It generates MD5 digest value of ciphertext, uploads the ciphertext to the cloud server provider or downloads ciphertext and returns to the user.

(4) The cloud server provider (CSP) is as a cloud storage layer. It interacts with the main SS, and provides storage services.

3.2 Application Scene

Assumption: the SS and CSP are services to be allowed to purchase, also can belong to the same service provider. We assume that they are honest but curious in this paper. The application scene of this article is network document sharing application in cloud storage [13]. An employee of a company uploaded a confidential document, and set the access policy of this document for the designation users getting. The access structure is as shown in Fig. 2.

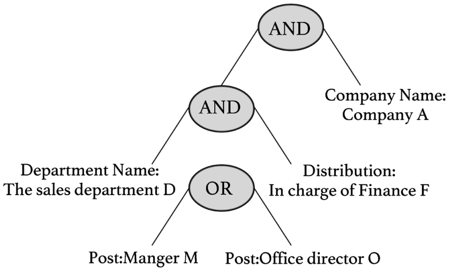
AND
AND
Company Name: Company A
Department Name: The sales department D
OR
Distribution: In charge of Finance F
Post:Manger M
Post:Office director O

Fig. 2. A user access structure example of CP-ABE

The authors proposed some symbol description in this paper, as shown in Table 1.

Table 1. Symbol description

Symbol	Explanation
MK	Master Key
PK	Public Key
SK_f, SK_c, SK_i	Private Key for encrypting document, Private Key generated by CP-ABE, Private Key for encrypting index
CT	Ciphertext
T	Access structure
S	User attributes value
I, I'	Index, Encryption Index
AES	Symmetric data encryption
DV, DV'	Message digest value by cloud client, Message digest value by the SS
Key, Key'	Query keywords, Encryption Key with SK_i
File, File', File"	User file, user encryption file in cloud client, user encryption file from CSP
Flog	Log Analysis

4 Encryption

4.1 Security Encryption

Security encryption includes two aspects in this paper: user initiative setting and system active monitoring service.

(1) Users active: It mainly uses the triple encryption scheme based on symmetric encryption and CP-ABE in this stage. It can be better assured that allowing users to submit data to the cloud storage service through this scheme.

(1) Index encryption

①GenerateStrategy()-> SK_i: After completing to create the index for the document uploaded by the user in the cloud client, then it needs to encrypt the index. The AS generates a unified secret key of the index by using UUID way, and return them back to the cloud client.

②$AES(I, SK_i)$-> *I'*: Encrypted the index by using symmetric encryption method after the cloud client obtains the encryption key of the index.

(2) File encryption

$AES(File, SK_f)$-> *File'*: Users set the key SK_f, and used symmetric data encryption their uploading documents.

(3) The symmetric key encryption by using CP-ABE

①Setup->(*MK*, *PK*): Generating master key *MK* and system public key *PK*;

②Encrypt(PK, SK_f, T)->CT: Used PK and access structures T to encrypt plaintext data SK_f, to generate a ciphertext CT;

③KeyGen(MK, S)->SK_c: Used MK and user attributes value S, to generate the corresponding user private key SK_c;

④Decrypt(CT, SK_c)->SK_f: Used *SKc* to decrypt the private key CT, and get plaintext data SK_f.

(2) System active: This stage is mainly aim at the audit and inspection of the SS by the AS, and verifies integrity of ciphertext data, and obtains available access data by the log analysis.

(1) Ciphertext integrity verification by MD5

①Cloud client and SS respectively generated message digest value for ciphertext by using MD5

a. MD5(*File'*)->*DV*: Cloud client encrypted ciphertext to generate message digest value *DV* by using MD5, and passed AS to store;

b. MD5(*File"*)->*DV'*: SS encrypted ciphertext to generate digest value *DV'* by using MD5, and passed AS;

②AS verified data integrity

```
  If DV=DV'
    return true      // If DV = DV', returns true, the data
is integrity
  Else
    return false     // Otherwise, it returns false, the
data has been tampered with
  End If
```

(2) Log analysis for user access operations

Analytics (Id, Unit, Username, IP, Action, Date)->*Flog*: AS analyzed log of user access operations in SS, and returned the data available to the document owner.

4.2 Algorithm Design

The functions of the ciphertext storage scheme designed in this paper include: (1) Users upload documents in cloud client: uploading files, create plaintext index, set file encryption key and CP-ABE user access policy. (2) The cloud client uses triple encryption with AS: encrypt the file, encrypt file encryption key by using CP-ABE, and encrypt new indexes. (3) SS Uploades ciphertext to CSP: SS uploaded ciphertext to CSP.

The pseudo-code of the ciphertext storage algorithm

```
Input: $File$, $Key$
Output: DEK($File$, $SK_f$)->$File'$ , and MD5($File'$ )->$DV$
        If $DV$ is null
            CreateIndex($File$), and DEK($I$, $SK_i$)
        Else
            CP-ABE($SK_f$)
            Upload($File'$) and Upload($I'$)
            MergeIndex($I'$) and Update the index, then Upload($File'$) to CSP
            Record data storage case in AS
        End
        $AES$($Key$, $SK_i$)->$Key'$
        Search($Key'$)
        Get the ciphertext $File''$ and MD5($File''$)->$DV'$
          If $DV' = DV$
            Decryption with CP-ABE and $AES$
          Else
            Ciphertext data has been tampered with
          End
          Record available user data in AS, and Return the available data to the data owner
```

5 Experiment

5.1 Function Realization

To verify the feasibility of the proposed model and its services in this paper, building a Hadoop cluster environment by using four ordinary PC based on CentOS6.5, we conducted experiment in network document sharing application system in cloud storage in the self-developed to test specific application examples.

We use four ordinary PC machine to build the cluster for the network document sharing application system, which includes the servers for system services and a Hadoop cluster. The cluster deploy one machine as SS and AS, and another three units as a Hadoop cluster. The operating system installed on PC is CentOS6.5, Java runtime environment is jdk1.7.0_21, Hadoop is hadoop-2.6, the program development platform is IntelliJ IDEA 13.1.2 and the data base is MySQL5.6.

We input “sales budget” for the query keyword and set a user access structure seeing in Fig. 2 for using CP-ABE. Only the users satisfied both in line with the ciphertext decryption policies and user role permission can get plaintext files in the company A. The results verify the data confidentiality and security of access control, as shown in Table 2.

Table 2. Files list for users access

Attributes	Available file number
Manager in change of finance of sales department	1, 2, 3
Manager of sales department	1, 2
Staff of sales department	1
Staff of personnel department	4

5.2 Security Analysis

Confidentiality and integrity of data is the basis for secure cloud storage. UESMCS adopts triple encryption scheme to encrypt user data to ensure the confidentiality of data, by using CP-ABE for encryption key of document file and symmetric encryption algorithm for document files, indexes and query keywords; it uses the MD5 algorithm to ensure data integrity; it brings in the trusted certification authority as a third party, which can store encryption key and user information, solves the key escrow problem and assures information security. The literatures [4, 10] demonstrated security of encryption algorithm and integrity verification algorithm, ensuring unless they have key information and access control authority, otherwise, the adversary cannot peep, tampering, theft and destruction the user data stored in cloud storage platform, and maintain a secure cloud storage platform.

5.3 Performance Analysis

This experiment uses File effectiveness E as the search results evaluation index. We tested 1000 document files. All searched document files can be decrypt effectively for the user by UESMCS. But mostly searched document files can not be decrypt effectively for the user by Non-UESMCS because of inconformity decryption strategy with CP-ABE and consuming flow with invalid files, E is 29%.

6 Conclusions

By building in the Hadoop cluster environment, using symmetric encryption, CP-ABE and MD5 encryption technology, the authors realized a user-controlled encrypted data sharing prototype system in cloud storage. The experiments show that, the proposed model UESMCS in this paper achieves an efficient and secure for sharing document network resources. Next we improve ciphertext access control algorithm to enhance security for storing and accessing resources by the user, while refine analysis of user access record data to provide more accurate system active services.

Acknowledgements. This paper is funded by the National Natural Science Foundation of China under grant numbers 61502163, 61632009, 61472451 and 61379058; the High Level Talents Program of Higher Education in Guangdong Province under Funding Support Number 2016ZJ01; the Hunan Provincial Natural Science Foundation of China under grant numbers 2016JJ5035, 2016JJ305 and 2015JJ3046; the Scientific Research Project of Hunan Province

Department of Education under grant numbers 16C0481, 14A037 and 16A059; the Project of China packaging federation under Funding Support Number 2014GSZJWT001KT010, 17ZBLWT001KT010; the Scientific Research Project of Zhuzhou City Department of science and technology bureau under grant numbers 2016-68-12.

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M.: Above the clouds: a berkeley view of cloud computing. University of California, Berkeley, Technical report, USB-EECS- 2009-28 (2009)
2. Liu, Q., Wang, G.J., Wu, J.: Time-based proxy re-encryption scheme for secure data sharing in a cloud environment. Inf. Sci. **258**(10), 355–370 (2014)
3. Feng, D.G., Zhang, M., Zhang, Y., Xu, Z.: Study on cloud computing security. J. Softw. **22**(1), 71–83 (2011)
4. Sun, G.Z., Dong, Y., Li, Y.: CP-ABE based data access control for cloud storage. J. Commun. **32**(7), 146–152 (2011)
5. Zhou, Z.B., Huang, D.J., Wang, Z.J.: Efficient privacy-preserving ciphertext-policy attribute based-encryption and broadcast encryption. IEEE Trans. Comput. **64**(1), 126–138 (2015)
6. Jung, T., Li, X.Y., Wan, Z.G., Wan, M.: Privacy preserving cloud data access with multi-authorities. In: Proceedings IEEE Infocom 2013, vol. 12, no. 11, pp. 2625–2633 (2013)
7. Dong, X., Yu, J.D., Luo, Y., Chen, Y.Y., Xue, G.T., Li, M.L.: Achieving an effective, scalable and privacy-preserving data sharing service in cloud computing. Comput. Secur. **42**(5), 151–164 (2014)
8. Yang, K., Jia, X.H., Ren, K., Zhang, B., Xie, R.T.: DAC-MACS: effective data access control for multiauthority cloud storage systems. IEEE Trans. Inf. Forensics Secur. **8**(11), 1790–1801 (2013)
9. Dodis, Y., Vadhan, S., Wichs, D.: Proofs of retrievability via hardness amplification. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 109–127. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00457-5_8
10. Ateniese, G., Kamara, S., Katz, J.: Proofs of storage from homomorphic identification protocols. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 319–333. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10366-7_19
11. Zhu, Y., Hu, H.X., Ahn, G.J., Han, Y.J., Chen, S.M.: Collaborative integrity verification in hybrid clouds. Int. J. Cooper. Inf. Syst. **21**(3), 191–200 (2012)
12. Zhu, Y., Ahn, G.J., Hu, H.X., Yau, S.S., An, H.G., Chen, S.M.: Dynamic audit services for outsourced storages in clouds. IEEE Trans. Serv. Comput. **6**(2), 227–238 (2013)
13. Wu, Y.Z., Liu, Q., Li, C.Y., Wang, G.J.: Research on cloud storage based network document sharing. J. Chin. Comput. Syst. **36**(1), 95–99 (2015)