International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 3 | Mar 2019 www.irjet.net p-ISSN: 2395-0072

A LIGHTWEIGHT PROTECTED DATA SHARING METHOD FOR MOBILE CLOUD COMPUTING

Suganaya. S[1] and Dr. S. Chandrasekaran[2]

suganyasundharesan@gmail.com and chandrudpi@gmail.com
[1]PG Student and [2]Assistant Professor, Department of Computer Science and Engineering
P.S.V. College of Engineering & Technology, Krishnagiri

Abstract - *With the popularity of distributed computing, cell phones can store/recover individual information from anyplace whenever. Subsequently, the information security issue in portable cloud turns out to be increasingly serious and forestalls further advancement of versatile cloud. There are significant examinations that have been led to improve the cloud security. Nonetheless, the vast majority of them are not material for portable cloud since cell phones just have restricted registering assets and power. Arrangements with low computational overhead are in incredible requirement for portable cloud applications. In this paper, we propose a lightweight information sharing plan (LDSS) for portable distributed computing. It embraces CP-ABE, an entrance control innovation utilized in typical cloud condition, yet changes the structure of access control tree to make it appropriate for portable cloud situations. LDSS moves an expansive segment of the computational serious access control tree change in CP-ABE from cell phones to outer intermediary servers. Besides, to lessen the client denial cost, it acquaints trait portrayal fields with actualize sluggish repudiation, which is a prickly issue in program based CP-ABE frameworks. The exploratory outcomes demonstrate that LDSS can successfully lessen the overhead on the cell phone side when clients are sharing information in versatile cloud situations.*

Key Words: *mobile cloud computing, data encryption, access control, user revocation*.

1. INTRODUCTION

The development of cloud computing and the popularity of smart mobile devices, people are gradually getting accustomed to a new era of data sharing model in which the data is stored on the cloud and the mobile devices are used to store/retrieve the data from the cloud. Typically, mobile devices only have limited storage space and computing power. On the contrary, the cloud has enormous amount of resources. In such a scenario, to achieve the satisfactory performance, it is essential to use the resources provided by the cloud service provider (CSP) to store and share the data.
Nowadays, various cloud mobile applications have been widely used. In these applications, people (data owners) can upload their photos, videos, documents and other files to the cloud and share these data with other people (data users) they like to share. CSPs also provide data management functionality for data owners.

The state-of-the-art privilege management/access control mechanisms provided by the CSP are either not sufficient or not very convenient. First, when people upload their data files onto the cloud, they are leaving the data in a place where is out of their control, and the CSP may spy on user data for its commercial interests and/or other reasons. Second, people have to send password to each data user if they only want to share the encrypted data with certain users, which is very cumbersome. To simplify the privilege management, we propose a Lightweight Data Sharing Scheme (LDSS) for mobile cloud computing environment. The main contributions of LDSS are as follows:

(1) We design an algorithm called LDSS-CP-ABE based on Attribute-Based Encryption (ABE) method to offer efficient access control over ciphertext.

(2) We use proxy servers for encryption and decryption operations. In our approach, computational intensive operations in ABE are conducted on proxy servers, which greatly reduce the computational overhead on client side mobile devices.

Meanwhile, in LDSS-CP-ABE, in order to maintain data privacy, a version attribute is also added to the access structure. The decryption key format is modified so that it can be sent to the proxy servers in a secure way.

(3) We introduce lazy re-encryption and description field of attributes to reduce the revocation overhead when dealing with the user revocation problem.

(4) Finally, we implement a data sharing prototype framework based on LDSS. The experiments show that LDSS can greatly reduce the overhead on the client side, which only introduces a minimal additional cost on the server side. Such an approach is beneficial to implement a realistic data sharing security scheme on mobile devices. The results also show that LDSS has better performance compared to the existing ABE based access control schemes over ciphertext. The rest of this paper is organized as follows. Section 2 presents some fundamental concepts in secure mobile cloud data sharing and the security premise. Section 3 gives the detailed design of LDSS. Section 4 and 5 give the safety assessment and performance evaluation, respectively. Section 6 presents related works. Finally, Section 7 concludes our work with the future work.

2 PRELIMINARIES AND ASSUMPTIONS

In this section, we first briefly present the technique preliminaries closely related to LDSS, and then present the system model and some security assumptions in LDSS.

2.1 Preliminary Techniques

2.1.1 Bilinear Pairing

In our implementation, we usually take as a group consisting points on an elliptic curve, as a multiplicative subgroup of a finite field, *e* as a Weil or the Tate pairing based on an elliptic curve over a finite field. Further descriptions on how these parameters are defined and generated can be found in [28]. 0 G 1 G

2.1.2 Attribute-Based Encryption

Attribute-based encryption (ABE) is proposed by Sahai and Waters [29]. It is derived from the Identity-Based Encryption (IBE) and is particularly suitable for one-tomany data sharing scenarios in a distributed and open cloud environment. In real applications, CP-ABE is more suitable since it resembles role-based access control. In CP-ABE, the data owner designs the access control policy and assigns attributes to data users. A user can decrypt the data properly if the user's attributes satisfy the access control policy.

2.1.3 Secret Sharing Scheme

Shamir secret sharing scheme is used to protect secret information.

2.2 Security Assumptions

2.2.1 Semi-trusted Server

LDSS is designed under the same assumptions proposed in 0 that the CSP is honest but curious, which means that the CSP will faithfully execute the operations requested by users, but it will peek on what users have stored in the cloud. The CSP will faithfully store users' data, undertake an initial access control, update data according to users' requests. However, CSP may do malicious actions such as collusion with users to get the data in plain text.

2.2.2 Trusted Authority

In this paper, to make LDSS feasible in practice, a trusted authority (TA) is introduced. It is responsible of generating public and private keys, and distributing attribute keys to users. With this mechanism, users can share and access data without being aware of the encryption and decryption operations.

2.2.3 Lazy Re-encryption

In ciphertext access control, data needs to be re-encrypted when some users' access privileges to the data are revoked. However, frequent re-encryption brings heavy computational overhead, and the accessed plaintext data may already be stored on these data users. Therefore, this paper adopts the lazy re-encryption method proposed in [3]. With lazy re-encryption, when a user's access privilege is revoked, data is not re-encrypted until the data owner updates the data.

3 OUR PROPOSED MECHANISM

In this section, we describe the LDSS system design. First, we give the overview of LDSS, and then we present LDSS-CP-ABE algorithm and system operations, which are the base of LDSS algorithm. Finally, we describe LDSS in details.

3.1 Overview

We propose LDSS, a framework of lightweight datasharing scheme in mobile cloud (see Fig. 1). It has the following six components.

(1) Data Owner (DO): DO uploads data to the mobile cloud and share it with friends. DO determines the access control policies.

(2) Data User (DU): DU retrieves data from the mobile cloud.

(3) Trust Authority (TA): TA is responsible for generating and distributing attribute keys.

(4) Encryption Service Provider (ESP): ESP provides data encryption operations for DO.

(5) Decryption Service Provider (DSP): DSP provides data decryption operations for DU.

(6) Cloud Service Provider (CSP): CSP stores the data for DO. It faithfully executes the operations requested by DO, while it may peek over data that DO has stored in the cloud.

LDSS-CP-ABE Algorithm

To better illustrate LDSS-CP-ABE algorithm, we first define the following terms.

Definition 1: Attribute An attribute defines the access privilege for a certain data file. Attributes are assigned to data users by data owners. A data user can have multiple attributes corresponding to multiple data files. A data owner can define a set of attributes for its data files. The data accesses are managed by access control policy specified by data owners. Let A = {A1, A2, A3, ..., An} be the set of attributes for a data owner. Each data user u also has a set of attributes Au, which is a non-empty subset of A, namely Au ⊆ {A1, A2, A3, ..., An}. For example, assume A is {relatives, colleagues, classmates, friends, teachers, peers, Hubei, Beijing, Shanghai, degree of intimacy}. Definition 2: Access Control Tree Access control tree is the specific expression of access control policies, in which the leaf nodes are attributes, and non-leaf nodes are relational operators such as and, or, n of m threshold. Each node in an access control tree represents a secret, and the secret of a top node can be split into multiple secrets by secret sharing scheme and distribute to lower level nodes

described in Definition 1.

Definition 3: Version Attribute. Version attribute is introduced in LDSS-CP-ABE algorithm to ensure security. It is an addition to the original access control tree, forming a new root node of and. We have the following definitions.

T: The new access tree with version attributes.

S: The secret related to the root of T. Ta, Ra, Sa: Ta is the initial access control tree and the left subtree of T. Ra is the root of Ta. Sa is the secret related to Ra. Tv, Rv, Sv: Tv is the right subtree of T and contains only one node, which represents the version attribute Rv. Sv is the secret related to Rv. Both Sa and Sv are derived from S based on the secretsharing scheme. For the example described in Definition 1, the access Encryption server provider（ESP） Decryption server provider（DSP).

Setup(A, V): Generate the master key MK, the public key PK based on attribute set A of the Data Owner and the version attribute V .

KeyGen(Au, MK): Generate attribute keys SKu for a data user U based on his attribute set Au and the master key MK.

Encryption(K, PK, T): Generate the ciphertext CT based on the symmetric key K, public key PK and access control tree T .

Decryption(*CT*,*T*,*SK*u): Decrypt the ciphertext *CT* using the access control tree *T* and the attribute keys *SK*u .
We explain all of these functions specifically below. First, function Setup() is called by the trusted third party (TA) to generate the master key and the public key. The master key is used to generate attribute keys and the public key is used to encrypt data files.

3.3 Attribute Description Field in LDSS-CP-ABE

Attribute description field is introduced in LDSS for dynamic user privilege management. It keeps access control strategy secret against the cloud.

3.4.1 System Initialization

In system initialization, Function 1 is executed. The specific process is described as follows.
(1) When the data owner (DO) registers on TA, TA runs the algorithm Setup() to generate a public key PK and a master key MK. PK is sent to DO while MK is kept on TA itself.
(2) DO defines its own attribute set and assigns attributes to its contacts. All these information will be sent to TA and the cloud.

3.4.2 File Sharing

The process of file sharing uses Function 3 to encrypt data files. The specific process is described as follows.
(1) DO selects a file *M* which is to be uploaded and encrypts it using a symmetric cryptographic mechanism (such as AES, 3DES algorithm) with a symmetric key *K*, generating ciphertext *C*.
(2) DO assigns access control policy for *M* and encrypts *K* with the assistance of ESP using Function 3, generating the ciphertext of *K* (*CT*).

3.4.3 User Authorization

The process of user authorization executes Function 2 to generate attribute keys for data users. The specific proc ess is described as follows.
(1) DU logins onto the system and sends, an authorization request to TA. The authorization request includes attribute keys (SK) which DU already has.
(2) TA accepts the authorization request and checks whether DU has logged on before. If the user hasn't logged on before, go to step (3) , otherwise go to step.

4. SECURITY ANALYSIS

The security assessment is based on the security assumptions we described in Section 3. The possible scenarios that malicious users may expose plaintext to others are not discussed.

4.1 Security Analysis of LDSS-CP-ABE

LDSS-CP-ABE algorithm is designed on top of Attribute- Based Encryption (ABE). The security of ABE is based on the bilinear diffie-hellman assumptions.

4.2 Data Confidentiality against Conspiracy

The data confidentiality is taken into account from two aspects. In LDSS, data are encrypted with a symmetric key. The security of this part is guaranteed by symmetric encryption mechanism. Next, the symmetric key is encrypted by attribute encryption. The security of this part depends on the encryption process.

5 PERFORMANCE EVALUATIONS

The cost of data sharing comes from the execution of the function Encryption(), which is executed every time when sharing data files. The function Encryption() includes exponentiation operation on G0 (the number of operations is proportional to the number of attributes included in the access strategy) and one exponentiation operation on G1.

	Bethencourt	BSW CP-ABS	
Data sharing	(2|*T*a|+1)*T*_ *G*0+*T*_*G*1	(4|*T*a|+1)*T*_ *G*0+*T*_*G*1	3T_G0+T_Gm
Data access	(2|*A*u|+1)T_*G*e	(2|*A*u|+1)*T*_*G*e	*T_G0+T_Gm*

Fig. 5.1 Computational overhead with different CP-ABES

5.3 Storage Overhead Evaluation

We also evaluate the storage overhead of LDSS and compare it with existing CP-ABE schemes.

5.3.1 Storage Overhead with Different CP-ABE Schemes

DO needs to keep PK, which is of the size (|*A*|+3)*L*G0+ *L*G1. DU also needs to keep SK, which is of the size (|*A*u|+4) *L*G0. TA needs to keep PK and MK. MK is of the size *L*G0. The cloud needs to keep the symmetric key ciphertext CT, which is of the size (2|*T*a|+3) *L*G0+ LG1. DSP / ESP only do calculations and need not retain any value.

5.4 Communication Overhead Evaluation

The communication overhead of access control happens when TA sends keys to DO/DU at the stage of system initialization and user authorization, and DO/DU encrypt/decrypt the symmetric key which is used to encrypt the data files.

6. RELATED WORKS

In this section, we focus on the works of ciphertext access control schemes which are closely related to our research. Access control is an important mechanism of data privacy protection to ensure that data can only be acquired by legitimate users. There has been substantial research on the issues of data access control in the cloud, mostly focusing on access control over ciphertext. Typically, the cloud is considered honest and curious. Sensitive data has to be encrypted before sending to the cloud. User authorization is achieved through key distribution.

7. CONCLUSION AND FUTURE WORK

In recent years, many studies on access control in cloud are based on attribute-based encryption algorithm (ABE). However, traditional ABE is not suitable for mobile cloud because it is computationally intensive and mobile devices only have limited resources. In this paper, we propose LDSS to address this issue. It introduces a novel LDSS-CP-ABE algorithm to migrate major computation overhead from mobile devices onto proxy servers, thus it can solve the secure data sharing problem in mobile cloud.

REFERENCES

[1]. Adam Skillen and Mohammad Mannan. On Implementing Deniable Storage Encryption for Mobile Devices. the 20th Annual Network and Distributed System Security Symposium (NDSS), Feb. 2013.

[2] Wang W, Li Z, Owens R, et al. Secure and efficient access to outsourced data. in: Proceedings of the 2009 ACM workshop on Cloud computing security. Chicago, USA: ACM pp. 55-66, 2009.

[3] Maheshwari U, Vingralek R, Shapiro W. How to build a trusted database system on untrusted storage. in: Proceedings of the 4th conference on Symposium on Operating System Design & Implementation-Volume 4. USENIX Association, pp. 10-12, 2000.

[4]Boneh D, Franklin M. Identity-based encryption from the Weil pairing. in: Proceedings of the Advances in Cryptology. Berlin, Heidelberg: Springer-Verlag, pp. 213–229, 2001.