Data Sharing and Self-Destruction Scheme in Cloud

Mohammadi Kamplee
ME Student
Department of CSE
NBNSCOE
Solapur, India

Babruvan Solunke
Guide
Professor
Department of CSE
NBNSCOE
Solapur, India

ABSTRACT
There is a huge use of cloud services in our day today life e.g. Google Drive, Dropbox, SharePoint etc. Sharing data within friends might consist of sensitive/personal information. It's always the users responsibility to safeguard own data while sharing and avoid misuse of it. It becomes a challenge for user to protect self-data on cloud network, to overcome this scenario it is important to design and allocate self-destruct period assigned by the user and access control to the data until the expiry period. The shared data should be self-destructed after the user-defined expiration time. With the help of KPABE (Key-policy ABE) and where can apply time interval to each attribute in the form of decryption attributes. In the KP-TSABE scheme, every cipher text is labeled with a time interval while private key is associated with a time instant. Deletion of data in a secure way is the task of deleting data irrecoverably from a physical storage medium. In this digital world, data is not securely deleted by default; instead, many approaches add secure deletion to existing physical medium interfaces.

General Terms
Attribute Based Algorithm, Dynamic Data structure creation, SQL injection Prevention.

Keywords
ABE, KP-ABE, CP-ABE, TSE, TRE, DTI, KP-STABE, Proxy Re-encryption, RBAC.

1. INTRODUCTION
CLOUD technology is the next big leap forward in the field of Information Technology, which is derived with Service architecture and Virtual environment. It's in common use to share data with help of cloud offering services with other users, friends etc. E.g. Such as Google Drive, Drop box etc. Shared data might consist of sensitive or personal information of the user sharing the data or vice versa (e.g. Profile information, health or property records, etc.). It's always the users responsibility to safeguard own data and avoid misuse of it. It becomes a challenge for user to protect self-data on cloud network, to overcome this scenario it is important to design and allocate self-destruct period assigned by the user and access control to the data until the expiry period. The shared data should be self-destructed after the user-defined expiration time. Basic can be to store the data in encrypted format but disadvantage with classic encryption is the owner should know what information the users wants to share and with whom this makes the process to sharing the data to many a bit hectic. To overcome this disadvantage of ABE (Attribute based encryption) which enables one to many encryptions.

ABE has the ability to provide data security as well as access control to the minimum level. Also have Timed-release encryption (TRE) which provides encryption service based on Time as variable, where an encryption key is associated with a predefined lease time, and an authorized receiver can construct the corresponding decryption key in this time instance. On this basis, Paterson et al. designed a time specific encryption (TSE) scheme, which is able to specify a suitable time interval such that the cipher text can only be decrypted in this interval (decryption time interval, DTI). ABE has issues with Time Constrains whereas TSE has problems with Access Control both these issues can be addressed with the help of KPABE (Key-policy ABE) and where can apply time interval to each attribute in the form of decryption attributes.

2. RELATED WORK
As cloud is a common platform for sharing and storing data, there are number of tools and techniques developed for both the users as well as data crawlers for public domain to audit the cloud data without hampering the actual data this breaches the data integrity on the cloud server. KPABE identity user of each block and shared data is kept private from public users, which is only used by authorized user and it efficiently verifies shared data, integrity is preserved without retrieving the entire file. In addition, our KPABE mechanism is able to perform multiple auditing tasks simultaneously instead of verifying them one by one.

Here considering the problem of sending messages in the future, commonly known as timed release Cryptography in Existing schemes for this task either solve the relative time problem with uncontrollable, coarse-grained release time (time-lock puzzle approach) or do not provide anonymity to senders and/or receivers and are not scalable (server-based approach). Using a bilinear pairing on any Gap Diffie-Hellman group, solving this problem by giving scalable, server-passive and user anonymous timed release public-key encryption schemes allowing precise absolute release time specifications. In previous server-based schemes, there was the trusted time server which was completely passive there was no interaction between it and the sender or receiver is needed actively; and not even aware of all the existence of a user, thus assuring the Privacy of a message and the anonymity of both its sender and receiver. Besides, this scheme also has a number of desirable properties including a single form of update for all users, self-authenticated time-bound key updates, and key insulation, making it a scalable and appealing solution. It could also be easily generalized to a more general policy lock mechanism.

To provide basic protection for the integrity of the data user should have a set of credentials or attributes like public key or private key. This can be achieved when storing the data on a single central server provided with access control from where user can access his\her related data, but if the central data server is compromised then this system fails on the data availability and integrity standards. For this there is the system for realizing complex access control on encrypted data that call Cipher text-

Policy Attribute-Based Encryption. The techniques of encrypted data can be kept secret even if the storage server is untrusted; moreover, these methods are secure against attacks. Previous Attribute Based Encryption systems used attributes to describe the encrypted data and built policies into user's keys; while in this system attributes are used to describe a user's credentials, and a party encrypting data determines a policy for who can decrypt. Thus, these methods are conceptually closer to traditional access control methods such as Role-Based Access Control (RBAC). Every secret key in Cipher text policy attribute-based encryption (CP-ABE) is associated with a set of attributes, and every cipher text is associated with an access structure on attributes. Decryption on the other end can only be performed if the user's attribute set satisfies the cipher text access requirements. This provides a basic requirement of access control on shared data in many practical scenarios.

Cloud computing is an emerging market as it provides scalability and infrastructure as service over the internet, this is helpful for many IT companies but it also addresses new challenges for the organizations in form or data security as all the users data rely on the untrusted cloud domain. To maintain the integrity of the user's data simple cryptography methods of only sharing decrypting keys only with authorized users can help. But these will not work when considering a bigger scenario when the distribution list might belong to an organization and not some users. Simple cryptography fails in terms of scalability and requirement of achieving fine-graininess, scalability, and data confidentiality of access control actually still remains.

So on one part it is enforcing access policies based on data attributes and on secondary part allowing the owner of data to delegate most computing tasks of data access control to the cloud domain without transferring the actual data. This is achieve by combining techniques of attribute-based encryption (ABE), proxy re-encryption, and lazy re-encryption. This scheme has salient features of user access privilege confidentiality and user private key accountability.

Migrating data to the cloud is useful in terms of economy, scalability, and accessibility, but important technical challenges remain unattended. Sensitive data stored in the cloud must be protected from being read in the clear by a cloud service provider that is honest-but-curious. Cloud-based data is used heavily and being accessed by resource constrained mobile devices for which the processing and communication cost must be minimized. Innovative modifications to attribute-based encryption are designed to allow authorized users access to cloud data based on the requirements of the attributes such that the higher computational load from cryptographic operations is assigned to the cloud service provider and the overall communication cost is lowered for the end user. Furthermore, data re-encryption may be optionally performed by the cloud provider to reduce the expense of user revocation in a mobile user environment while preserving the privacy of user data stored in the cloud.

Deletion of data in a secure way is the task of deleting data irrecoverably from a physical storage medium. In this digital world, data is not securely deleted by default; instead, many approaches add secure deletion to existing physical medium interfaces. Interfaces to the physical medium exist at different layers, such as user-level applications, the file system, the device driver, etc. Depending on which interface is used, the properties of an approach can differ significantly. Related work in detail and organize existing approaches in terms of their interfaces to physical media. Further present taxonomy of adversaries differing in their capabilities as well as systematization for the characteristics of secure deletion approaches. Characteristics include environmental assumptions, such as how the interface's use affects the physical medium, as well as behavioral properties of the approach such as the deletion latency and physical wear. We perform experiments to test a selection of approaches on a variety of file systems and analyze the assumptions made in practice.

A general approach to the design and analysis will be the solution for secure deletion for persistent storage that relies on encryption and key wrapping. Defined a key disclosure graph models of the adversarial knowledge of the history of key generation and wrapping. Making use of a generic update function and prove that it achieves secure deletion of data against a coercive attacker; instances of the update function implement the update behavior of all data structures including B-Trees, extendible hash tables, linked lists, and others. This implementation is at the lowest level of data storage i.e. block-device layer, allowing any block-based file system to be used on top of it. Using different workloads, found that the storage and communication overhead required for storing and retrieving B-Tree Nodes is small and that this therefore constitutes a viable solution for many applications requiring secure deletion from persistent media.

3. ENCRYPTION SCHEMES AND TECHNIQUE USED:

3.1 Attribute-based encryption

Attribute-based encryption is one of the important applications of fuzzy identity-based encryption. ABE comes in two flavors called KP-ABE and cipher text-policy ABE (CP-ABE).In CP-ABE, the cipher text is associated with the access structure while the private key contains a set of attributes. Bethencourt et al. proposed the first CPABE scheme , the drawback of their scheme is that security proof was only constructed under the generic group model. To address this weakness, Cheung et al. presented another construction under a standard 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 cipher text contains a set of attributes and the private key is related to the access structure. The first construction of KP-ABE scheme was proposed in. In their scheme, when a user made a secret request, the trusted Authority determined which combination of attributes must appear in the cipher text for the user to decrypt. Instead of using the Shamir secret key technique in the private key, this scheme used a more generalized form of secret sharing to enforce a monotonic access tree. Ostrovsky et al. presented the first KP-ABE system which supports the non-monotone formulas in key policies. Yu et al. used a combining technique of KP-ABE, proxy re encryption, and lazy re-encryption which allows the data owner to delegate most of the computation tasks involved in fine-grained data access control to untrusted cloud servers without disclosing the underlying data contents. Tysowski et al. modified the ABE and leveraged re-encryption algorithm to propose a novel scheme to protect mobile user's data in cloud computing environment. Due to the lack of time constraints, the above-mentioned ABE schemes do not support user-defined authorization period and secure self-destruction after expiration for privacy-preserving of the data lifecycle in cloud computing.

3.2 Secure self-destruction scheme

A well-known method for addressing this problem is secure deletion of sensitive data after expiration when the data was used. Recently, Caching et al. employed a policy graph to describe the relationship between attributes and the protection

class and proposed a policy-based secure data deletion scheme. Reardon et al. leveraged the graph theory, Btree structure and key wrapping and proposed a novel approach to the design and analysis of secure deletion for persistent storage devices. 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 . A data self-destructing scheme, first proposed by Geambasu et al., 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) . In the SSDD scheme, a data is encrypted into a cipher text, which is then associated and extracted to make it incomplete to resist against the traditional cryptanalysis and the brute-force attack. Then, both the decryption key and the extracted cipher text are distributed into a distributed hash table (DHT) network to implement self-destruction after the update period of the DHT network. However, 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. So the security of the SSDD scheme is also questionable. To address this problem, Zeng et al. proposed a SeDas system, which is a novel integration of cryptographic techniques with active storage techniques. Xiong et al. leveraged the DHT network and identity-based encryption (IBE) and proposed an IBE-based secure self-destruction (ISS) scheme. In order to protect the confidentiality and privacy security of the composite documents within the whole lifecycle in cloud computing, Xiong et al. applied the ABE algorithm to propose a secure self-destruction scheme for composite documents. Recently, 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 (Full PP), 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. The main idea of the above-mentioned schemes is that they respectively combine different cryptographic techniques with the DHT network to provide fine-grained data access control during the lifecycle of the protected data and to implement data self-destruction after expiration. However, using of the DHT network will result in the fact that the lifecycle

3.3 Time-specific encryption

The time-specific encryption scheme TSE, proposed by Peterson et al., was introduced as an extension of TRE. In TRE, a protected data can be encrypted in such a way that it cannot be decrypted (even by a legitimate receiver who owns the decryption key for the cipher text) until the time (called the release-time) that was specified by the encryptor. Most of the previous TRE schemes that adopt a time-sever model are in fact public-key TRE schemes. They do not consider the sensitive data privacy after expiration. In the TSE scheme, a time sever broadcasts a time instant key (TIK), a data owner encrypts a message into a cipher text during a time interval, and a receiver can decrypt the cipher text if the TIK is valid in that interval. Kasamatsu designed an efficient TSE scheme by using forward-secure encryption (FSE) in which the size of the cipher text is greatly small than that generated by the previous schemes. The time interval may be considered as the authorization period of the protected data, and TSE schemes are able to meet this requirement. However, it is a tricky problem when the traditional TSE is used in the cloud computing environment: cloud computing environment needs a fine-grained access control, which cannot be provided by the traditional TSE schemes. How to achieve the time-specified cipher text into a fine-grained access control level is a problem to be explored.

3.4 System Analysis

Sharing data among users is perhaps one of the most important features that motivate cloud storage. So variability of files, there are a series of encryption techniques which are used as far as allowing a third-party auditor to check the availability of files on behalf of the data owner without revealing data, or without compromising the data owner's anonymity.

Disadvantages of Existing System:

- Data Privacy issues
- Large Amount of space need in Cloud storage
- Calculation overhead at user's side for encryption and decryption.

3.5 Proposed System:

Proposed KP-TSABE scheme, which is a novel in securing data and automatically self-destructing in cloud computing. In this KP-TSABE scheme, every cipher text is labeled with a time interval, while private key is associated with a time instant. The cipher text can only be decrypted if both attributes such as the time instant is in the allowed time interval and the attributes associated with the cipher text satisfy the key's access structure.

Advantages of Proposed System:

- Security issue is addressed.
- Privacy issues are minimized.
- Reducing the space required to store data in cloud.

4. SYSTEM ARCHITECTURE

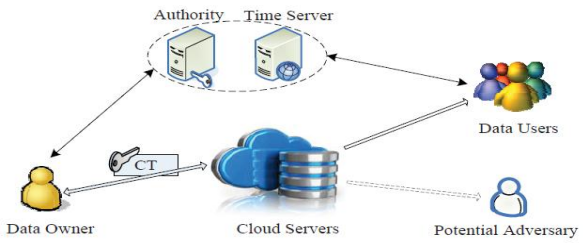

Fig.1 System Architecture

4.1 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.

4.2 Authority:

It is an indispensable entity which is responsible for generating, distributing and managing all the private keys, and is trusted by all the other entities involved in the system.

4.3 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.

4.4 Data Users:

Data users are some peoples 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.

4.5 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 server.

4.6 Potential Adversary:

Can either be end user or a challenger.

5. COMPARISION WITH EXSITING SYSTEM

Following table will demonstrate the security properties of existing system with the prosed system:

Table:1 Comparison between existing system with KP-STABE

Security Properties	Vanish	SSDD	KP-STABE
Attack on VDO	Yes	Yes	No need
Type of Algorithm	Symmetric	Symmetric	KP-STABE
Cipertext Destruction	No	Yes	No need
Key Destruction	Yes	Yes	No need
Resistance on cryptanalysis	No	No	Yes
Supporting FGAC	No	No	Yes
Privacy protection	No	No	Yes

6. IMPLEMENTATION DETAILS

Initially, the user choose a security parameter κ and attribute universe U, which calls the algorithm Setup(1_, U) belonging to the algorithm level to create system parameter params and master key MSK.

The user chooses an attribute set S for the shared message M and set a time interval TS for S. Then, the data owner calls given algorithm as Encrypt(M, params, S, TS); this function encrypt M to its ciphertext CT which is sent to cloud servers. This CT has its associated with the set S and TS.

When a user wants to access the shared data M during its authorization period, he must have authorization and should execute the following processes:

Primarily, the current time instant tx is provided by the time server with tx ∈ T′, which is associated with each attribute x. If T′ ⊆ TS and the attribute set of the user matches the access tree Y. Then, the Authority runs the below algorithm KeyGen(MSK,Y, T′); that generates the private key SK and sends it to the user.

Once the user received the SK, he will get the CT from the cloud servers and invokes the algorithm Decrypt(CT, SK); to decrypt CT to obtain the shared data M. Because each attribute x is associated with a current time instant tx, if and only if tx ∈ TS and attribute set matches Y, the user can obtain the correct private key SK to decrypt CT. Therefore the KP-TSABE scheme allows flexible execution of fine-grained access control through merging different attributes with corresponding time intervals.

Once the current time instant tx grow into the threshold value (expiration time) of the valid time interval tR; x, the user will be unable to have private key SK. Therefore, the ciphertext CT will be unable to decrypted in polynomial time, enabling self-destruction of the shared data after expiration

7. CONCLUSION AND FUTURE SCOPE

Intensive use and development of versatile cloud services, a lot of new challenges have emerged. One of the most important issues is how to securely curb or delete the outsourced data stored in the cloud severs. In this paper, proposed method is a novel KP-TSABE scheme which is able to achieve the time-specified cipher text 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. Also gave a system model and a security model for the KPTSABE scheme. The comprehensive analysis indicates that the proposed KP-TSABE scheme is superior to other existing schemes.

Since this project is all about sharing files, what have aimed and achieved creating is not a product but a tool to a better automotive environment, a tool can be used to shape many things in the future, thus this project will give rise to many future modifications forking in all directions. Some of the near future scopes of this project are as follows. There are few interesting problems will continue to study for our future work. One of them is can share a file to multi users at a time. Also used AES (Advanced Encryption Scheme) to encrypt the Data

REFERENCES

[1] B. Wang, B. Li, and H. Li, “Oruta: Privacy--preserving public auditing for shared data in the cloud,” Cloud Computing, IEEE Transactions on, vol. 2, no. 1, pp. 43–56, 2014.

[2] J. Xiong, Z. Yao, J. Ma, X. Liu, Q. Li, and J. Ma, “Priam: Privacy preserving identity and access management scheme in cloud,” KSII Transactions on Internet and Information Systems (TIIS), vol. 8, no. 1, pp. 282–304, 2014.

[3] 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,” Peerto- Peer Networking and Applications[Online].Available:http://dx.doi.org/10.1007/s 12083-014-0295-x

[4] R. Ostrovsky, A. Sahai, and B. Waters, “Attribute-based encryption with non-monotonic access structures,” in Proceedings of the 14th ACM Conference on Computer and Communications Security. ACM, 2007, pp. 195–203.

[5] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving secure, scalable, and fine-grained data access control in cloud computing,” in Proceedings of the 29th IEEE International Conference on Computer Communications. IEEE, 2010, pp. 1–9.

[6] P. Tysowski and M. Hasan, “Hybrid attribute- and reencryption-based key management for secure and scalable mobile applications in clouds,” Cloud Computing, IEEE Transactionson, vol. 1, no. 2, pp. 172–186, 2013.

[7] J. Reardon, D. Basin, and S. Capkun, “Sok: Secure data deletion,” in Proceedings of the 34th IEEE Symposium on Security and Privacy. IEEE, 2013, pp. 1–15.

[8] C. Cachin, K. Haralambiev, H.-C. Hsiao, and A. Sorniotti, “Policy-based secure deletion,” in Proceedings of the ACM

Conference Computer and Communications Security. ACM, 2013, pp. 152–167.

[9] J. Reardon, H. Ritzdorf, D. Basin, and S. Capkun, "Secure data deletion from persistent media," in Proceedings of the 2013 ACM Conference on Computer and Communications Security. ACM, 2013, pp. 271–284. [22] J. Xiong, Z. Yao, J. Ma, F. Li, and X. Liu, "A secure selfdestruction scheme with ibe for the internet content privacy," Chinese Journal of Computers, vol. 37, no. 1, pp. 139–150,2014.

[10] R. Geambasu, T. Kohno, A. Levy, and H. M. Levy, "Vanish: Increasing data privacy with self-destructing data," in Proceedings of the 18th USENIX Security Symposium, 2009, pp. 299–315.

[11] G. Wang, F. Yue, and Q. Liu, "A secure self-destructing scheme for electronic data," Journal of Computer and System Sciences, vol. 79, no. 2, pp. 279–290, 2013.

[12] S. Wolchok, O. S. Hofmann, N. Heninger, E. W. Felten, J. A. Halderman, C. J. Rossbach, B. Waters, and E. Witchel, "Defeating vanish with low-cost sybil attacks against large dhts," in Proceedings of the 17th Annual Network and Distributed System Security Conference, NDSS. ISOC,2010, pp. 1–15.

[13] L. Zeng, S. Chen, Q. Wei, and D. Feng, "Sedas: A selfdestructing data system based on active storage framework," 2168-7161 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

[14] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," SIAM Journal on Computing, vol. 32, no. 3, 586–615, 2003. P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud migration research: A systematic review," Cloud Computing, IEEE Transactions on, vol. 1, no. 2, pp. 142–157,2013.