Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/090360

Cloud Storage Architecture Achieving Privacy Protection and Sharing

Xiaoyan Sun[1], *Shenming Qu*[2], *Xiaoshu Zhu*[1], *Maosheng Zhang*[3,4,*], *Zhengwei Ren*[4] *and Cheng Yang*[5]

[1] School of computer science and engineering, Yulin normal university, 537000, Yulin Guangxi, China
[2] Department of School of Software, Henan University, 475004,Kaifeng Henan, China
[3] School of mathematics and information science, Yulin normal university, 537000, Yulin Guangxi, China
[4] Guangxi Key Lab of Multi-source Information Mining and Security, Guangxi Normal University, 541004,Guilin, China
[5] School of computer science, Wuhan University, 430072, Wuhan Hubei, China

Received: 30 Aug. 2014, Revised: 31 Nov. 2014, Accepted: 1 Dec. 2014
Published online: 1 May 2015

Abstract: Privacy protection is one of key challenges in cloud storage platform. The most popular cloud storage platforms provide good storage services while the privacy information protection is rarely referred. A novel cloud storage architecture with privacy protection is proposed. The privacy information is encrypted using symmetric encryption algorithm with encryption keys which are generated from an encryption tree. And then the secret keys are encrypted using public key cryptosystem to send them to receivers. Receivers are able to decrypt the privacy information correctly. However, the cloud storage provider and any illegal users cannot read the privacy information. The analysis show that the proposed system protect privacy efficiently with no extra storage consumptions.

Keywords: Cloud storage, cloud computing, privacy, protection, key

1 Introduction

Cloud computing platform provides users with a distributed, reliable, low cost, powerful computing power and unlimited storage capacity of data, information, and knowledge [1]. Human information technology is promoted rapidly because of the ubiquitous availability and service ability [2,3] caused by cloud computing. As a basis and core technology of cloud computing, the cloud storage has been received a widespread attention and research [4]. Although cloud storage platform can significantly reduce the costs of enterprises as well as improve the efficiency of work, but when users upload data to cloud storage platforms, information will be completely exposed to the Cloud Service Provider(CSP). Besides, it is possible that the privacy will be visited by illegal users. As a result, the problem of privacy protection in cloud storage platform has received a lot of concerns from users [5].

Data encryption is a classic technique to protect privacy in cloud storage. Data is encrypted before users upload it to the CSP. Though the encryption is able to protect privacy, the management of keys is rather inconvenient. For example, when users need to share some information to others (such as collaborators), the key must be sent first. As a consequence, the other information which is not shared will be under the risk of exposure as well. In addition, the key distribution is a great challenge in such a system. Goyal, Bethencourt [6, 7] propose a method named ABE encryption scheme, where the decryption rule is hidden in the encryption algorithm. The key distribution in cipher-text access control is not necessary here. However, this method requires owners encrypt the private information for many times, which leads to low encryption efficiency [8,9]. Mao presents a cloud storage privacy protection scheme with a third party acting as a trusted server [10]. The trusted server is acted as a communication medium between users and the cloud storage server. Users must apply for authorization to the trusted server to get the store permissions before they need to store data. After being authenticated, the third trusted server sends the same storage authentication codes that corresponds to a file to the cloud server. Users use the authentication code as a storage access credentials and then store data to the cloud server. In this structure, users and cloud storage

* Corresponding author e-mail: eterou@gmail.com

server frequently interact with the trusted server. And, the data access right in CSP is under the control of the trusted server. Both of these two features reduce the efficiency of data interaction.

The privacy protection in cloud storage platform is under research in this paper. Using a key derivation tree technology and a third-party certification, a cloud storage architecture is proposed to protect users privacy. Information owner randomly generates a root key and key sequences based on a key derivation tree. The privacy is encrypted using key sequences before uploading to CSP. It is obvious that CSP cannot read owner's private data. On the other hand, if data owner wants to share some information with another user while the others cannot obtain it, the owner encrypt the encryption keys using the target users public key which is extracting from the target users certificate. Since the target user is the only one who knows the private key, he is also the only one who can decrypt the encryption keys. There is only one conversation between the owner and authentication platform without changing the structure of CSP. The burden of CSP is not increased and extra services from CSP are not necessary. Because of different information blocks are encrypted with different keys, the proposed technique prevent users privacy from CSP and illegal users effectively and securely.

2 Framework Of Cloud Storage Platform With Privacy Protection

2.1 Google File System

Google File System (GFS), which is used in the Google's cloud computing platform, is a system with master-slave (Master/Slave) structure. GFS can provide a large number of clients with high performance of services since it is a large-scale, data-intensive and scalable distributed File systems. A GFS cluster consists of a directory server (Master) and a large number of servers (*i.e.* chunk servers). The directory server locates the resources of the whole structure. While the chunk server is responsible for storing data [11]. If a client needs to read or write some data, the data source name and chunk index must be sent to the master, and then the master sends the corresponding chunk information to the client. At last, the client reads or writes data from chunk servers according to the chunk information from the master.

2.2 Key Derivation Tree

A GFS file is divided into fixed-length blocks. Every block is uniquely labeled as a chunk-handle and duplicated to several chunk servers. In order to ensure the data privacy, every block must be encrypted. Data is divided into blocks in the local before it is encrypted. And

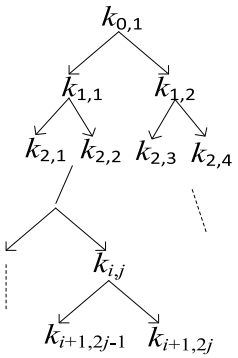

Fig. 1: A Key Tree

then it will be uploaded or updated to the CSP by data owners. To improve security, different keys are used for different blocks in the proposed system. It is clear that the key generation and management is rather complicated. To overcome this problem, we design a key derivation tree to generate encryption keys based on the method proposed in literature [12]. Using two derived algorithms F_L and F_R, one key is able to generate two sub-keys, which are called the left child key and right child key respectively. By repeating this process, a complete binary tree is produced. The node in the ith layer the jth one is denoted as $k_{i,j}$, which is used as an encryption key in the proposed system. Keys are generated as shown in eq.(1) and demonstrated in figure (1).

$$\begin{aligned} k_{i+1,2*j-1} &= F_L(k_{i,j}) \\ k_{i+1,2*j} &= \mathrm{F}_R(k_{i,j}) \end{aligned} \tag{1}$$

In the proposed system, users need to keep the root key $k_{0,1}$, generation algorithms F_L and F_R in secret. All the other keys are generated efficiently and conveniently with little information preserved.

2.3 privacy data accessing and sharing

The owner divides privacy data M into n blocks $M = \{M_1, \cdots, M_n\}$ and randomly picks the root key $k_{0,1}$ at first. And then two key generation functions F_L and F_R are designed. According to the value of n, a key tree with $d+1$ depth is generated. Key nodes are calculated as shown in equation (2).

$$\begin{aligned} &k_{0,1} = Random(T) \\ &\mathrm{F}_L(k_{i,j}) = H_1(k_{i,j}) \\ &\mathrm{F}_R(k_{i,j}) = H_2(k_{i,j}), i = 1,2,...,d \\ &d = \lceil \log_2^n \rceil \end{aligned} \tag{2}$$

Where $\lceil x \rceil$ is the minimum integer greater than x. T is a random number seed driven by time or other disturbance factors. H_1 and H_2 are the two key derivation functions. For the security considerations, one-way hash

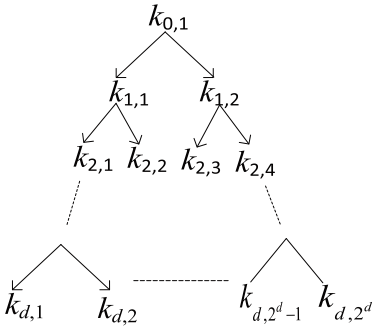

Fig. 2: The generated key tree

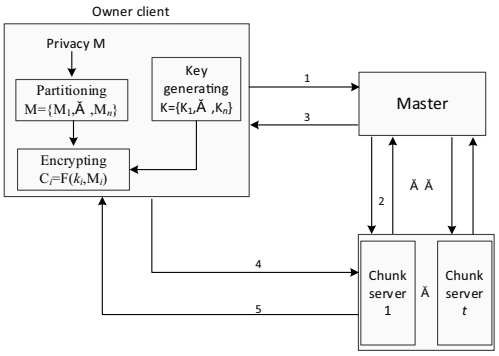

Fig. 3: privacy storage processing

functions should be used to calculate left and right child nodes. This process makes sure that it is easy to estimate left and right children nodes from parent node while it's impossible to deduce the parent from children nodes. At the same time, two different hash functions, such as SHA-2 and MD5, should be used to reduce the security threat by cipher-text leaking. The generation functions are shown in equation (3).

$$\begin{cases} \mathrm{F}_L(k_{i,j}) = \mathrm{SHA}(k_{i,j}) \\ \mathrm{F}_R(k_{i,j}) = \mathrm{MD5}(k_{i,j}) \end{cases}, i = 1,2,...,d \tag{3}$$

A key tree is generated by repeating the above computing process until the number of leaf nodes of the generation tree is greater than or equal to the block number n. The keys used in the proposed system is shown in figure (2). The encryption keys are composed of leaf nodes: $K = \{k_1,\ldots,k_n\} = \{k_{d,1},\ldots,k_{d,n}\}$. Though there are lots of keys used in the encryption process, the owner only needs to preserve the root key $k_{0,1}$ and the key tree generation algorithms F_L and F_R in practice. After generating the encryption keys, every block M_i is able to be transformed to cipher-text C_i using an encryption function F. The encryption function F must have advantages of both security and efficiency. As a matter of fact, the symmetric encryption system, Advanced Encryption Standard (AES), is an appropriate encryption algorithm to meet these two demands. The encryption process is shown in equation (4)

$$\mathrm{C}_i = \mathrm{F}(k_i, \mathrm{M}_i), i = 1,2,...,n \tag{4}$$

Encrypted blocks $C = \{C_1, C_2, \cdots, C_n\}$ are stored on chunk servers by uploading from the owner. The overall framework is illustrated in figure (3). And the communication is summarized as follows: the owner sends a storage request to the master first. And then, the master communicates with chunk servers to determine the available servers and return the information of available servers to the owner. According to the information from the CSP, the owner uploads the encrypted privacy data to chunk servers and finally, chunk servers send storage information to the owner and master.

It is easy to share privacy to specified objects. Assuming that the owner wants to send blocks $M = \{M_1, \cdots, M_n\}$ to another users (target users) along with the decryption keys, in addition to block position information. Meanwhile the privacy must be kept away from being obtained by any other illegal users (including CSP). The owner gets the target users public key from a third party certification platform. And then, the encryption keys of privacy are encrypted using target users public key and the encrypted keys are sent to the target user. Since the target user is the only one who has the private key to decrypt encryption keys, the encryption keys are transformed safely. The framework is represented graphically in figure 4 and the communication process is summarized as follows:

(1) The owner sends (filename, chunk handle) to the master to request accessing file filename
(2) The master communicates with chunk servers to confirm the location of files and available chunk servers;
(3) The master returns (Chunk handle, Chunk location) to tell the locations of available chunk servers to the owner.
(4) The owner requests the target user's certificate from a key management center (KMC) to get the user's public key.
(5) KMC encrypts the target user's public key k_{up} using his own private and sends it to the Owner;
(6) The owner decrypts the information from the KMC using KMC's public key to get the target user's publick k_{up}, and then the owner encrypts the encryption keys $K = \{k_1, \cdots, k_n\}$ and (Chunk handleChunk location) and sends the encrypted data to the target user.
(7) The target user decrypts the encrypted data to get encryption keys $K = \{k_1, \cdots, k_n\}$ and (Chunk handle, Chunk location). Then the target user needs to communicate to the corresponding chunk servers for accessing data $C = \{C_1, \cdots, C_n\}$
(8) Chunk servers send data $C = \{C_1, \cdots, C_n\}$ to the target user.

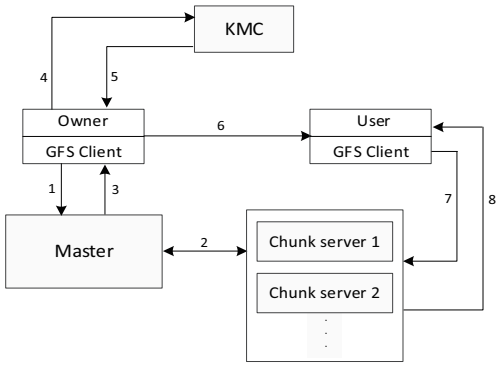

Fig. 4: privacy sharing process

(9) The target user decrypts $C = \{C_1, \cdots, C_n\}$ using encryption keys to recover the data blocks $M = \{M_1, \cdots, M_n\} : M_i = AES^{-1}(K_i, C_i)$.

3 Security Analysis

The critical factors in the proposed privacy protection system in cloud storage platform are the content security and the key security, *i.e.* whether the CSP or illegal users are able to get the privacy, or the encryption keys are leaked.

(1) The CSP is unable to read the privacy. Though the CSP storages the owner's privacy, the privacy is impossible to be leaked to the CSP since the privacy is encrypted with AES, which is a secure symmetric encryption algorithm proved by top scientists all over the world and the encryption efficiency is much higher than public key cryptosystems. So the privacy is guaranteed against CSP. In addition, because of the abundance of keys produced by the key generation tree, the encryption key is varied for different blocks. The encryption system is an approximate one-time pad system which is proved to be an ideal safe system by Shannon.

(2) The privacy is impossible to be exposed to illegal users. Supposing there are s chunk servers and the owner's privacy is divided into n blocks, the probability of all blocks are preserved in the same chunk server is $1/s^{n-1}$. If there are q users and every user store k blocks in the CSP, the possibility of illegal users get the specified owners blocks is $1/s^{n-1} \times C^n_{m,k+n}$ [9]. Apparently, it is impossible for illegal users to get all blocks of the owner's in one chunk server. On the other hand, even if illegal users get some of blocks, the privacy will not be leaked since every block is encrypted by a secure encryption algorithm (*i.e.* AES) with an unique key.

(3) The encryption keys are secure. When the owner requests for the target user's public key, the request is encrypted using KMC's public key. The KMC is the only one who owns the private key to decrypt the request. After that, the KMC encrypts the target users public key using his own private key and sends the public key to the owner. The owner decrypts it using the KMCs public key. The transformation process ensures that the owner gets the target user's public key correctly. The owner encrypts the encryption keys using the target user's public key. Since the corresponding private key is kept by the target user, the target user is the only one who can decrypt. On the contrary, anyone else is unable to get the encryption keys. As a result, the target user obtains the encryption keys, block names and chunk indexes successfully. That the data transformed among the owner, KMC and target user is encrypted is a guarantee that the data is safe.

(4) The key management is efficient. Since encryption keys are shared to the target user, it is a big issue whether the target user is able to estimate the other keys, *i.e.* if the target user deduces the other keys in key generation system based on the shared keys, the proposed system is absolutely not reliable. Though the target user gets some of keys, the generation functions f_L, f_R are kept in secret. As a consequence, the target user is unable to estimate children keys. Because of the one-way and collision-resistant characteristics of hash functions, anyone is impossible to measure the parent key even if he knows children keys. In a word, the target user cannot obtain other keys from the shared keys. By the way, though there are lots of keys needed in the proposed system, the owner only needs to storage one root key and two generation functions. All children keys are not necessary to be preserved in owner's hard disks or other devices.

4 Conclusion

A private protection scheme integrating multiple key generation technique, symmetric encryption scheme and public key cryptosystem in cloud storage platform is proposed. The multiple key generation technique is used to generate encryption keys and privacy is encrypted using a symmetric encryption scheme. The encryption keys and location of shared data are encrypted utilizing public key cryptosystem before sending to the target user. A KMC acts as a key transformation agency to correctly provide real keys to the owner. There is only one conversation between the owner and KMC, in addition, it is not necessary to assume secure channels. The time cost is definitely reduced. Security analysis shows the proposed scheme is able to protect clients' privacy and share the specified information to other users without leaking and extra communication burden.

Acknowledgement

This research was supported by Key project of Education Department of Guangxi (No.2013ZD056), Guangxi Key Lab of Multi-source Information Mining and Security (MIMS13-06), Key project of Yulin Normal university (No.2013YJZD04), Nature Science Foundation of Guangxi Province (No.2013GXNSFAA019337, 2014GX NSFBA118268, 2014GXNSFBA118010), Science and Technology Foundation of Guizhou Province (No.LKS[2011]1).
The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] S Manvi, G Shyam. Journal of Network and Computer Applications **41**, 424-440 (2014)
[2] W.Y Zeng, Y.L Zhao, M Shang. Journal of Computer Research and Development **48**, 234-239 (2011)
[3] Z.W Liu, Z.L Wen, H.T Zhang. Journal Of Computer Research And Development **48**, 26-31 (2012).
[4] YK Yeo, TI Kwon, KH Lee, An energy effective PID tuning method for the control of polybutadiene latex reactor based on closed-loop identification, Korean Journal of Chemical Engineering, **21**, 935-941 Springer (2004).
[5] L Sun, Z.S Dai. Journal Of Computer Applications **32**, 13-15 (2012)
[6] G Vipul, P Omkant, S Amit et.al. Proceedings of the 13th ACM Conference on Computer and Communications Security. 89-98 (2006)
[7] B John, S Amit, W Brent. IEEE Symposium on Security and Privacy, 321-334 (2007)
[8] H Cheng, M Zhang, D.G Feng. Journal Of Communications **32**, 125-131 (2011). 125-131.)
[9] E.A Victor, L.L M, D.W Shin. Proceedings of the Computer Software and Applications Conference Workshops (COMPSACW). 371-375 (2010).
[10] J Mao, K Li, X.D Xu. Journal of tsinghua University(Science And Technology) **51**, 1357-1362 (2011).
[11] K Chen, W.M Zheng. Journal of Software **20**, 1337-1348 (2009).
[12] M.J Atallah, M Blanton, N Fazio, et al. ACM Transactions on Information and System Security (TISSEC) **12**, 3-18 (2009).

Xiaoyan Sun received the bachelor degree from Guangxi Normal University in 2004 and master degree in computer science from the school of mathematics and computing science in Guilin University of Electronic Technology. Currently, she is a lecturer at Yulin normal University, China. Her research interests include cryptography, software protection and watermarking. She has published over 10 papers and 2 books on journals and/or international conferences. Her research has been supported by 8 provincial-level research projects.

Qu shengming received the B.S degree in computer science from HebeiUniversity, Baoding, China, in 2004, the M.S degree in computer applicationtechnology from Henan University, Kaifeng, China, in 2010. He is currently pursuing the Ph.D. degree in National Engineering Research Center forMultimedia Software, School of Computer, Wuhan University, Wuhan, China.Currently he is a lecturer at Henan University. His research interests include image/video processing, computer vision.

Xiaoshu Zhu received the bachelor degree from Jiangsu University of Science and Technology in 1996 and master degree in computer science from the school of mathematics and computing science in Guilin University of Electronic Technology in 2006. Currently, she is a professor at Yulin normal University, China. Her research interests include network and cloud computing.

Maosheng Zhang received the bachelor degree from Hubei University in 2004 and master degree in mathematics from Dalian university of technology in 2009. He is a Ph.D. candidate of Wuhan University. Currently he is a associate professor at Yulin normal university. His research interests are in cryptography, watermarking and multimedia coding.

Zhengwei Ren received the master degree in computer science in 2010 and Doctor of Philosophy in computer science from Wuhan University in 2014. Currently, he is working on network security and privacy. His research interests include applied cryptology and cloud computing security.

Cheng Yang received his B.S degree in physics from Guizhou Normal University and M.S degree in computer application technology from Guizhou University, Guiyang, China, in 2002 and 2010 respectively. He is currently pursuing his Ph.D. degree in National Engineering Research Center for Multimedia Software, School of Computer, Wuhan University, Wuhan, China. His research interests include audio compressing and processing, video compressing and processingdistributed and parallel computing.