ISSN:0975-9646

Public Auditing and Self-Destructing Approach for Shared Data on Cloud

Mr. Amir Ali[#1], Prof Savitri Patil[*2]

#Department of Computer Science and Engineering
**Department of Information and Technology Engineering*

[#,]G.H. Raisoni College of Engineering & Management Wagholi, Pune.*

***Abstract*— In past years, the rapid development of cloud storage services makes it easier than ever for cloud users to share data with each other. To ensure users' confidence of the integrity of their shared data on cloud, a number of techniques have been proposed for data integrity auditing with focuses on various practical features, secure data destructing, public integrity auditing etc. Since it is not feasible to implement full lifecycle privacy security, access control becomes a challenging task, especially when we share sensitive data on cloud servers. In order to tackle this problem, this paper proposes a key-policy attribute-based encryption with time-specified attributes (KP-TSABE), a novel secure data self-destructing scheme in cloud computing. Additionally this paper introduces public integrity auditing scheme for cloud data sharing services that checks the integrity of user's sensitive data being stored on the cloud. In the KP-TSABE scheme, every ciphertext is labelled 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.**

***Keywords*— Sensitive data, assured deletion, fine-grained access control, privacy preserving, public auditing.**

I. INTRODUCTION

Cloud computing has been envisioned as the next generation information technology (IT) architecture for enterprises, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk. As a disruptive technology with profound implications, cloud computing is transforming the very nature of how businesses use information technology. One fundamental aspect of this paradigm shifting is that data are being centralized or outsourced to the cloud. From users' perspective, including both individuals and IT enterprises, storing data remotely to the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, universal data access with location independence, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc.

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 [1]. The shared data in cloud servers, however, usually contains users' sensitive information (e.g., personal profile, financial data, health records, etc.) and needs to be well protected [2]. As the ownership of the data is separated from the administration of them [3], the cloud servers may migrate users' data to other cloud servers in outsourcing or share them in cloud searching. Therefore, it becomes a big challenge to protect the privacy of those shared data in cloud, especially in cross-cloud and big data environment [5]. In order to meet this challenge, it is necessary to design a comprehensive solution to support user-defined authorization period and to provide fine-grained access control during this period. The shared data should be self-destroyed after the user-defined expiration time. One of the methods to alleviate the problems is to store data as a common encrypted form. The disadvantage of encrypting data is that the user cannot share his/her encrypted data at a fine-grained level. When a data owner wants to share someone his/her information, the owner must know exactly the one he/she wants to share with [6]. In many applications, the data owner wants to share information with several users according to the security policy based on the users' credentials. Attribute-based encryption (ABE) has significant advantages based on the tradition public key encryption instead of one-to-one encryption because it achieves flexible one-to-many encryption. 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.

For correct execution of real time examples like Dropbox applications, one problem is to assure data integrity, i.e., each data modification operation is indeed performed by an authorized group member and the data remains intact and update to date thereafter. This problem is important given the fact that cloud storage platforms, even well-known cloud platforms, may experience hardware/software failures, human errors and external malicious attacks. To fully ensure the data integrity and save the cloud users' computation resources as well as online burden, it is of critical importance to enable public auditing service for cloud data storage, so that users may resort to an independent third-party auditor (TPA) to audit the outsourced data when needed. The TPA, who has expertise and capabilities that users do not, can periodically check the integrity of all the data stored in the cloud on behalf of the users, which provides a much more easier and affordable way for the users to ensure their storage correctness in the

cloud. Moreover, in addition to help users to evaluate the risk of their subscribed cloud data services, the audit result from TPA would also be beneficial for the cloud service providers to improve their cloud-based service platform, and even serve for independent arbitration purposes. In a word, enabling public auditing services will play an important role for this nascent cloud economy to become fully established; where users will need ways to assess risk and gain trust in the cloud.

The remainder of this paper is organized as: A basic overview data encryption is specified in Section II. The Section III briefly discusses certain related work. Section IV describes the motivation and goals of system. In section V, proposed system is concrete with a system design and lastly, in Section VI, the conclusion is summarized.

II. DATA ENCRYPTION OVERVIEW

The shared data in cloud servers usually contains users' sensitive data and needs to be well protected. The cloud servers may migrate users' data to other cloud servers in outsourcing or share them in cloud searching. Therefore, it becomes a big challenge to protect the privacy of those shared data in cloud, especially in cross-cloud and big data environment. One of the methods to alleviate the problems is to store data as a common encrypted form. The disadvantage of encrypting data is that the user cannot share his/her encrypted data at a fine-grained level. When a data owner wants to share someone his/her information, the owner must know exactly the one he/she wants to share with [6]. In many applications, the data owner wants to share information with several users according to the security policy based on the users' credentials. Attribute-based encryption (ABE) has significant advantages based on the tradition public key encryption instead of one-to-one encryption because it achieves flexible one-to-many encryption.

ABE scheme provides a powerful method to achieve both data security and fine-grained access control. In the key-policy ABE (KP-ABE) scheme to be elaborated in this paper, the ciphertext is labeled with set of descriptive attributes. Only when the set of descriptive attributes satisfies the access structure in the key, the user can get the plaintext. In general, the owner has the right to specify that certain sensitive information is only valid for a limited period of time, or should not be released before a particular time. Timed-release encryption (TRE) provides an interesting encryption service where an encryption key is associated with a predefined release time, and a receiver can only construct the corresponding decryption key in this time instance. On this basis, Paterson and Quaglia proposed a time specific encryption (TSE) [10] scheme, which is able to specify a suitable time interval such that the ciphertext can only be decrypted in this interval. However, applying the ABE to the shared data will introduce several problems with regard to time-specific constraint and self-destruction, while applying the TSE will introduce problems with regard to fine-grained access control. Thus, in this paper, we attempt to solve these problems by using KP-ABE and adding a constraint of time interval to each attribute in the set of decryption attributes.

III. RELATED WORK

Bethencourt et al. projected the first CPABE technique [12], which encrypted data, and may be kept personal though the storage cloud server is untrusted; furthermore, these techniques are secure beside collusion attacks. The disadvantage of their technique is that security proof was simply created below the generic group model.

To address this problem, Cheung and C. Newport [13] presented another construction under a standard model called as CP-ABE scheme. The scheme permits an encryptor to use every AND gate on positive as well as negative attributes by way of an access policy on the ciphertext.

Waters [14] used a linear secret sharing scheme (LSSS) matrix by means of a common set of access structures above the attributes in addition projected an efficient and provably protected CP-ABE system under the typical model.

Yu et al. used a combining approach of KP-ABE, proxy re-encryption, as well as lazy re-encryption which permits the data owner to represent most of the calculation responsibilities elaborate in fine-grained data access control in the direction of untrusted cloud servers deprived of disclosing the underlying data contents [17].

Tysowski and Hasan modified the ABE and leveraged re-encryption algorithm to propose a novel scheme to protect mobile user's data in cloud computing environment [18]. They proposed tis scheme to allow authorized users access to cloud data depend on the essential attributes satisfaction such that the greater evolutional load from cryptographic processes is allocated to the cloud provider and the whole communication cost is let down for the mobile user.

Cachin et al. worked on a policy graph to define the association among attributes as well as the protection class and projected a policy-based secure data deletion system [20]. Deletion processes are conveyed in terms of a policy that designates data destruction over deletion attributes in addition over protection classes. The policy links attributes as indicated in deletion processes to the protection classes that require to be removed consequently.

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 [21].

A data self-destructing scheme, first proposed by Geambasu et al. [23], is a promising approach which designs a Vanish system enables users to control over the lifecycle of the sensitive data.

Wang et al. improved the Vanish system and proposed a secure self-destructing scheme for electronic data (SSDD) [24]. Authors attain aim by first encrypting the data, and after that distributing together the decryption key and a part of the cipher text into the distributed hash table (DHT) network.

Wolchok et al. made a lot of experiments and confirmed that the Vanish system is vulnerable to Sybil attacks by using the Vuze DHT network [25]. So the security of the SSDD scheme is also becomes questionable.

To address this problem, Zeng et al. proposed a SeDas system, which is a novel integration of cryptographic techniques with active storage techniques [26].

Boneh and Franklin leveraged the DHT network and identity-based encryption (IBE) and proposed an IBE-based secure self-destruction (ISS) scheme [22].

Xiong et al. employed identity-based timed release encryption (ID-TRE) algorithm and the DHT network and proposed a full lifecycle privacy protection scheme for sensitive data (FullPP), which is able to provide full lifecycle privacy protection for users' sensitive data by making it unreadable before a predefined time and automatically destructed after expiration [3].

Peterson and Quaglia [10] introduce and discover the concept of Time-Specific Encryption (TSE). They extend Plain TSE into the public-key as well as identity-based settings, where receivers are moreover equipped by private keys and both public keys as well as identities, and where decryption needs the use of the private key in addition to an appropriate Time Instant Key (TIK).

Kasamatsu et al. designed an efficient TSE scheme by using forward secure encryption (FSE) in which the size of the ciphertext is greatly small than that generated by the previous schemes [33].

IV. Motivation And Goal

A. Motivation

Some limitation of previous implemented approaches is as follows:

- Secure self-destruction scheme SSDD does not consider the issue of the desired release time of the sensitive data, since expiration time of SSDD is limited by the DHT network and cannot be determined by the user.
- SSDD and many other schemes are dependent on the ideal assumption of "No attack on VDO (vanishing data object) before it expires".
- Vanish scheme [1] is vulnerable to the Sybil attacks from the DHT network; the SSDD scheme and other schemes are similar.

As a result, unauthorized users can freely access to the sensitive data and this flaw would lead to a serious privacy disclosure [25].

To address these problems, in this paper, we propose a novel solution called key-policy attribute based encryption with time-specified attributes (KP-TSABE) scheme, which is based on our observation that, in practical cloud application scenarios, each data item can be associated with a set of attributes and every attribute is associated with a specification of time interval The data owner encrypts his/her data to share with users in the system, in which every user's key is associated with an access tree and each leaf node is associated with a time instant. If the time instant is not in the specified time interval, the ciphertext cannot be decrypted, i.e., this ciphertext will be self-destructed and no one can decrypt it because of the expiration of the secure key. Therefore, secure data self-destruction with fine-grained access control is achieved.

As well as this paper motivate the public auditing system of data storage security in cloud computing and provide a privacy-preserving auditing protocol. This technique enables an external auditor to audit user's cloud data without learning the data content.

B. Goal

This framework has the following advantages with regard to security and fine-grained access control compared to other secure self-destructing schemes and data integrity auditing scheme.

- This scheme is the first to support scalable and efficient privacy-preserving public storage auditing in cloud.
- KP-TSABE supports the function of user-defined authorization period and ensures that the sensitive data cannot be read both before its desired release time and after its expiration.
- KP-TSABE does not require the ideal assumption of "No attacks on VDO before it expires".
- KP-TSABE is able to implement fine-grained access control during the authorization period and to make the sensitive data self-destruction after expiration without any human intervention.

V. Proposed Model

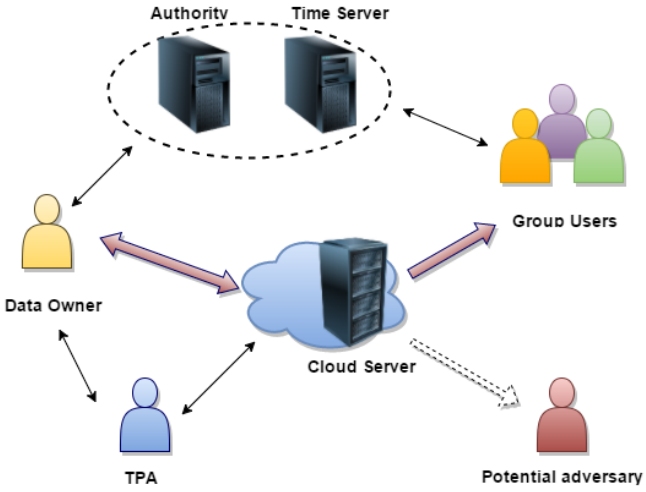

Fig. 1 Proposed Architecture

A. Concept Explanation

1) Authorization period: It is a time interval predefined by a data owner, starting from the desired release time and ending at the expiration time [1].

2) Expiration time: It is a threshold time instant predefined by the owner. The shared data can only be accessed by the user before this time instant because the shared data will be self-destructed after expiration.

3) *Full lifecycle:* It is a time interval from the creation of the shared data, authorization period to expiration time.

B. System Model

In this system, mainly focus on how to achieve fine grained access control during the authorization period of the

shared data in cloud and how to implement self-destruction after expiration and public integrity auditing.

Specifically, system is divided in seven entities as follows:

1) Data owner: Data owner can provide data or files that contain some sensitive information, which are used for sharing with his/her friends (data users). All these shared data are outsourced to the cloud servers to store.

2) Authority: It is a time reference server without any interaction with other entities involved in the system. It is responsible for a precise release time specification

3) Time server: It is a time reference server without any interaction with other entities involved in the system. It is responsible for a precise release time specification.

4) Data users: Data users are some people who passed the identity authentication and access to the data outsourced by the data owner. Notice that the shared data can only be accessed by the authorized users during its authorization period.

5) Cloud servers: It contains almost unlimited storage space which is able to store and manage all the data or files in the system. Other entities with limited storage space can store their data to the cloud servers.

6) Third Party Auditor: The third party auditor is able to publicly verify the integrity of shared data for a group of users without retrieving the entire data. The TPA issues an audit message or challenge to the cloud server to make sure that the cloud server has retained the data properly at the time of the audit

7) Potential adversary: It is a polynomial time adversary described in the security model of the KPTSABE scheme [1].

C. Formal Model

The proposed scheme can be described as a collection of the following four algorithms: Setup, Encrypt, KeyGen, and Decrypt.

1) Setup (1^k, U): This algorithm is run by the Authority and takes as input the security parameter 1^k and attributes universe *U*. It also generates system public parameters *params* and the master key *MSK*. The Authority publishes *params* and keeps *MSK* secret to itself.

2) Encrypt(M, params, S, TS): Given the public parameters *params*, the shared message *M* which the owner wants to encrypt, the attribute set *S* and the set of time intervals T_S in which every element in T_S is associated with a corresponding attribute in *S*, this algorithm generates the ciphertext *CT* which is associated with the fuzzy attribute set *S*

3) KeyGen (MSK, Tr ,T'): This algorithm takes as input the master key MSK, the access tree Tr and the time set T'. Every attribute x in Tr is associated with a time instant tx ϵ T'. It outputs a private key SK which contains Tr.

4) Decrypt (CT, SK): This algorithm takes as input the ciphertext CT and the private key SK. When a set of time-specific attributes satisfies Tr, it is able to decrypt the ciphertext and return the plaintext M.

D. Auditing Model:

Auditing Model of this proposed approach uses two algorithms as: GenProof and VerifyProof.

1) GenProof: Now it comes to the "core" part of the auditing process. When TPA sends request to server then GenProof is run by the cloud server to generate a proof of data storage correctness.

2) VerifyProof: VerifyProof is run by the TPA to audit the proof after derive a response message from server.

E. Security Model

The security model is defined by the following games between an adversary *A* and a challenger *B*.

1) Initialization:

The adversary *A* declares the attribute set *Tr** that he wishes to be challenged upon.

2) Setup:

The challenger *B* runs the Setup algorithm to generate *params* and *MSK*. The *params* is given to *A*.

A generates repeated private keys corresponding to many access structures A_j and time instants in which none of these attribute structures satisfies that $Tr^* \epsilon A_j$.

3) Challenge:

A submits two equal-length messages *M0, M1,* and a challenge attribute set *Tr**. *B* flips a random coin *b,* and encrypts M_b under *Tr**. The ciphertext *CT** is given to *A*.

4) Guess:

A outputs a guess *b'* of *b.* The advantage of *A* in this game is defined as:

$$\mathrm{Adv_A} = \Pr[b' = b] \quad 1/2 \ .$$

Detail explanation of this model is referred from [1] for study.

VI. CONCLUSIONS

In this paper, a novel approach is projected which is able to achieve the time-specified ciphertext in order to resolve securely delete the outsourced data difficulties by introducing flexible fine-grained access control throughout the authorization phase and time-controllable self-destruction subsequently expiration to the shared and outsourced data stored on cloud computing. Additionally, this paper intends a privacy-preserving public auditing approach for data storage security in cloud computing using TPA. The TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage.

REFERENCES

[1] Jinbo Xiong, Ximeng Liu, Zhiqiang Yao, Jianfeng Ma, Qi Li, Kui Geng, and Patrick S. Chen, *A Secure Data Self-Destructing Scheme in Cloud Computing*, IEEEtrans on cloud computing, vol. 2, no. 4, oct-dec 2014.

[2] B. Wang, B. Li, and H. Li, *Oruta: Privacy-preserving public auditing for shared data in the cloud*, IEEEtrans. Cloud Computer, vol. 2, no. 1, pp. 43–56, Jan.–Mar. 2014.

[3] 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.

[4] J. Xiong, F. Li, J. Ma, X. Liu, Z. Yao, and P. S. Chen, *A full lifecycle privacy protection scheme for sensitive data in cloud computing*, Peer-to-Peer Network Appl., Jun. 2014.

[5] P. Jamshidi, A. Ahmad, and C. Pahl, *Cloud migration research: A systematic review*, IEEEtrans. Cloud Computing, vol. 1, no. 2, pp. 142–157, Jul.–Dec. 2013.

[6] R. Lu, H. Zhu, X. Liu, J. K. Liu, and J. Shao, *Toward efficient and privacy-preserving computing in big data era*, IEEE Networking, vol. 28, no. 4, pp. 46–50, Jul.-Aug. 2014.

[7] X. Liu, J. Ma, J. Xiong, and G. Liu, *Ciphertext-policy hierarchical attribute-based encryption for fine-grained access control of encryption data*, Int. J. Network Security, vol. 16, no. 4, pp. 351–357, 2014.

[8] K. G. Paterson and E. A. Quaglia, *Time-specific encryption*, in Proc. 7th Int. Conf. Security Cryptography Networks, 2010, pp. 1–16.

[9] J. Bethencourt, A. Sahai, and B. Waters, *Ciphertext-policy attribute-based encryption*, in Proc. 28th IEEE Symp. Security Privacy, 2007, pp. 321–334.

[10] B. Waters, *Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization*, in Proc. Public Key Cryptography, 2011, pp. 53–70.

[11] S. Yu, C. Wang, K. Ren, and W. Lou, *Achieving secure, scalable, and fine-grained data access control in cloud computing*, in Proc. 29th IEEE Int. Conf. Computer Communication, 2010, pp. 1–9.

[12] P. Tysowski and M. Hasan, *Hybrid attribute- and re-encryption based key management for secure and scalable mobile applications in clouds*, IEEEtrans. Cloud Computing, vol. 1, no. 2, pp. 172–186, Jul. 2013.

[13] J. Reardon, D. Basin, and S. Capkun, *Sok: Secure data deletion*, in Proc. 34th IEEE Symp. Security Privacy, 2013, pp. 1–15.

[14] C. Cachin, K. Haralambiev, H.-C. Hsiao, and A. Sorniotti, *Policy based secure deletion*, in Proc. ACM Conf. Computer Communication Security, 2013, pp. 152–167.

[15] J. Reardon, H. Ritzdorf, D. Basin, and S. Capkun, *Secure data deletion from persistent media*, in Proc. ACM Conf. Computer Communication Security, 2013, pp. 271–284.

[16] J. Xiong, Z. Yao, J. Ma, F. Li, and X. Liu, *A secure self-destruction scheme with IBE for the internet content privacy*, Chinese J. Computer, vol. 37, no. 1, pp. 139–150, 2014.

[17] R. Geambasu, T. Kohno, A. Levy, and H. M. Levy, *Vanish: Increasing data privacy with self-destructing data*, in Proc. 18th USENIX Security Symp., 2009, pp. 299–315.

[18] G. Wang, F. Yue, and Q. Liu, *A secure self-destructing scheme for electronic data*, J. Computer System Science, vol. 79, no. 2, pp. 279–290, 2013.

[19] S. Wolchok, O. S. Hofmann, N. Heninger, E. W. Felten, J. A. Alderman, C. J. Rossbach, B. Waters, and E. Witchel, *Defeating vanish with low-cost sybil attacks against large DHTS*, in Proc. 17th Annu. Netw. Distributed Syst. Security Conf., 2010, pp. 1–15.

[20] L. Zeng, S. Chen, Q. Wei, and D. Feng, *Sedas: A self-destructing data system based on active storage framework*, IEEEtrans. Magnetics, vol. 49, no. 6, pp. 2548–2554, Jun. 2013.

[21] J. Xiong, Z. Yao, J. Ma, X. Liu, and Q. Li, *A secure document selfdestruction scheme: An abe approach*, in Proc. 15th IEEE Int. Conf. High Perform. Computer Communication, 2013, pp. 59–64.

[22] J. Xiong, Z. Yao, J. Ma, F. Li, X. Liu, and Q. Li, *A secure selfdestruction scheme for composite documents with attribute based encryption*, Acta Electronica Sinica, vol. 42, no. 2, pp. 366–376, 2014.

[23] R. Kikuchi, A. Fujioka, Y. Okamoto, and T. Saito, *Strong security notions for timed-release public-key encryption revisited*, in Proc. Inf. Security Crypto., 2012, pp. 88–108.

[24] K. Kasamatsu, T. Matsuda, K. Emura, N. Attrapadung, G. Hanaoka, and H. Imai, *Time-specific encryption from forwardsecure encryption*, in Proc. 8th Int. Conf. Security Crypto. Netw., 2012, pp. 184–204.