ISSN 2395-1621

Client Server Based Robust Data Self Extraction Strategy

[#1]Rupali Welekar, [#2]Komal Jain, [#3]Sarika Karpe, [#4]Pournima More

[1]welekarroop10@gmail.com
[2]komairaj142@gmail.com
[3]sarikahargude700@gmail.com
[4]pournima.more1@gmail.com

[#1234]Department of Computer Engineering,
G.H. Raisoni College Of Engineering And Management,Pune

ABSTRACT

In this today's world, important data is stored and shared on the internet. Cloud computing is one of the great technologies used for handling large amount of data and its storage. The Cloud verifies the believability of the server without knowing the users identity before storing the information. Our system also has the added feature of access control in which only legal users are able to decode the information which they stored. Access control procedure protect sensible information from unauthorized users. Hence when sensitive information is shared and privacy protection execution is not in place and authorized users can still arrange the privacy of a person ruling to identity disclosure. So we are likely presenting a key policy attribute based encryption with time specified attribute and a secure data self disaster strategy in cloud computing which satisfies the security necessity and is preferable to other existing strategy.

Keywords: **Sensible data, secure self-destructing, fine-grained access control, privacy-preserving, Server.**

ARTICLE INFO

Article History

Received :8th March 2016

Received in revised form :

10th March 2016

Accepted : 13th March 2016

Published online :

18th March 2016

I. INTRODUCTION

There is a way for sensitive user data to be stored by third parties on the Internet. For example, personal email, data, and personal preferences are stored on web sites such as Google and Yahoo. Given the variety, amount, and importance of information stored at these sites, there is reason for concern that personal data will be compromised. The shared data which is stored servers which contains users sensitive data such as personal profile, financial data , health records etc. So it needs to be well secured as the owner of data is detached from the organization of the owner. The servers may migrate user's data to other servers in outsourcing or share them in cloud searching. And so it becomes a big test to overprotect the privacy of those shared data in cloud. Hence it is necessary to plan a super solution to support user defined authorization time and to offer fined grained access control during this period. The shared data should be self destruct after the user defined termination time.One of the procedure to solve this problem is to store data in common encrypted form. The data owner wants to share information with many users according to the security policy which is based on the users documents. Attribute Based Encryption 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 add to in this paper, the cipher text is labelled with set of descriptive attributes satisfies the access structure in the key, the user can get the plain text.The owner has the right to specify the that certain sensitive data is only valid for a limited period of time or should not be released before a particular time,Time Release Encryption (TRE) provides an fascinating encryption service where an encryption key is associated with a predefine release time. And a receiver can only construct the equivalent decryption key in this time instance.However applying the ABE to the shared data will introduce various problems with regard to time specific constraint and self destruction while applying the Time Specific Encryption will introduce the problems with regard

to fine grained access control.Thus, in this paper we try to solve this problem by using KP-ABE and adding a constraint of time interval to each attribute in the set of decryption attribute.

II. RELATED WORKS

2.1 Attribute-based encryption

Encryption is process of converting plain text into the cipher text. Attribute-based encryption is one of the important applications.

ABE comes in two types

called Key Policy Attribute Based Encryption (KP-ABE) and Cipher Text-Policy ABE (CP-ABE).The only difference between the KP-ABE and CP-ABE is that in CP-ABE, the cipher text is associated with the access structure while the private key contains a set of attributes. And in KP-ABE the cipher text contains a set of attributes and the personal key is related to the access structure. The sensitive data is shared and stored by third-party sites on the Server, there will be a need to encrypt data which is stored .

One drawback of encrypting data, is that it can be selectively shared only at a rough-grained level (i.e., giving another party your private key). We introduce a new strategy for fine-grained sharing of encrypted data that we call Key-Policy Attribute-Based Encryption (KP-ABE). In our strategy, cipher texts are labelled with sets of attributes and private keys are associated with access structures that control which cipher texts a user is able to decode. In an ABE system, a user's keys and cipher texts are tagged with format of descriptive attributes and a particular key can decrypt a particular cipher text only if there is a similarity between the attributes of the cipher text and the user's key. Due to the lack of time constraint, the above named 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.

2.2 Time-specific encryption

A timed-release cryptosystem allows a sender to encipher a message so that only the intended recipient can read only after a specified time. We explain the concept of a secure timed-release public-key cryptosystem and show that, if a third party is call upon to guarantee decryption after the specified date, this concept is equivalent to identity-based encryption; this explains the measure that all known constructions use identity-based encryption to achieve timed-release security.The time-specific encryption scheme TSE, was introduced as an extension of TRE . In TRE, a protected data can be encrypted in such a way that it cannot be decrypted until the release-time that was specified by the encryptor. They do not consider the sensitive data privacy after expiration . In the Time Specific Encryption 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. The time interval can be considered as the authorization period of the protected data which is shared between clients and server.

2.3 Secure self-destruction scheme

Securely data self extraction strategy is one of the method to provide the security. And again secure deletion of sensitive data after expiration when the data was used. A data self-destructing scheme, a promising progress which designs a Vanish system allows users to control over the lifecycle of the sensitive data. We proposed a secure self-destructing scheme for electronic data (SSDD) . In the SSDD scheme, a data is encrypted into a cipher text. 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. Again to protect the important documents or files which is shared between he servers and client we introduce the ABE algorithm to propose a secure self-destruction scheme for composite documents (SelfDoc). Recently, we employed identity-based timed-release encryption (ID-TRE) algorithm and the DHT network and proposed a full lifecycle privacy protection scheme for sensitive data , which is able to provide full lifecycle privacy security for users' important data by making it unreadable before a predefined time and automatically destructed after expiration.

III. MOTIVATION

As the secure self-destruction scheme, both SSDD and FullPP have some limitations. First, SSDD does not consider the issue of the demand release time of the sensitive data, the expiration time of both SSDD and FullPP strategy is limited by the DHT network and cannot be determined by the user. Second, SSDD and many other strategy are dependent on the ideal possibility of "No attacks on VDO (vanishing data object) before it expires".Third, it is incontestable that the Vanish scheme is dangerous to the Sybil attacks from the DHT network, the SSDD scheme and other schemes are same. As a result, unauthorized users can freely approach to the sensitive data and this defect would lead to a serious privacy disclosure. To solve these problems, in this paper, we declare a novel solution called key-policy attribute based encryption with time-specified attributes (KPTSABE) strategy, which is based on our observation that, in practical cloud application script, each data item can be associated with a set of attributes and every attribute is associated with a specification of time interval (decryption attribute time interval, DATI), e.g., [09:00,17:00], denoting that the encrypted data item can only be decoded between 09:00 to 17:00 on a specified date and it will not be retrievable before 09:00 and after 17:00 that day. The data owner encipher his/her data to share with users in the system, in which every users key is relate with an access tree and each leaf node is associated with a time instant, e.g., 14:30. As the logical expression of the access tree can correspond to any desired data set with any time interval, it can achieve small-grained access control. If the time instant is not in the nominal time interval, the cipher text cannot be decoded, i.e., this cipher text will be self-destroyed and no one can decode it because of the termination of the procure key. Therefore,

secure information self-destruction with fine-grained access control is succeed.

IV. CONTRIBUTIONS

We propose a KP-TSABE strategy, which provide secure self-extraction strategy for data sharing between server and client. We first present the notion of KP-TSABE, formalize the model of KP-TSABE and give the security model of it. Then, we give a specific construction method about the scheme. Lastly we will prove that the KP-TSABE scheme is secure. Especially, KP-TSABE has the following advantages with regard to security and fine-grained access control over the other secure self-destructing schemes.

1. It 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.

2. It does not require the ideal assumption of "No attacks on VDO before it expires".

3. KP-TSABE is able to apply fine-grained access control during the authorization period and can self destroyed the data without any human intervention.

V. CONCEPTS & MODELS

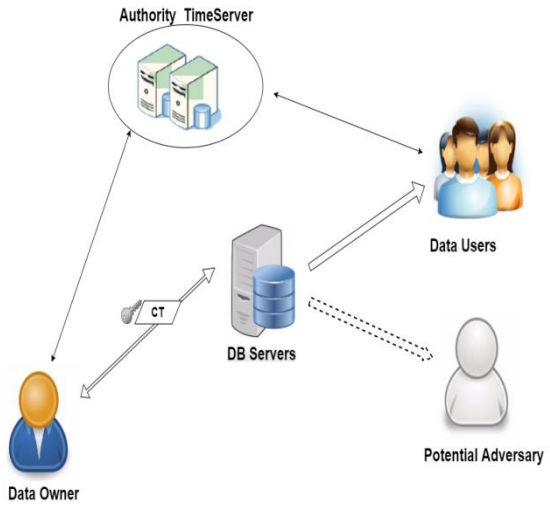
5.1Concepts

To form a basis for the KP-TSABE scheme, we introduce the following concepts.

(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. The ciphertext is associated with this interval; the user can construct the decryption key only when the time instant is within this interval.
(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. This paper provides full lifecycle privacy protection for shared data in cloud computing.

VI. SYSTEM ARCHITECTURE

In our system, we 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. Specifically, we define the system model by dividing the KPTSABE scheme into the following six entities as -



1) ***Data Owner*.** Data owner can provide data or files that contain some sensitive information, which are used for sharing with data users. All these shared data are outsourced to the servers to store.

(2) ***Authority*.** It is an indispensable entity which is responsible for generating, distributing and managing all the private keys.

(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 peoples who passed the identity authentication and access to the data outsourced by the data owner.
(5)***DB Server*.** It contains storage space which is able to store and manage all the data or files in the system.

(6) ***Potential Adversary*.** It is a polynomial time adversary.

VII. ALGORITHM

SetUp:
KGC takes as input a security parameter k to generate two primes p and q such that $q|p-1$. It then performs the following steps:
1) Pick a generator g of $Z*p$ with order q.
2) Select $x \in Z*q$ uniformly at random and compute
$y = gx$.
3) Choose cryptographic hash functions
$H1:\{0,1\}*\times Z*p \rightarrow Z*q$, $H2:\{0,1\}*\times Z*p\times Z*p\rightarrow Z*q$,$H3:\{0,1\}* \rightarrow Z*q$, $H4:Z*p\rightarrow \{0,1\}n+k0$,$H5:Z*p\rightarrow \{0,1\}n+k0$, and $H6:Z*p\times \{0,1\}n+k0 \times Z*p\times \{0,1\}n+k0 \rightarrow Z*q$,
where n, $k0$ are the bit-length of a plaintext and a random bit string, respectively.
The system parameters *params* are $(p, q, n, k0, g, y, H1, H2, H3, H4, H5, H6)$. The master key of KGC is x. The plaintext space is $M = \{0,1\}n$ and the ciphertext space is $C = Z*p \times \{0,1\}n+k0 \times Z*q$.

• **Set PrivateKey**:
The entity **A** chooses $zA \in Z *q$ uniformly at random
as the private key of **A**. • SetPublicKey: The entity **A** computes $UA = gzA$. • SEM-KeyExtract:
KGC selects $s0$, $s1 \in Z *q$ and computes $w0 = gs0$, $w1 = gs1$, $d0 = s0 + xH1\ (IDA,w0)$, $d1 = s1 +$
$xH2\ (IDA,w0,w1)$. KGC sets $d0$ as the SEM-key for **A**. After **A** proves the knowledge of the secret value
zA such that $UA = gzA$, KGC sets $(UA,w0,w1, d1)$ as the **A**'s public keys.
• Encrypt:
To encrypt a plaintext $M \in \{0, 1\}n$ for the entity **A** with identity IDA and public keys $(UA,w0,w1, d1)$, it
performs the following steps: 1) Check whether $gd1 = w1 \cdot yH2(IDA,w0,w1)$. If the checking result is not valid, encryption
algorithm must be aborted.
2) Choose $\sigma \in \{0, 1\}k0$ and
compute $r = H3\ (M, \sigma, IDA,UA)$.
3) Compute $C1 = gr$.
4) Compute $C2 = (M||\sigma) \oplus H4\ \_UA$
$r\_ \oplus H5\ \_wr$
0
.
$yH1(IDA,w0)\cdot r\_$.
5) Compute $C3$
$C3 = H6\ \_UA, (M||\sigma) \oplus H4\ \_UrA$
$\_ ,C1,C2\_$.

Output the ciphertext $C = (C1,C2,C3)$

VIII. 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 severs. we proposed a novel KP-TSABE strategy 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.We also gave a system model and a security model for the KPTSABE scheme. Furthermore, we proved that KPTSABE is secure under the standard model with the decision *l*-Expanded BDHI assumption. The comprehensive analysis indicates that the proposed KP-TSABE scheme is superior to other existing schemes.

REFERENCE

[1] "Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds" IEEE Transactions On Parallel And Distributed Systems VOL:25 NO:2 YEAR 2014

[2] "Accuracy Constrained Privacy-Preserving Access Control Mechanism for Relational Data" IEEE Transactions On Knowledge And Data Engineering, VOL. 26, NO. 4, APRIL 2014

[3] "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage" IEEE Transactions on Parallel and Distributed Systems. VOL: 25, ISSUE: 2. YEAR :2014.

[4] Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data.