International Journal of Research in Advent Technology, Special Issue, March 2019
E-ISSN: 2321-9637
International Conference on Technological Emerging Challenges (ICTEC-2019)
Available online at www.ijrat.org

A Frivolous Protected Data Allocation Pattern for Mobile Cloud Computing

V.Nagagopiraju, Dr.K.Ramchand H Rao,
[1]*Research Scholar in Acharya Nagarjuna University, Guntu, A.P*
[2] *Principal in ASN Women's Engg College, Tenali, A.P*
[1]*gopi.raju524@gmail.com,* [2]*ramkolasani@gmail.com*

Abstract— With the prevalence of distributed computing, cell phones can store/recover individual information from anyplace whenever. Subsequently, the information security issue in versatile cloud turns out to be increasingly extreme and counteracts further advancement of portable cloud. There are considerable investigations that have been led to enhance the cloud security. Be that as it may, the vast majority of them are not material for portable cloud since cell phones just have constrained registering assets and power. Arrangements with low computational overhead are in incredible requirement for versatile cloud applications. In this paper, we propose a lightweight information sharing plan (LDSS) for versatile distributed computing. It receives CP-ABE, an entrance control innovation utilized in ordinary cloud condition, however changes the structure of access control tree to make it reasonable for portable cloud situations. LDSS moves a vast part of the computational concentrated access control tree change in CP-ABE from cell phones to outer intermediary servers. Moreover, to diminish the client repudiation cost, it acquaints property depiction fields with execute sluggish disavowal, which is a prickly issue in program based CP-ABE frameworks. The test results demonstrate that LDSS can adequately diminish the overhead on the cell phone side when clients are sharing information in portable cloud situations.

Index Terms— mobile cloud computing, data encryption, access control, user revocation

1. INTRODUCTION

With 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. Since personal data files are sensitive, data owners are allowed to choose whether to make their data files public or can only be shared with specific data users. Clearly, data privacy of the personal sensitive data is a big concern for many data owners. The state-of-the-art privilege management/access control mechanisms provided by the CSP are either not sufficient or not very convenient. They cannot meet all the requirements of data owners. 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[15]. To simplify the privilege management, the data owner can divide data users into different groups and send password to the groups which they want to share the data. However, this approach requires fine-grained access control. In both cases, password management is a big issue.

Apparently, to solve the above problems, personal sensitive data should be encrypted before uploaded onto the cloud so that the data is secure against the CSP[16]. However, the data encryption brings new problems. How to provide efficient access control mechanism on cipher text decryption so that only the authorized users can access the plaintext data is challenging. In addition, system must offer data owner's effective user privilege management capability, so they can grant/revoke data access privileges easily on the data users[17]. There have been substantial researches on the issue of data access control over ciphertext. In these researches, they have the following common assumptions. First, the CSP is considered honest and curious. Second, all the sensitive data are encrypted before uploaded to the Cloud. Third, user authorization on certain data is achieved through encryption/decryption key distribution. In general, we can divide these approaches into four categories: simple ciphertext access control, hierarchical access control, access control based on fully homomorphic encryption [1][2] and access control based on attribute-based information encryption (ABE). All these proposals are designed for non-mobile cloud environment. They consume large amount of storage and computation resources, which are not available for mobile devices. According to the experimental results in [11], the basic ABE operations take much longer time on mobile devices than laptop or desktop computers. It is at least 27 times longer to execute on a smart phone than a personal computer (PC). This means that an encryption operation which takes one minute on a PC will take about half an hour to finish on a mobile device. Furthermore, current solutions don't solve the user privilege change problem very well. Such an operation could result in very high revocation cost. This is not applicable for mobile devices as well. Clearly, there is no proper solution which can effectively solve the secure data sharing problem in mobile cloud. As the mobile cloud becomes more and more popular, providing an efficient secure data sharing mechanism in mobile cloud is in urgent need.

To address this issue, in this paper, 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[18]. 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[19].

(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[20]. 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[21].

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

Define a function *e* as follows:

$e : G_0 \square G_0 \square G_1$

In this function, both G_0 and G_1 are multiplicative cyclic groups of the prime order *p*.

Assume that *g* is a generator of G_0 , *Zp* is a finite field.

Then *e* is a bilinear pairing if *e* has the following properties:

(1) Bilinear: : $\square u,v \square G_0$,$\square a$, $b \square Z_p$, $e(u^a , v^b) \square e(u,v)^{ab}$ ..

(2) Non-degeneracy: $e(g, g)$ is a member of G_1 if *g* is a member of G_0 .

(3) Computability: $\square u,v \square G_0$, $e(u,v)$ can be calculated.

In our implementation, we usually take G_0 as a group consisting points on an elliptic curve, G_1 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 [15].

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-to-many data sharing scenarios in a distributed and open cloud environment. Attribute-based encryption is divided into two categories: one is the Ciphertext-Policy Attribute Based Encryption (CP-ABE), in which the access control policy is embedded into ciphertext; the other one is Key-Policy Attribute Based Encryption (KP-ABE), in which the access control policy is embedded in the user's key attributes. 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.

2.2 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.

In LDSS, proxy encryption server and proxy decryption server are introduced to assist users to encrypt and decrypt data so that user-side overhead can be minimized. In essence, proxy servers are also machines in the cloud. Thus, we consider that they are honest but curious just as the CSP.

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.

We assume TA is entirely credible, and a trusted channel exists between the TA and every user. The fact that a trusted channel exists doesn't mean that the data can be shared through the trusted channel, for the data can be in a large amount. TA is only used to transfer keys (in a small amount) securely between users. In addition, it's requested that TA is online all the time because data users may access data at any time and need TA to update attribute keys.

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 data-sharing 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.

As shown in Fig. 1, a DO sends data to the cloud. Since the cloud is not credible, data has to be encrypted before it is uploaded. The DO defines access control policy in the form of access control tree

(refer to Definition 2 in Section 3.2) on data files to assign which attributes a DU should obtain if he wants to access a certain data file. In LDSS, data files are all encrypted with the symmetric encryption mechanism, and the symmetric key for data encryption is also encrypted using attribute based encryption (ABE). The access control policy is embedded in the ciphertext of the symmetric key. Only a DU who obtains attribute keys that satisfy the access control policy can decrypt the ciphertext and retrieve the symmetric key. As the encryption and decryption are both computationally intensive, they introduce heavy burden for mobile users. To relieve the overhead on the client side mobile devices, encryption service provider (ESP) and decryption service provider (DSP) are used. Both the encryption service provider and the decryption service provider are also semi-trusted. We modify the traditional CP-ABE algorithm and design an LDSS-CP-ABE algorithm to ensure the data privacy when outsourcing computational tasks to ESP and DSP.

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.

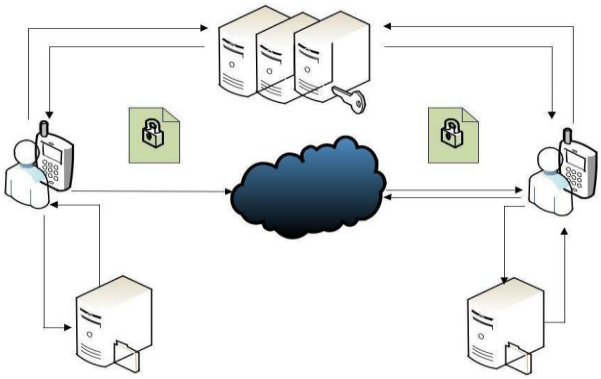

Fig. 1. A lightweight data-sharing scheme (LDSS) framework

3.2 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 $\{A1, A2, A3, ..., An\}$.

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

3.3 System Operations of LDSS

LDSS scheme is designed for data sharing in mobile cloud. The whole process of LDSS includes system initialization, file sharing, user authorization, and file access operations. It also has to support attribute revocation and file update operations.

3.3.1 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) TA and the cloud receive the information and store it.

3.3.2 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)DO uploads C, CT and access control policy to the cloud.

3.3.3 User Authorization

The process of user authorization executes Function 2 to generate attribute keys for data users. The specific process 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 hasnt logged on before, go to step (3) , otherwise go to step (4).

(3) TA calls Function 2 to generate attribute keys (SK) for DU.

(4) TA compares the attribute description field in the attribute key with the attribute description field stored in database. If they are not match, go to step (5), otherwise go to step (6).

(5) For each inconsistent bit in description field, if it is 1 on data users side and 0 on TA side, it indicates that

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.

Bilineardiffie-hellman assumptions: When attackers only have a, b, c, z Zp, there exists no polynomial algorithm that can get the relationship between

(A=ga, B=gb, C=gc, Z=e(g, g)ab/c) and (A=ga, B=gb, C=gc, Z=e(g, g)z). In other words, attackers cannot get Z=e(g, g)z that corresponds to e(g, g)ab/c.

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.

4.3 Confidentiality of Access Control Policy

The security of access control policy is that no participants could know the specific content of the access control policy except data owners. LDSS introduces attribute description field so that access control policy is described by the corresponding attribute description bit. ESP and the Cloud can only get the relationships between different attribute description bits, but not the specific content of access control strategy, thus protecting the access control strategy

5. PERFORMANCE EVALUATION

Table:1

COMPUTATIONAL OVERHEAD OF BASIC OPERATIONS OF ABE SCHEMES

Types of Devices	Pairing	Exponentiation	Multiplication
PC	20 ms	5 ms	0.7 ms
Mobile	550 ms	177 ms	26 ms

Table:

The cost of access control mechanisms is closely related to the size of access control policy. To reflect closely to the reality, in our experiment, the number of attributes owned by individual users is fixed, and the size of access control policy varies. We assume that the average number of attributes owned by DO is 10, and the number of attributes included in the access policies varies from 1 to 32.

Table:2
COMPUTATIONAL OVERHEAD OF DATA SHARING

	Exponentiation on G0	Exponentiation on G1	Paring on G0
ESP	2|Ta|	0	0
DO	3	1	0

TABLE 3
COMPUTATIONAL OVERHEAD OF DATA ACCESS

	Exponentiation on G0	Exponentiation on G1	Paring on G0
DSP	0	|Ta|	2|Ta|+1
DO	0	1	0

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. The cost of this function depends on which one does the encryption operation. Before introducing ESP, the cost is on DO. After the usage of ESP, the cost on DO is reduced to a constant value, and is no longer associated with the number of attributes in access control strategies. The overhead on ESP and DO is shown in Table 2.

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. The research can be generally divided into four areas: simple ciphertext access control, hierarchical access control, access control based on fully homomorphic encryption [1][2] and access control based on attribute-based encryption (ABE).

Simple ciphertext access control refers to that after data file encryption, the encryption keys are distributed in a secure way to achieve authorization for trusted users [3]. To reduce the overhead of massive user key distribution, Skillen and Mannan [4] designed a system called Mobiflage that enables PDE (plausibly deniable encryption) on mobile devices by hiding encrypted

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. The experimental results show that LDSS can ensure data privacy in mobile cloud and reduce the overhead on users side in mobile cloud. In the future work, we will design new approaches to ensure data integrity.

To further tap the potential of mobile cloud, we will also study how to do ciphertext retrieval over existing data sharing schemes.

REFERENCES

[1] Gentry C, Halevi S. Implementing gentry s fully-homomorphic encryption scheme. in: Advances in Cryptology EUROCRYPT 2011. Berlin, Heidelberg: Springer press, pp. 129-148, 2011.

[2] Brakerski Z, Vaikuntanathan V. Efficient fully homomorphic

encryption from (standard) LWE. in: Proceeding of IEEE Symposium on Foundations of Computer Science. California, USA: IEEE press, pp. 97-106, Oct. 2011.

[3] Qihua Wang, Hongxia Jin. "Data leakage mitigation for discertionary access control in collaboration clouds". the 16th ACM Symposium on Access Control Models and Technologies (SACMAT), pp.103-122, Jun. 2011.

[4] 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.

[5] 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.

[6] 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.

[7] Kan Yang, Xiaohua Jia, Kui Ren: Attribute-based fine-grained access control with efficient revocation in cloud storage systems. ASIACCS 2013, pp. 523-528, 2013.

[8] Crampton J, Martin K, Wild P. On key assignment for hierarchical access control. in: Computer Security Foundations Workshop. IEEE press, pp. 14-111, 2006.

[9] Shi E, Bethencourt J, Chan T H H, et al. Multi-dimensional range query over encrypted data. in: Proceedings of Symposium on Security and Privacy (SP), IEEE press, 2007. 350-364

[10] Cong Wang, Kui Ren, Shucheng Yu, and Karthik Mahendra Raje Urs. Achieving Usable and Privacy-assured Similarity Search over

Outsourced Cloud Data. IEEE INFOCOM 2012, Orlando, Florida, March 25-30, 2012

[11] Yu S., Wang C., Ren K., Lou W. Achieving Secure, Scalable, and

Fine-grained Data Access Control inCloud Computing.INFOCOM 2010, pp. 534-542, 2010

[12] Kan Yang, Xiaohua Jia, Kui Ren, Bo Zhang, Ruitao Xie: DAC-MACS: Effective Data Access Control for Multiauthority Cloud Storage Systems. IEEE
Transactions on Information Forensics and Security, Vol. 8, No. 11, pp.1790-1801, 2013.

[13] Stehlé D, Steinfeld R. Faster fully homomorphic encryption. in: Proceedings of 16th International Conference on the Theory and Application of Cryptology and Information Security. Singapore: Springer press, pp.377-394, 2010.

[14] Junzuo Lai, Robert H. Deng ,Yingjiu Li ,et al. Fully secure key-policy attribute-based encryption with constant-size ciphertexts and fast decryption. In: Proceedings of the 9th ACM symposium on Information, Computer and Communications Security (ASIACCS), pp. 239-248, Jun. 2014.

[15] Lakshman Narayana Vejendla and Bharathi C R ,(2018),"Multi-mode Routing Algorithm with Cryptographic Techniques and Reduction of Packet Drop using 2ACK scheme in MANETs",Smart Intelligent Computing and Applications, Vo1.1, pp.649-658.

[16] Lakshman Narayana Vejendla , A Peda Gopi and N.Ashok Kumar,(2018)," Different techniques for hiding the text information using text steganography techniques: A survey", Ingénierie des Systèmes d'Information, Vol.23, Issue.6,pp.115-125.

[17] A Peda Gopi, Lakshman Narayana Vejendla and N.Ashok Kumar,(2018)," Dynamic load balancing for client server assignment in distributed system using genetical gorithm", Ingénierie des Systèmes d'Information, Vol.23, Issue.6,pp.87-98.

[18] Lakshman Narayana Vejendla and A Peda Gopi, (2017)," Visual cryptography for gray scale images with enhanced security mechanisms", Traitement du Signal,Vol.35, No.3-4,pp.197-208.

[19] A Peda Gopi and Lakshman Narayana Vejendla, (2017)," Protected strength approach for image steganography", Traitement du Signal,Vol.35, No.3-4,pp.175-181.

[20] Gopi, A., et al. "Designing an Adversarial Model Against Reactive and Proactive Routing Protocols in MANETS: A Comparative Performance Study." International Journal of Electrical & Computer Engineering (2088-8708) 5.5 (2015).

[21] Kumar, S. Ashok, et al. "An Empirical Critique of On-Demand Routing Protocols against Rushing Attack in MANET." International Journal of Electrical and Computer Engineering5.5 (2015).