JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

An Enhanced Lightweight data access control scheme for Mobile Cloud Computing

Priyanka
Assistant Professor, Computer Engineering, Aravali college of Engineering and Management, Faridabad, India
Priyanka798257@gmail.com
Deepak Singh
Cloud Engineer
deepaksinghsarao@gmail.com

Abstract: *The rapid growth of mobile computing is seriously challenged by the resource constrained mobile devices. However, the growth of mobile computing can be enhanced by integrating mobile computing into cloud computing, and hence a new paradigm of computing called mobile cloud computing (MCC) emerges. Here, the data is stored in cloud infrastructure and the actual execution is shifted to cloud environment so that a mobile user is set free from resource constrained issue of existing mobile devices. Moreover, to avail the cloud services, the Communications between mobile devices and clouds are held through wireless medium. Thus, some new levels of security and privacy challenges are introduced. Mobile Cloud Computing increases the efficiency of the mobile devices. So, when data owner is sharing their confidential information on cloud then it is not in their hands and the control of that data is in the hands of some untrusted parties like service providers. Hence, to resolve this issue , we propose an Enhanced Lightweight data access control scheme , an (EL-CP-ABE) approach for improving the privacy and secrecy of data for mobile cloud computing. It is responsible to improve performance of the system by reducing the encryption and decryption activities overhead. The experimental results shows that proposed approach is secure, efficient and appropriate for lightweight mobile devices.*

Keywords – *Cloud Computing, Mobile Cloud Computing, Access Control*

1. Introduction:

Cloud Computing means storing data and accessing that data from the Internet for most of the operations. More than 50% of IT companies have moved their Business to the cloud. Sharing of data over the cloud is the new trend that is being set on. The amount of data generated on a day to day basis is increasing and to store that data in traditional hardware is not possible because of limited storage capacity. Therefore, transferring the data to the cloud is a necessity where user can get unlimited storage. Once the data is on cloud securing that data becomes of paramount importance. Also, once the data is uploaded to the cloud, an individual loses its control over that data. [1] 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. Therefore, privacy of the personal sensitive data is a big concern for many data owners. When any of the people upload the data onto the cloud they are leaving their data in a place where monitoring over that data is out of their control, the cloud service provider can also spy on the personal data of the users. When someone has to share data over the cloud, they have to share the password to each and every user for accessing the encrypted data which is cumbersome. Therefore, to solve this problem data should be encrypted before uploading it onto the cloud which can be safe from everyone. But with encryption comes its own challenges.

There have been substantial researches on the issue of data access control over cipher text. In these researches, they have the following common assumptions. First, the Cloud Service Provider (CSP) is considered honest. Second, all the sensitive data are encrypted before it is uploaded onto 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 cipher text access control, hierarchical access control, access control based on fully homomorphic encryption [1][2] and access control based on attribute-based encryption (ABE). An attribute defines the access privilege for a certain data file. Attributes are assigned to 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= $\{A_1, A_2, A_3 \dots A_n\}$ be the set of attributes for a data owner. Each data user *u* also has a set of attributesA_u, which is a non-empty subset of *A*, namely $\{A_1, A_2, A_3 \dots A_n\}$.

For example, assume *A* is {*relatives, colleagues, classmates, friends, teachers, peers, User, Ram, Sita, degree of intimacy*}. A data user's subset *A*u could be {*friend, User, degree of intimacy=3*}. The access control policy for a data file *M* could be: ((*friends and degree of intimacy > 1 and User) or* (relatives *and peers*)), which means a data user cannot access *M* unless these conditions are met.

ABE is a type of public key encryption in which the secret key of a user and the cipher text are dependent upon attributes. All these proposals are designed for non-mobile cloud environment. Large amount of storage and computation resources are needed, which are not available for mobile devices. According to the experimental results in [7], the basic ABE operations take much longer time on mobile devices than laptop or desktop computers. It is at least 27 [7] 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 (means the access control on a per user is not reduced i.e. the perfect balance between user productivity and security is to control user privileges) 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. The next big concern is time consumption for encryption. Traditional Hardware with big configuration can encrypt data in short amount of time but limited resource devices suffer from this problem. They require more amount of time for encryption and decryption. So, an efficient crypto system is to be proposed which can work equally on all of the devices.

To address this issue, in this paper, we propose an approach for providing privacy and security of data for mobile cloud computing environment.

The main contributions of proposed work are as follows:

1. We design a system which is based on Attribute-Based Encryption (ABE) method to offer efficient access control over cipher text for Mobile environment.
2. The proposed approach takes away the need of performing computational intensive operations from the local device. For this, use of proxy servers has been brought to use. Furthermore in order to maintain data privacy, an attributes are 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. Another salient feature of our approach is to introduce lazy re-encryption reduce the revocation overhead when dealing with the user revocation problem.

In this paper, we propose a secure Enhanced lightweight data access control scheme for MCC. As [32], [33] do, we also exploit Cipher text-Policy ABE (CP-ABE) [15] technique for data confidentiality and fine-grained data access control. However, unlike those schemes, our scheme is constructed on a secure and efficient CP-ABE algorithm, which is proposed by Ibraimi et al. [34] that is in turn inspired by [15]. In the following of this paper, we refer to our scheme as EL-CP-ABE. By greatly reducing the computation overheads in the process of encryption and decryption, EL-CP-ABE can achieve efficiency of the system compared with the previous related schemes. Meanwhile, most of the computation overheads at mobile device are outsourced to cloud servers. Furthermore, EL-CP-ABE provides flexible and expressive data access control policy. Consequently, mobile users will get better quality of service with smaller cost.

The remainder of this paper is organized as follows: Section II provides the survey of existing security frameworks for Mobile Cloud Computing (MCC) .Section III describes the basic technique and security assumption and also explains the overview of proposed work. Section IV explains the implementation and results. Finally future work and conclusion are identified in section V.

2. RELATED WORK

There are various methods of cipher text access control schemes which are closely related to our work. 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 cipher text. Typically, the cloud is considered honest. Confidential information 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 cipher text access control, hierarchical access control, access control based on fully homomorphic encryption [5][9] and access control based on attribute-based encryption (ABE).

1. Simple cipher text access control refers to the data file encryption, the encryption keys are distributed in a secure way to achieve authorization for trusted users [6]. To reduce the overhead of massive user key distribution, Skillen and Mannan [10] designed a system called Mobiflage that enables PDE (plausibly deniable encryption) on mobile devices by hiding encrypted volumes via random data on a device's external storage. However, the system needs to obtain large amount of information of keys. [11] borrows the access control method used in conventional distributed storage [10][12][18][20], separating users into different groups according to access rights and assign different keys to groups. This reduces the overhead of key management, but it cannot satisfy the demand for fine-grained access control.
2. Hierarchical access control has good performance in reducing the overhead of key distribution in cipher text access control [13]. As a result, there are substantial research on cipher text access control [20][21][22][23] based on hierarchical access control method. In hierarchical access control method, keys can be derived from private keys and a public token table. However, the operation on token table is complicated and generates high cost. Besides, the token table is stored in the cloud. Its privacy and security cannot be guaranteed [24].
3. Full homomorphic encryption algorithm can operate directly on the cipher text. Its operating results are the same with operating on plaintext and then encrypting the data. [25] By using full homomorphic encryption algorithm, operations such as

retrieval and calculation directly on cipher text. It can solve the problem that the cloud is untrustworthy fundamentally because all data update operations and user privilege change operations can be done directly on cipher text. However, this encryption scheme is too complex to implement in practical applications.

4. Attribute-based encryption algorithm is derived from identity-based encryption. It embeds decryption rules in the encryption algorithm, which avoids frequent key distribution. Lai et al [20] and Bethencourt et al [21] proposed key-policy attribute-based encryption (KP-ABE) and cipher text-policy attribute-based encryption (CP-ABE). In practical applications, CP-ABE has been extensively studied [22][23][24] since it is similar to role-based access control (RBAC) scheme [25]. In CP-ABE, the possession of one attribute key means that the key owner owns corresponding attribute, and attribute keys cannot be reclaimed once they are distributed. As a result, when a data user's attribute is revoked, how to ensure data privacy becomes a difficult issue [20]. Liang et al [22] propose attribute-based proxy re-encryption (ABPRE) scheme to solve this problem. However, in their solution, when a user's attribute is revoked, all other users who own this attribute will lose this attribute at the same time, which cannot satisfy fine-grained access control needs. Tian et al [26] combine CP-ABE and public key cryptography to achieve cipher text access control. However, it brings high cost to data owners. Di Vimercati et al [27] add a time stamp to attributes to limit the use of attribute keys to deal with attribute revocation problem. However, in this scenario, data users need to periodically apply for attribute keys and the users' attribute cannot be revoked before the time stamp expires. Yu et al [28] propose some work of revocation can be outsourced to CSP, whereas CSP should have certain credibility, and access control policy that contains "or" relationship or "threshold" relationship is not supported. Yu et al [29] also proposed a scheme to address the cloud computing challenging that keep sensitive user data confidential against untrusted servers by exploiting and uniquely combining techniques of attribute-based encryption (ABE), proxy re-encryption, and lazy re-encryption. Yang et al. [28] proposed a novel scheme that enabling efficient access control with dynamic policy updating for big data in the cloud that focusing on developing an outsourced policy updating method for ABE systems. It also designed policy updating algorithms for different types of access policies like Role based access control (RBAC) and it is defined as an approach to restricting system access to unauthorized users , Discretionary access control(DAC) and it is defined as a type of security access control that grants or restricts object access via an access policy determined by an object's owner group, Mandatory access control (MAC) and it defines and ensures a centralized enforcement of confidential security policy parameters.

All the above works focuses on the issue of data access control in the cloud. They are mainly for non-mobile devices and cannot be applied for data sharing in mobile cloud environment. With regards to data privacy in mobile cloud, some works have been done in this field [29]. Huang et al [30] propose MobiCloud, in which traditional Mobile Ad-hoc NETworks (MANETs) is transformed into service-oriented communication architecture. In this architecture, each mobile device is regarded as a service node, and the operations are outsourced to the cloud. However, in MobiCloud, users need to completely trust the cloud, which is not the case in reality. Livshits and Jung [31] designed and implemented a graph theoretic algorithm to place mediation prompts that protect every resource access, while avoiding repetitive prompting and prompting in background tasks or third-party libraries, for the problem of mediating resource accesses in mobile applications. Zhou et al [7] proposed an ABDS scheme to achieve secure data storage in the cloud. However, this scheme is not suitable for data sharing and has no clear solution for attribute revocation. Tysowski et al. [8] considered a specific cloud computing environment where data are accessed by resource-constrained mobile devices, and proposed novel modifications to ABE, which assigned the higher computational overhead of cryptographic operations to the cloud provider and lowered the total communication cost for the mobile user. Ibraimi et al. [34] proposed a more efficient CP-ABE algorithm, in which the secrets are assigned using the Unanimous Consent Control by Modular Addition (UCCMA) technique and do not need to be reconstructed by the polynomial interpolation.

A critical look at the literature survey highlights the following areas of gap w.r.t MCC.

1. Current proposals on data access control in the cloud are mostly for non-mobile terminals, which is not suitable for mobile devices.
2. Current solutions don't solve the problem of user privilege change scenarios very well since they bring high revocation cost. This is not applicable for mobile devices which only have limited computing capacity and power.
3. Existing studies on mobile cloud don't have a good solution to secure data sharing when servers are not credible.
4. There is no proper solution that can solve the problem of secure data sharing in mobile cloud.

So, proposed approach *EL-CP-ABE* scheme can effectively solve the problems mentioned above. The overall system performance is improved obviously, and it provides expressive and flexible data access control. *EL-CP-ABE* is secure, highly efficient and well suited for lightweight mobile devices.

Next section is an attempt to provide the various basic techniques which are a Pre – requisite for the proposed *EL-CP-ABE* approach.

3. Pre Requisites For the proposed approach

3.1 Attribute-Based Encryption

Attribute-Based Encryption (ABE) is proposed by Sahai and Waters [3]. It is derived from the Identity-Based Encryption (IBE) where *IBE* is a type of public key encryption in which the public key of a user is some unique information about the identity of the user for e.g. Email address. This 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 Cipher text-Policy Attribute Based Encryption (CP-ABE), in which the access control policy is embedded into cipher text; 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 *KP-ABE*, user's secret keys are generated based on the access tree that defines the privileges scope of the concerned user and data are encrypted over a set of attributes. So, it is not suitable in real environment. In real applications, CP-ABE is more suitable since it resembles *RBAC*. 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.

3.2 Lazy Re-encryption

In cipher text 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 the data users. Therefore, this paper adopts the lazy re-encryption method proposed in [6]. With lazy re-encryption, when a user's access privilege is revoked, data is not re-encrypted until the data owner updates the data.

In our approach, when the data owner revokes a user's privilege, the file of the access control policy that contains these attributes will be marked. Later, when the data owner updates this file, it first checks the mark to see if it has been marked as revoked. If that is the case, this file will be re-encrypted.

3.2 Security Assumption

Security threats like data breaches are growing rapidly and very common now a days. While almost everyone has knowledge of data breaches but it is harder to deal with them. So, given below are the common security assumptions to avoid security threats.

3.2.1 Semi-trusted Server

Proposed scheme is designed under the same assumptions proposed in [5] that the Cloud Service Provider (CSP) is honest but curious, which means that the CSP will faithfully execute the operations requested by users, but it will peak on to 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 combine with users to get the data in plain text.

In proposed work, 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.

3.2.2 Trusted Authority

In this paper, to make the system feasible, a trusted authority (TA) is introduced. It is responsible for generating public and private keys, and distributing those attributes 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.

The next section is an attempt to discuss the detailed working or methodology related to our proposed work. It gives a novel approach for providing secrecy and privacy of data when the data owner put his/her confidential data on cloud service provider.

4. PROPOSED MECHANISM

As highlighted in the literature review, all the major work is done on cloud computing, leaving the serious gaps pertaining to security and authentication in MCC. The reason is that existing approaches of cloud computing fail pertaining to resource crunch on MCC. This paper proposes a novel approach to tackle the above security issue on MCC.

4.1 Overview of proposed work

We propose a framework for up the information privacy and secrecy in mobile cloud (see Fig. 1). during this projected work, security is necessary parameter and additionally the service provider build positive that there is no unauthorized access to the sensitive info. To tackle the authentication drawback idea of trust authority is taken. We are dividing the data into splits and by using the AES algorithm the file is encrypted with the help of keys which are generated by the trusted authority and then these splits are encrypted and are stored over the distributed cloud to ensure security .

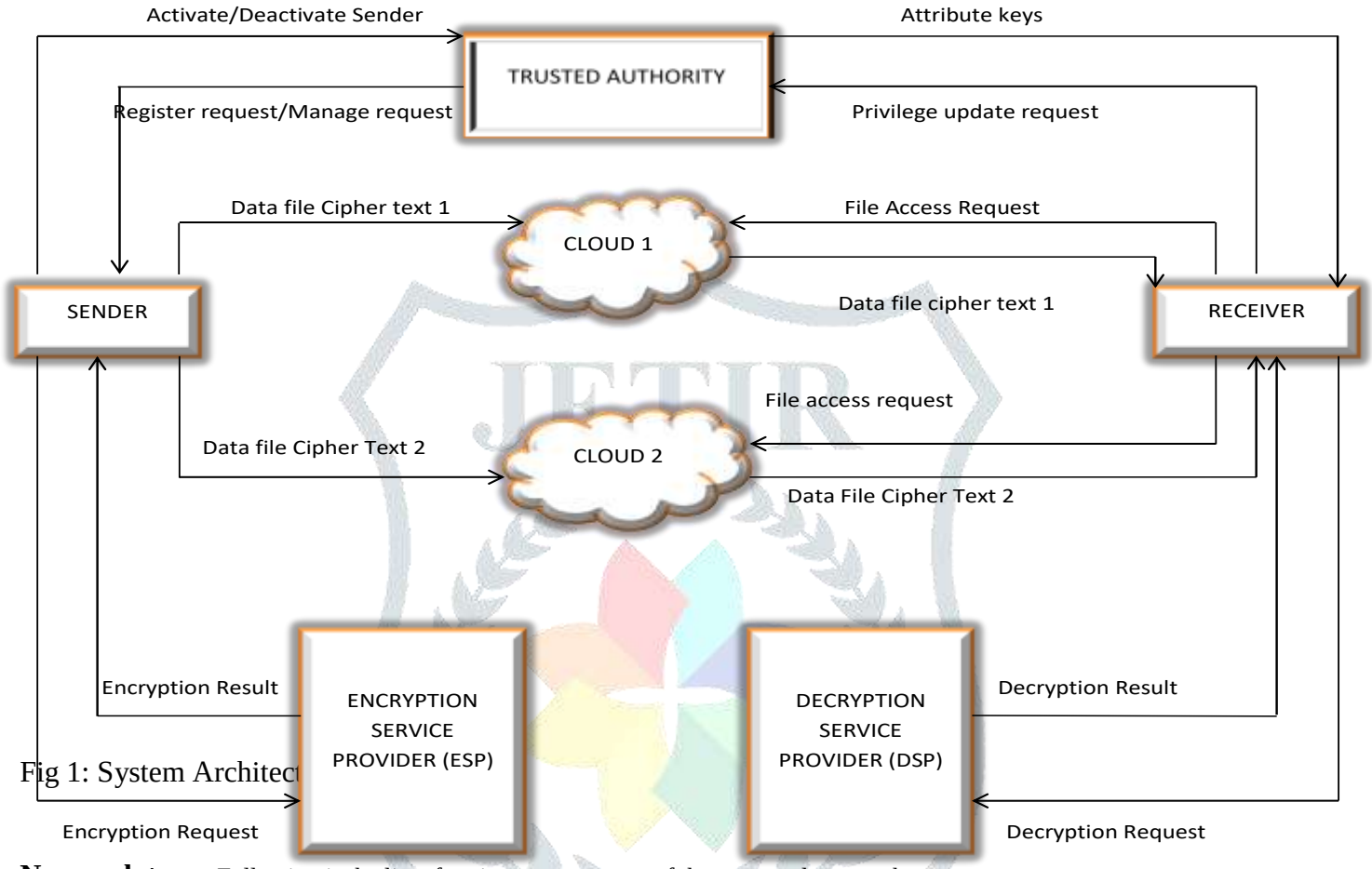

Fig 1: System Architec

Nomenclature: Following is the list of various components of the proposed approach.

- **Sender/Data owner:** Sender allocations information to the portable/mobile cloud and shares it with friends. Sender is also data owner, decides the access control policies.
- **Receiver/Data user:** It recovers information from the mobile cloud.
- **Trust Authority (TA):** TA is in charge of creating and handling the attribute keys.
- **Encryption Service Provider (ESP):** It provides data encryption activities for sender/DO.
- **Decryption Service Provider (DSP):** It provides data decryption activities for receiver/DU.
- **Cloud Service Provider (CSP):** It stores the information for Sender i.e. Data Owner (DO). It loyally executes the task mentioned by DO, while it might look over information that DO have stored in the cloud.

As appeared in Fig. 1, a Sender sends data to the 2 distinct clouds. Since the cloud is not secure, data should be encoded before it's transferred onto the clouds. The DO defines access management policy on files to assign that attributes a DU ought to get if he needs to access a precise knowledge file. In planned work, splitted data files are all encrypted with the symmetric coding mechanism, and therefore the symmetric key for {data coding|encoding|encryption} is additionally encrypted victimisation attribute based mostly encryption (ABE). The access management policy is embedded within the cipher text of the symmetric key. Only a Receiver i.e. Data User (DU) who obtains attribute keys that satisfy the access management policy will decipher the cipher text from the various clouds and retrieve the symmetric keys. because the encryption and decryption square measure each computationally intensive, they introduce significant burden for mobile users. To scale back the overhead on the mobile devices, Encryption Service supplier (ESP) and Decryption Service supplier (DSP) square measures are used. Each of the ESP and DSP are also semi-trusted. we tend to modify the normal CP-ABE algorithmic program and style a system to confirm the info privacy once outsourcing machine tasks to ESP and DSP.

4.2 Framework Activities:

This scheme is designed for secure data sharing in mobile cloud. The whole process includes system initialization, file sharing, user authorization, and file access operations. It also has to support attribute revocation and file update operations. Fig 2 shows the flow diagram of the proposed framework.

4.2.1 System Initialization

In system initialization the process is described as follows.

(1) When the Sender registers on TA, TA generate a public key (PK) and a master key (MK). PK is sent to the Sender while MK is kept on TA itself.

(2) Sender defines its own attribute set and assigns attributes to its contacts. All these information will be sent to TA.

(3) TA receives the information and stores it.

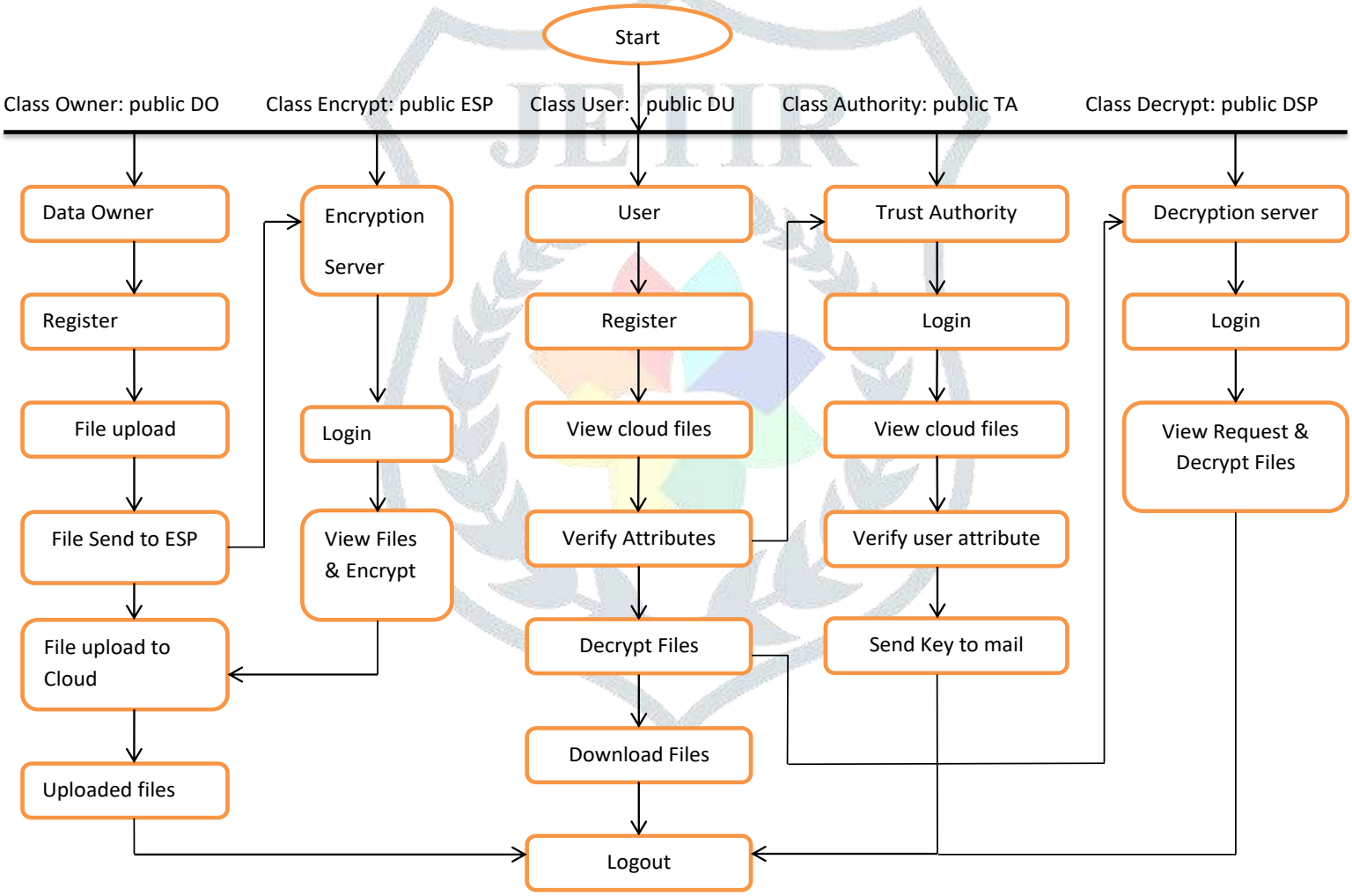

Fig 2: Flow diagram

4.2.2 File Sharing

The process of file sharing uses Function to encrypt data files. The specific process is described as follows.

(1) Sender selects a file *M* splits it into chunks which is to be uploaded on different clouds and encrypts it using a symmetric cryptographic mechanism i.e. AES algorithm with a symmetric key *K*, generating cipher text *C*.
(2) Sender assigns access control policy for *M* and encrypts *K* with the assistance of ESP generating the cipher text
(3) Sender uploads C to the cloud.

4.2.3 User Authorization

The process of user authorization is used to generate attribute keys for data user i.e. Receiver. 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 DU hasn't logged on before, go to step (3), otherwise go to step (4).
(3) TA generates 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 user's side and 0 on TA's side, it indicates that DU attribute has been revoked, and then TA does nothing on this bit. If it is reversed scenario, it indicates that DU has been assigned with a new attribute, and then TA generates the corresponding attribute key for DU.
(6) TA checks the version of every attribute key of DU. If it's not the same with the current version, then TA updates the corresponding attribute key for DU.

In the stage of user authorization, TA updates attribute keys for Receiver according to the attribute description field, which is stored with *SK*. It describes which attributes Receiver has and their corresponding versions. TA also keeps attribute description field of Receiver in database. When Sender changes the attribute of Receiver, the attribute description field on the TA side is also updated. Thus, when Receiver logins on the system, the attribute description field on itself may be different from that of TA. TA has to update the attribute keys for Receiver according to the attribute description field just as described above.

4.2.4 Access Files

When DU requests to access a certain data file, Function is used to decrypt data. The specific process is described as follows:

(1) DU sends a request for data to the cloud.
(2) Cloud receives the request and checks if the DU meets the access requirement. If DU can't meet the requirement, it refuses the request; otherwise it sends the cipher text to DU.
(3) DU receives the cipher text, which includes cipher text of data files and cipher text of the symmetric key. Then Receiver decrypts the cipher text of the symmetric key with the assistance of DSP.
(4) Receiver uses the symmetric key, Cloud A key and Cloud B key to decrypt the cipher text of data files.

4.2.5 Privilege Revoked

Data Owner can revoke attributes from a Data User. The process is as follows.

(1) DO inform TA and the cloud that one attribute has been revoked from a specific DU.
(2) TA and the cloud update the information of DU in database.
(3) DO mark the corresponding bit of the attribute description field of data files.

This strategy implements the asynchronous processing of attribute revocation and attribute keys update operations. When DO want to revoke one attribute from a DU, TA only updates the database and doesn't update attribute keys for DU simultaneously.

4.3 Working of EL-CP-ABE

Overview

In EL-CP-ABE, *TA* first performs the setup work and publishes *PK* to every data owner. Then Receiver gets Symmetric Key *(SK)* associated with his attributes from TA. In the encryption phrase, Owner of the data first split the file into two different chunks and performs a few computation operations on message m to get an intermediate cipher text. DO then send the intermediate cipher text and some no-critical parameters to Encryption Service Provider (ESP). ESP performs the computation intensive exponentiation operations generates the other intermediate cipher text. ES finally combines two parts together to get a complete cipher text and uploads it to the Cloud Service Provider (CSP). While in the decryption phrase, Receiver invokes Decryption Service Provider to generate an intermediate parameter used for decrypting m, and the computation intensive pairing operations are outsourced to DSP. Receiver then performs the final step to get the real message i.e. plain text.

After completing the above operations, most of the computation overheads have been securely outsourced to proxy servers located in cloud. Thus mobile users only need to perform a small number of computation operations locally. Lightweight mobile devices can easily complete the data processing operations with the help of proxy servers without disclosing data contents and valid user secret keys, and meanwhile achieve fine-grained data access control efficiently.

In brief, Contrasted with ancient PCs, cell phones are much weaker in processing capacity. It is hard for lightweight cell phones to complete the encryption and decryption tasks of CP-ABE all alone, which will significantly upset clients' quality of experience. To reduce the calculation overheads at versatile/mobile client, we take consideration of securely outsourcing computation intensive CP-ABE operations to the Encryption Server (ES) and Decryption Server (DS) situated in cloud. So, *EL-CP-ABE* on Ibraimi et al's CP-ABE calculation is efficient since it is secure and exceptionally effective.

4.3.1 The Construction of EL-CP-ABE

EL-CP-ABE consists of four parts: system setup, key generation, encryption and decryption and also how the data file is splitted. The detailed construction of SL-CP-ABE is as follows:

Setup (*K*): Generate the master key *MK*, the public key *PK* based on the security parameters K.

KeyGen (*MK,PK, User attributes*): Generate attribute keys *SK* associate with user attributes for a receiver D*U* based on his attribute set and the master key *MK*.

Encryption (*m, PK, and SK*): Generate the cipher text *CT* from the plain text *m* based on the symmetric key, public key *PK*.

Decryption (*CT, SK, and PK*): Decrypt the cipher text *CT* using the Secret key and decrypts the message m from CT.

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.

1. System Setup

TA runs the Setup algorithm, as shown in Algorithm 1, to generate a system public key PK and a system master key MK according to the attribute universe.

Algorithm 1. Setup ( )
INPUT: Security parameter *k* based on sender's attribute.
OUTPUT: Public Key PK, Master Key MK.
1. Sender/DO sends the request to TA.
2. TA checks the Sender and based on the attributes it will activate /deactivate the sender of the data.
3. If TA activates the sender then the sender can share its data on the cloud.
4 TA generate the two keys one is Public Key (PK) for sender and the other is Master Key (MK) which is kept by TA itself.
5. If TA deactivates the sender of the data then the sender is not able to upload the data on cloud.

Every party can get access to PK while MK is secretly stored by TA and not published to the public.

2. Key Generation

TA runs the Keygen algorithm, as shown in Algorithm 2, to generate a user secret key SK associated with an attribute set, according to MK.

Algorithm 2. Keygen ( )

INPUT: System master key *MK*, attribute set describing the identity of the user.
OUTPUT: Secret key *SK* associated with user Random Key for cloud A and Cloud B.
1. KeyGen.init (128);
2. Compute Secret Key = keygen.generateKey ( )
3. Encode secret key
Byte [ ] b = secretkey.Encoded ( )
4. Converting secret key to String

String skey = Base 64. Encode (b)
5. Compute Cloud A key
Random r = new Random ( );
Int i = r. nextInt (10000 – 5000) +5000
String public key
6. Compute Cloud B key
Random r1 = new Random ( );
Int i1 = r1. NextInt (10000 – 5000) +5000
String Private Key.

3. Encryption

The encryption algorithm , as shown in Algorithm 3 , to convert the plain text into cipher text by using Advanced Encryption Standard (AES) encryption Technique.

Algorithm 3. Encryption ( )

INPUT: Message *m*, system public key *PK, Secret Key SK*.
OUTPUT: Intermediate cipher text CT.

1. Getting AES Instance
AesCipher = Cipher.getInstance ("AES")

2. Initiating Cipher encryption using secret key
aesCipher.init (Cipher.ENCRYPT_MODE, secret key)
3. Converting plain text to byte
ByteDataToEncrypt = PlainData.getByte ( )
4. Encrypting String data one by one
ByteCipherText = aesCipher.doFinal (ByteDataToEncrypt)
5. Converting encrypted data to string
Cipher Text = new Base64Encoder ( ).encode (byteCipherText)
6. Return *CT*.

4. Decryption

The decryption phrase is computationally expensive as secret key takes heavy overheads. To reduce the computation overheads at mobile client, we outsource these pairing operations to DPS. DU first requires data from CSP. Then CSP checks whether the DU's attributes set satisfies or not. If so, CSP sends CT to DSP. Next, DSP runs the Decryption algorithm, as shown in Algorithm 5, to generate the plain text of the file and sends it to DU.

Algorithm4. Decryption A ( )

INPUT: Attribute set of user, Secret Key (SK), Cipher Text.
OUTPUT: Message m.
1. Initiating plain text decryption using secret key
aesPlain.init (Cipher.DECRYPT_MODE, secret key)
2. Converting Cipher text to byte
ByteDataToDecrypt = CipherData.getByte ( )
3. Encrypting String data one by one
BytePlainText = aesPlain.doFinal (ByteDataToDecrypt)
4. Converting decrypted data to string
Plain Text = new Base64Decoder ( ).decode (byte Plaintext)
5. Return the message *m*.

5. Splitting of data

Input: Data File
Output: File is splitted into chunks

```
int fileSize = (int) inputFile.length();
        int nChunks = 0, read = 0, readLength = fileSize/2;
        byte[] byteChunkPart;
          inputStream = new FileInputStream(inputFile); //taking the complete data file
              while (fileSize > 0) {
                    if (fileSize <= 2) {
                          readLength = fileSize;
                    end if
                    byteChunkPart = new byte[readLength];  //dividing into chunks
                    read = inputStream.read(byteChunkPart, 0, readLength);
                    fileSize -= read;
                    assert (read == byteChunkPart.length);
                    nChunks++
                end while
```

For Cloud A

```
if (nChunks==1)       // if chunk part value is1

  int i = r.nextInt(10000 - 5000) + 5000;  //key value in integer form
  String dkey = i+""
End if
```

For Cloud B

```
else if(nChunks==2)        // if chunk part value is 2

  int i = r.nextInt(10000 - 4000) + 5000;    //key value
  String dkey = i+""
End if
```

. SECURITY ANALYSIS AND PERFORMANCE EVALUATION

5.1. Security Analysis

We now discuss the security properties of EL-CP-ABE, including data confidentiality, fine-grained data access control, user access privilege confidentiality and collusion resistant.

5.1.1 Data confidentiality: In the encryption algorithm, ESP is bound to deal with the computations that related to access policy. And the data owner of the file only knows the basic attributes so data is confidential. Thus ESP cannot get the data content of the cipher text. While in the decryption algorithm, sending all the secret key components will also not reduce security level. DSP can use these secrets to get an intermediate result, but the final setup of decryption is performed by mobile client. So DSP also cannot get the data content of the cipher text. Furthermore, since our scheme is based on Ibraimi et al CP-ABE algorithm which is secure. So EL-CP-ABE is as secure as their scheme.

5.1.2 Fine-grained data access control: In EL-CP-ABE, since there are no special restrictions on the access tree and it meets any access policy, the data owner is able to define and enforce flexible and expressive data access control policy for each outsourced data. Specifically, the access control policy of each data is defined as an access tree over data owner's attributes and will be embedded into the cipher text in encryption phrase. A user will only be able to decrypt the cipher text if his attributes pass through the access tree.

5.1.3 User access privilege confidentiality: The cloud servers can get the partial attributes of a receiver in encryption and decryption phrases, but the complete set of attribute is still hidden. They only get to know that the attributes of the user's can only pass through a process performed by TA . Consequently, it is difficult for cloud servers to get the identity and the access privilege of a user.

5.1.4 Collusion resistant: In EL-CP-ABE, the Keygen algorithm generates a different random value for each user, which is embedded in each secret key component of the user. This means that each user's secret key is randomly created and cannot be match with the others. And three different keys are generated i.e. cloud A key , Cloud B key and a Secret Key So attacker cannot collude to expand their access privileges in EL-CP-ABE, including the cloud servers.

5.2 Performance Evaluation

This section provides the performance evaluation of the proposed framework. Fig 6 explains the encryption analysis given by EL-CP-ABE and CP-ABE algorithm. Fig 7 defines the decryption analysis provided by EL-CP-ABE and CP-ABE. And fig. 8 briefly shows that the encryption and decryption algorithm i.e. AES encryption provided by an EL-CP-ABE algorithm is more secure than the other different algorithms.

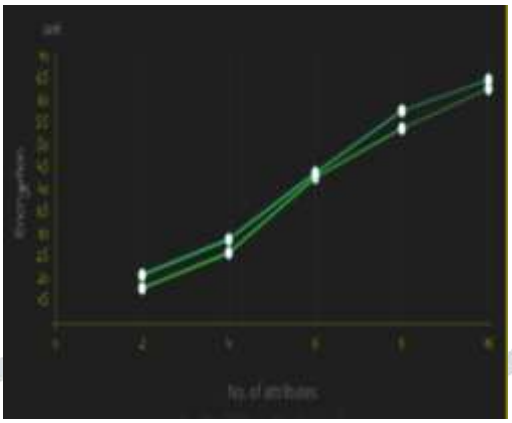

Fig.6 Encryption analysis between EL-CP-ABE and CP-ABE

The above figure shows the comparison of encryption algorithm between CP-ABE and EL-CP-ABE .From the above graph we can easily conclude that encryption algorithm used in EL-CP-ABE is more secure than the encryption algorithm used in CP-ABE.

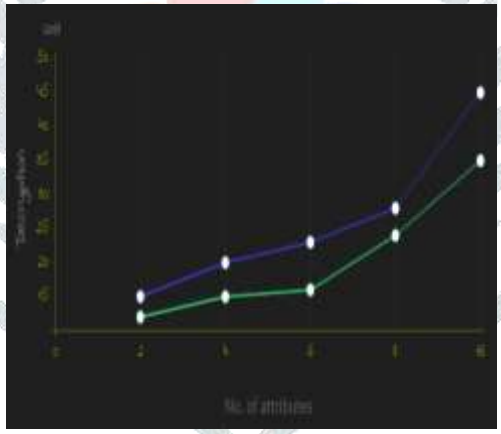

Fig.7 Decryption analysis between EL-CP-ABE and CP-ABE

The above figure shows the comparison of decryption algorithm between CP-ABE and EL-CP-ABE .From the above graph we can easily conclude that decryption algorithm used in EL-CP-ABE is also more secure than the decryption algorithm used in CP-ABE.

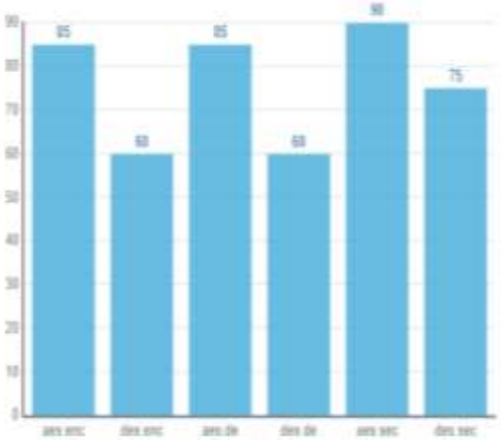

Fig.8 Secure Encryption and Decryption analysis of EL-CP-ABE

The above figure shows the comparison of encryption algorithm and decryption algorithm used in our proposed work and in the existing work. In our proposed work we use AES Encryption and Decryption techniques for scrambling and unscrambling the information. From the above graph we can conclude that AES Techniques are more efficient and more secure than the DES Technique.

At last we conclude from the given result that EL-CP-ABE is more secure and efficient than CP-ABE for mobile cloud computing.

5. CONCLUSION AND FUTURE WORK

In this proposed framework we first talked about the various limitations of securely storing and transferring data in MCC. To resolve this problem, we proposed an Enhanced lightweight data access control scheme named EL-CP-ABE, which can provide the confidentiality, authenticity, integrity of data and meanwhile it attains fine-grained data access control effectively in MCC. Compared with previous works, EL-CP-ABE has better results as it increases the security level. Firstly, the overall system performance is improved obviously by splitted the data files into two and also it greatly reducing the computation overheads in encryption and decryption phrases. Secondly, there is not even any special restriction on access policy, which gives the meaningful and variable data access control. Ultimately, cell phones can easily finish their data processing activities by sending most of the computation activities to proxy servers located in cloud. For future work, we will work onto improve the security in the mobile environment and make it more efficient

References:

[1] "A Lightweight Secure Data Sharing Scheme for Mobile Cloud Computing "Ruixuan Li, Member,IEEE Chenglin Shen, Heng He, Zhiyong Xu, and Cheng-Zhong Xu, Member, IEEE

[2] 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.

[3] Sahai A, Waters B. Fuzzy identity based encryption. in: Proceedings of the Advances in Cryptology. Aarhus, Denmark: Springer-Verlag, pp.457-473.

[4] Shamir A. How to share a secret. Communications of the ACM,1979, 22 (11): 612-613

[5] 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.

[6] 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.

[7] Zhou Z, Huang D. Efficient and secure data storage operations for mobile cloud computing. in: Proceedings of 8th International Conference on Network and Service Management (CNSM 2012), Las Vegas, USA: IEEE, pp. 37-45, 2012.

[8] P. K. Tysowski and M. A.Hasan. Hybrid attribute- and re-encryption-based key management for secure and scalable mobile applications in clouds. IEEE Transactions on Cloud Computing, vol. 1, no. 2, pp. 172-186, Nov. 2013.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

[15] 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

[16] 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

[17] Yu S., Wang C., Ren K., Lou W. Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing. INFOCOM 2010, pp. 534-542, 2010

[18] 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.

[19] 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.

[20] 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.

[21] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute- based encryption. in: Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP). Washington, USA: IEEE Computer Society, pp. 321-334, 2007.

[22] Liang Xiaohui, Cao Zhenfu, Lin Huang, et al. Attribute based proxy re-encryption with delegating capabilities. in: Proceedings of the 4th International Symposium on Information, Computer and Communications Security. New York, NY, USA: ACM press, pp. 276-286, 2009.

[23] Pirretti M, Traynor P, McDaniel P, et al. Secure atrribute-based systems. in: Proceedings of the 13th ACM Conference on Computer and Communications Security. New York, USA: ACM press, pp. 99-112, 2006.

[24] Yu S., Wang C., Ren K., et al. Attribute based data sharing with attribute revocation. in: Proceedings of the 5th International Symposium on Information, Computer and Communications Security (ASIACCS), New York, USA: ACM press pp. 261-270, 2010.

[25] Sandhu R S, Coyne E J, Feinstein H L, et al. Role-based access control models. Computer, 29(2): 38-47, 1996.

[26] Tian X X, Wang X L, Zhou A Y. DSP RE-Encryption: A flexible mechanism for access control enforcement management in DaaS. in: Proceedings of IEEE International Conference on Cloud Computing. IEEE press, pp.25-32, 2009

[27] Di Vimercati S D C, Foresti S, Jajodia S, et al. Over-encryption: management of access control evolution on outsourced data. in: Proceedings of the 33rd international conference on Very large data bases. Vienna, Austria: ACM, pp. 123-134, 2007.

[28] Kan Yang, Xiaohua Jia, Kui Ren, Ruitao Xie, Liusheng Huang: Enabling efficient access control with dynamic policy updating for big data in the cloud. INFOCOM 2014, pp.2013-2021, 2014.

[29] Jia W, Zhu H, Cao Z, et al. SDSM: a secure data service mechanism in mobile cloud computing. in: Proceedings of 30th IEEE International Conference on Computer Communications. Shanghai, China: IEEE, pp. 1060-1065, 2011.

[30] D. Huang, X. Zhang, M. Kang, and J. Luo. Mobicloud: A secure mobile cloud framework for pervasive mobile computing and communication. in: Proceedings of 5th IEEE International Symposium on Service-Oriented System Engineering. Nanjing, China: IEEE, pp. 90-98, 2010.

[31] Benjamin Livshits, Jaeyeon Jung. Automatic Mediation of Privacy-Sensitive Resource Access in Smartphone Applications. USENIX Security, pp.113-130, Aug. 2013.

[32] S. Yu, C. Wang, K. Ren, and W. Lou, “Attribute based data sharing with attribute revocation,” Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS ’10), 2010.

[33] J. Hur and D. K. Noh, “Attribute-based access control with efficient revocation in data outsourcing systems,” IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 7, pp. 1214-1221, 2011.

[34] L. Ibraimi, Q. Tang, P. Hartel and W. Jonker, “Efficient and provable secure ciphertext-policy attribute-based encryption schemes,” Lecture Notes in Computer Science, Berlin Heidelberg: Springer, pp.1-12, 2009.

[35] L. Cheung and C. Newport, “Provably secure ciphertext policy ABE,” Proc. 14th ACM Conf. Computer and communications security, 2007, pp. 456-465.