Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

An Efficient Time specified Access Mechanism for Cloud Storage

Minu Varghese[1], Prema Mani[2]
[1,2]*Department of Computer Science, IIET,Nellikuzhi*

Abstract—This research work generally focused on providing an efficient time specified access mechanism for shared cloud data. With the rapid development of versatile cloud services, it becomes largely susceptible to use cloud services to share data in a friend circle in the cloud computing environment. Since it is not feasible to implement full lifecycle privacy security, access control becomes a difficult task, especially when share sensitive data on cloud servers. In order to solve this problem, proposed a key control mechanism with time-specified attributes (KCM-TSA), which is able to enforce the security of user privacy over the cloud in a controllable way. In the KCM-TSA scheme, every ciphertext is associated with a time interval while private key is associated with a time instant. The ciphertext can only be decrypted if both the time instant is in the allowed time interval and the attributes associated with the ciphertext satisfy the key's access structure. The KCM-TSA is able to solve some important security problems by supporting user-defined authorization period and by providing fine-grained access control during the period. The sensitive data will be securely self-destructed after a user-specified expiration time. This scheme satisfies the security requirements and is superior to other existing schemes.

Keywords— Self destruction, Access control, Attribute based encryption.

I. INTRODUCTION

Cloud computing is an emerging computing paradigm in which resources of the computing infrastructure are provided as services over the Internet. As promising as it is, this paradigm also brings forth many new challenges for data security and access control when users outsource sensitive data for sharing on cloud servers, which are not within the same trusted domain as data owners. To keep sensitive user data confidential against untrusted servers, existing solutions usually apply cryptographic methods by disclosing data decryption keys only to authorized users. However, in doing so, these solutions inevitably introduce a heavy computation overhead on the data owner for key distribution and data management when fine-grained data access control is desired, and thus do not scale well. The problem of simultaneously achieving fine-grainedness, scalability, and data confidentiality of access control actually still remains unresolved.

Cloud computing is considered as the next step in the evolution of on-demand information technology which combines a set of existing and new techniques from research areas such as service-oriented architectures (SOA) and virtualization. With the rapid development of versatile cloud computing technology and services, it is routine for users to leverage cloud storage services to share data with others in a friend circle, e.g., Dropbox, Google Drive and AliCloud. Along with this new paradigm, various business models are developed, which can be described by terminology of "X as a service (XaaS)" where X could be software, hardware, data storage, and etc. Successful examples are Amazon's EC2 and S3, Google App Engine and Microsoft Azure which provide users with scalable resources in the pay-as-you- use fashion at relatively low prices. For example, Amazon's S3 data storage service just charges $0.12 to $0.15 per gigabyte-month. As compared to building their own infrastructures, users are able to save their investments significantly by migrating businesses into the cloud. With the increasing development of cloud computing technologies, it is not hard to imagine that in the near future more and more businesses will be moved into the cloud Security.

With the development of cloud computing and popularization of mobile web, cloud online services are becoming more useful for people's life. People are requested to submit their personal information to the cloud by the internet. When people do this, they hope the service provider will provide security to protect their data from leaking, so other people will not invade their privacy. With more and more services and applications are emerging in the internet, it becomes easier to expose the user's sensitive data in the internet. Without much attention to this problem, it will result in the break out of leaking messages. For e.g. Emails can be leaked via cloud service provider, negligence or hackers. One of the most important reasons for exposing sensitive user data is that the user data will be stored in the cloud environment for a long time, which may not be controlled by the user himself. These are the major challenges to protect people's privacy. Self-destructing data systems are designed to address these concerns. Their goal is to destroy data after a pre-specified timeout. Self destruction is implemented by encrypting data with a key and then escrowing the information needed to reconstruct the decryption key with one or more third parties. Assuming that the key reconstruction information disappears from the escrowing third parties at the intended time, encrypted data will become permanently unreadable: (1) even if an attacker obtains a copy of the encrypted data and the user's cryptographic keys and passphrases after the timeout, (2) without the user or user's agent taking any explicit action to destroy it, (3) without need to change any stored or archived copies of that data, and (4) without the user relying on secure hardware. Once the key-renovation details disappears, data owners can be confident that their data will remain inaccessible to powerful attacks, whether from hackers who obtain copies of backup archives and passphrases or through legal means.

In this paper, proposes a key control mechanism with time-specified attributes (KCM-TSA), a controllable data self-destruction system for cloud storage networks. KCM-TSA not only inherits the advantages of SeDas to leverage active storage systems to manage the survival time of secret key but also integrate the attribute-based encryption (ABE) algorithms to implement the access control over the encrypted data. Additionally, KCM-TSA deals with the untrusted cloud storage networks in two ways. One is the data self-destruction. Instead of dividing and storing the raw encryption key directly as in SeDas, it leverages the ABE-based data self-destruction framework, a consequence, even an attacker can gather the sufficient number of parts to recover the encrypted key, he/she still cannot decrypt it due to lack of the secret key.

II. RELATED WORKS

Attribute-based encryption is one of the important applications of fuzzy identity-based encryption. ABE comes in two flavors called KP-ABE [10] and ciphertext-policy ABE (CP-ABE) [5]. In CP-ABE, the ciphertext is associated with the access structure while the private key contains a set of attributes. Bethen court et al. proposed the first CPABE scheme [5], the drawback of their scheme is that security proof was only constructed under the generic group model. Waters used a linear secret sharing scheme (LSSS) matrix as a general set of access structures over the attributes and proposed an efficient and provably secure CP-ABE scheme under the standard model.

In KP-ABE, the idea is reversed: the ciphertext contains a set of attributes and the private key is related to the access structure. The first construction of KP-ABE scheme was proposed by Goyal et al. [10]. In their scheme, when a user made a secret request, the trusted authority determined which combination of attributes must appear in the ciphertext for the user to decrypt. Instead of using the Shamir secret-sharing technique in the private key, this scheme used a more generalized form of secret sharing to enforce a monotonic access tree.

A well-known method for addressing this problem is secure deletion of sensitive data after expiration when the data was used [6]. Reardon et al. leveraged the graph theory, B-tree structure and key wrapping and proposed a novel approach to the design and analysis of secure deletion for

persistent storage devices [7]. Because of the properties of physical storage media, the above-mentioned methods are not suitable for the cloud computing environment as the deleted data can be recovered easily in the cloud servers.

III. PROPOSED WORK

Proposed a key control mechanism with time-specified attributes (KCM-TSA), which is able to enforce the security of user privacy over cloud in a controllable way. In the KCM -TSA scheme, every ciphertext is associated with a time interval while private key is associated with a time instant. The ciphertext can only be decrypted if both the time instant is in the allowed time interval and the attributes associated with the ciphertext satisfy the key's access structure.. It incorporates attribute-based encryption (ABE) algorithms to implement the access control over the encrypted data. This method uses Attribute based encryption with time specified attributes and key management for securely deleting data. It uses meta servers and multiple storage servers for providing fast and efficient accessing of data. Efficient load balancing is also included in this system. That is scheduling of storage servers perform effectively. Instead of dividing and storing the raw encryption key directly as in existing system a secret message exchanging encryption is employed.

The proposed method is shown in Figure 1, which includes five phases such as (1)Data owner (2)User (3) storage servers (4) Metadata server, and (5) Key Server. The design is centered around the cooperation between the metadata server and the Key Server to accomplish the authorized accesses to the encrypted data stored in the Storage servers for the data owner or other users.

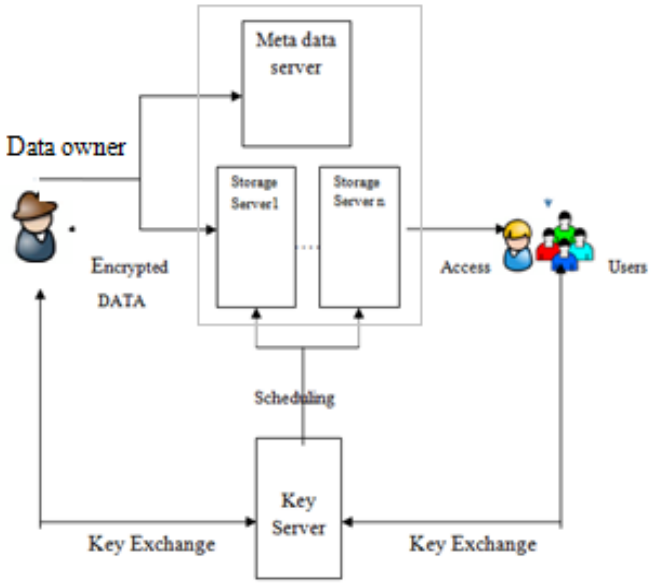

Figure 1: Proposed framework

1. Data owner: Data owner is a user who owns data, and publishes to store it into the Storage Servers. A data owner is responsible for defining (time attribute based) access policy, and enforcing it on its own data by encrypting the data under the policy before storing it. The data owner also acts as the authority and is in charge of key generation.

2. User: User is an entity who wants to access the encrypted data. If a user satisfies the access policy of the encrypted data defined by the data owner, then he will be able to get the encrypted files and decrypt the ciphertext and obtain the data. The only difference between a data owner or user is that the data owner defines the access policy.

3. Storage Servers: Storage Servers represent a set of storage nodes. It contains almost unlimited storage space which is able to store and manage all the data or files in the system.

4 Metadata server (MDS): The metadata server (MDS) is responsible for user management, server management, session management and object metadata management. It is responsible for metadata maintenance, workload balance as well as other information related to the access control policies.

5. Key Server: The Key Servers is in charge of controlling the accesses from users to the encrypted data in storage servers and providing corresponding content services. It supplies and controls the encryption key for the encrypted data by assembling its divided parts from storage servers. It performs scheduling of storage servers and key management

3.1. User Authentication

A new user has to first create a profile. This is done by registration. A user id and password are submitted by the user. The user can login successfully only if user id and password are entered correctly. The login is a failure if the incorrect user id or wrong password is entered by the user. This helps in preventing unauthorized access.

3.2. Key Exchange :

The client must first register itself with the server after which storage node also continues the registration with the key server. Attribute based encryption with time specified attributes is used to encrypt the data and for this a secret key is necessary. Once the keys are generated it has to be shared between the user application and server using message key exchange technique. Key exchange works by encrypting user's private key with the server's public key. Only the indented server can decrypt the message because it requires the use of server's private key. Therefore a third party who interprets the secret cannot decrypt and use it.

3.3. Data Process

User's applications should implement logic of data process and act as a client node. There are two different processes: uploading and downloading.
Process of uploading files: When a user uploads a file, he can specify its accessibility. That is he can decide whom it can be available and can specify expiration time. There are multiple storage services for a user to store data. Time specified attribute based encryption is used for data encryption. Encrypted ciphertext is stored in storage servers while time attributes and metadata are stored in metadata servers.

Process of downloading files: Any user who has relevant permission can download data stored in the data storage system. The data must be decrypted before use. The user can download the file only before its expiration time. After authorization period the file will be unavailable to him.

3.4. Scheduling

Files are stored in multiple servers based on the easiness of accessibility. So an effective load balancing mechanism is adopted. Based on the weightage of different parameters like storage, distance, bandwidth etc for each available server, the best server can be selected. This scheduling task is performed by key server.

3.5. Data Self-Destruction:

After user defined expiration time the shared data will be self destructed. The ciphertext can only be decrypted if both the time instant is in the allowed time interval and the attributes associated with the ciphertext satisfy the key's access structure. The proposed scheme able to solve some important security problems by supporting user-defined authorization period and by providing fine-grained access control during the period. The sensitive data will be securely self-destructed after a user-specified expiration time.

IV. CONCLUSION

With the rapid development of versatile cloud services, a lot of new challenges have emerged. One of the most important problems is how to securely delete the outsourced data stored in the cloud severs. So here proposed a novel KCM-TSA scheme which is able to achieve the time specified ciphertext in order to solve these problems by implementing flexible fine-grained access control during the authorization period and time-controllable self-destruction after expiration to the shared and outsourced data in cloud computing. It also gave a system model and a security model for the KCM-TSA scheme. The proposed KCM-TSA scheme is superior to other existing schemes. Here attempt to provide an efficient access mechanism for shared cloud data with user defined authorization period.

REFERENCES

[1] A. F. Chan and I. F. Blake, "Scalable, server-passive, user-anonymous timed release cryptography," in Proc. Int. Conf. Distrib. Comput. Syst., 2005, pp. 504–513
[2] D. A. W. Dent and Q. Tang, "Revisiting the security model for timed release encryption with pre-open capability," in Proc. Inf. Security, 2007, pp. 158–174.
[3] B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," IEEE Trans. Cloud Comput., vol. 2, no. 1, pp. 43–56, Jan.–Mar. 2014.
[4] G. Wang, F. Yue, and Q. Liu, "A secure self-destructing scheme for electronic data," J. Comput. Syst. Sci., vol. 79, no. 2, pp. 279–290, 2013.
[5] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute- based encryption," in Proc. 28th IEEE Symp. Security Privacy, 2007, pp. 321–334.
[6] J. Reardon, D. Basin, and S. Capkun, "Sok: Secure data deletion," in Proc. 34th IEEE Symp. Security Privacy, 2013, pp. 1–15.
[7] J. Reardon, H. Ritzdorf, D. Basin, and S. Capkun, "Secure data deletion from persistent media," in Proc. ACM Conf. Comput. Commun Security, 2013, pp. 271–284.
[8] J. Xiong, Z. Yao, J. Ma, X. Liu, Q. Li, and J. Ma, "Priam: Privacy preserving identity and access management scheme in cloud," KSII Trans. Internet Inf. Syst., vol. 8, no. 1, pp. 282–304, 2014.
[9] L. Zeng, S. Chen, Q. Wei, and D. Feng, "Sedas: A self-destructing data system based on active storage framework," IEEE Trans. Magnetics, vol. 49, no. 6, pp. 2548–2554, Jun. 2013.
[10] Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. 13th ACM Conf. Comput. Commun. Security, 2006, pp. 89–98.

