International Journal of Engineering Research in Computer Science and Engineering Engineering (IJERCSE) Vol 2, Issue 3, March 2015

SECURE AND REVOCABLE DATA ACCESS CONTROL FOR MULTI-AUTHORITY CLOUD STORAGE

R.Devendran[1] , K.Ganeshnathan[2] , D.Yukeshkumar[3] , Mrs.K.Amsavalli[4]
[1] [2] [3]B.E, IV year Department of Computer Science and Engineering
[4]Assistant professor of Computer Science and Engineering
Anand Institute of Higher Technology ,Chennai

Abstract — **Data access control is an effective way to ensure the data security in the cloud. Due to un trusted cloud servers and data outsourcing, the data access control becomes a challenging issue in cloud storage system. Cipher text-Policy Attribute based Encryption is regarded as one of the most suitable technologies for data access control in cloud storage, because it gives data owners to more direct control on access policies. However, it is difficult to directly apply the existing CP-ABE schemes to data access control in cloud storage system, because of the attribute revocation problem. In this paper, we design secure and revocable data access control for multi-authority cloud storage system and also it will efficient and effective cloud storage system. Were there are multiple authorities co-exist and each authority is able to issue attribute independently. Specifically, we propose most suitable encryption is called hybrid encryption method and it has two types of different encryption algorithm one is the most suitable of AES algorithm and another one is the jasypt algorithm, It will give more security in the cloud storage system and a revocable multi- authority CP-ABE scheme, and applying it as the underlying technique to design the data access control scheme. Our attribute revocation method and hybrid encryption method can efficiently achieve both forward and backward security and secure storage system. The analysis and simulation results show that our proposed data access control scheme is secure in the random oracle model and it is more secure and efficient than previous work.**

I. INTRODUCTION

The field of cloud computing is still in its infancy as far as implementation and usage, partly because it is heavily promoted by technology advancement. Cloud computing is not an innovation per us, but a means to constructing IT services that use advanced computational power and improved storage capabilities. And our paper is introduce the current state of cloud computing, with its development challenges. And it mainly describes the cloud computing security. Cloud storage is an important service of cloud computing, Which offers services for data owners to host their data in the cloud. The new paradigm of data hosting and data access services introduce a great challenges to data access control. Because the cloud server cannot be fully trusted by data owners, they can no longer rely on servers to data access control. Cipher-text Policy Attribute Based Encryption is regarded as one of the most suitable technologies for data access control in cloud storage system. Because it gives the data owner more direct control in access policies. In our paper we mainly introduce most suitable technology that is called hybrid encryption method and revocation CP-ABE scheme, there is an authority that is responsible for attribute management and key distribution. The authority can be the registration office in a university, the human resource department in a company, etc. The data owner defines the access policies and encrypts the data according to the policies. Each user will be issued a secret key reflecting its attributes. A user can decrypt the data only when its attributes satisfy the access policies. Cipher text policy attribute based encryption is only able to decrypt the message if the attributes in the policy match with the attribute in secret key, which provides a new flexible and efficient mechanism for realizing one-to-many encryption, and due to its flexible expressiveness, it is regarded as a promising tool for enforcing fine grained access control over encrypted data.

Commercial interests is one of the root causes for the user to apply cloud computing, but when general cloud storage services are unable to meet user's security needs, they will turn to choose relatively expensive but more secure encryption cloud storage service. In CP-ABE, size of cipher text and secret key will increase linearly with the number of attributes in policy, it will increase the transmission and user's cost , inevitably, there will be some system attributes revocation and user permission change operation. And in CP-ABE scheme have two types one is single-authority CP-ABE it will all attribute managed by single authority and another one is multi-authority CP-ABE will attribute are from different domains and managed by different authorities. In recently years, researchers have proposed a series of attribute encryption schemes. One of the efficient construction of the CP-ABE with (t, n) can be found in the [1,2]; the size of cipher text in [1] is n + O(1) and in [2] is 2(n – t)+ O(1). In [3], authors proposed a scheme in which cipher text remains constant in length, irrespective of the number of attributes, but not support attributes revocation.

In this paper to overcome the lack of the literature , we introduce the idea of hybrid encryption method and attribute revocation CP-ABE proposed in our scheme to support revocation problem. And our attribute revocation method is efficient in the sense that it incurs less communication cost and computation cost and it can achieve both forward security and backward security. In the forward security the newly joined user can also decrypt the previously published cipher text, it has the sufficient attributes. In backward security the revoked user cannot decrypt any new cipher text that requires the revoked attribute to decrypt. In hybrid encryption is a mode of encryption that merges two or more encryption system so we incorporates the Advanced Encryption System (AES) and japsyt encryption method. So these strength are respectively defined as

speed and security and hybrid encryption is considered a highly secure type of encryption as long as the public and private keys are fully secure. On the basis of combination of CP-ABE technology and hybrid encryption technology control the length of the cipher text at the same time achieve high revocation of the system attributes. Cloud storage service provider in the scheme is only responsible for the storage and hybrid encryption of the cipher text, so do not worry about its own security problem of cloud storage service providers.

And used our scheme we have many improvements, that are we modify the framework of the scheme and make it more practical to cloud storage system, in which data owners are not involved in the key generation. Specifically, a users secret key is not related to the owners key, such that each user only needs to hold one secret key from each authority instead of the multiple secret keys associated to multiple owners. And greatly improve the efficiency of the attribute revocation method. Specifically , our new attribute revocation method, only the cipher texts that associated with any attribute from the authority(corresponding to the revoked attribute) should be updated. Moreover, in our new attribute revocation method, both the key and the cipher text can be updated by using the same updated key, instead of requiring the owner to generate an update information for each cipher text, such that owners are not required to store each random number generated during the encryption. And we also highly improve the expressiveness of our access control scheme, where we remove the limitation that each attribute can only appear at most once in a cipher text. And we mainly used hybrid encryption for more security.

2.System model and security model

2.1.System model:

We consider a data access control system in multi-authority cloud storage. There are five types of entity in the system : a certificate authority(CA), attribute authorities(AAs), data owners(owner), the cloud server(server) and data consumer(users). The CA is a global trusted certificate authority in the system. It sets up the system and accepts the registration of all the users and AAs in the system.

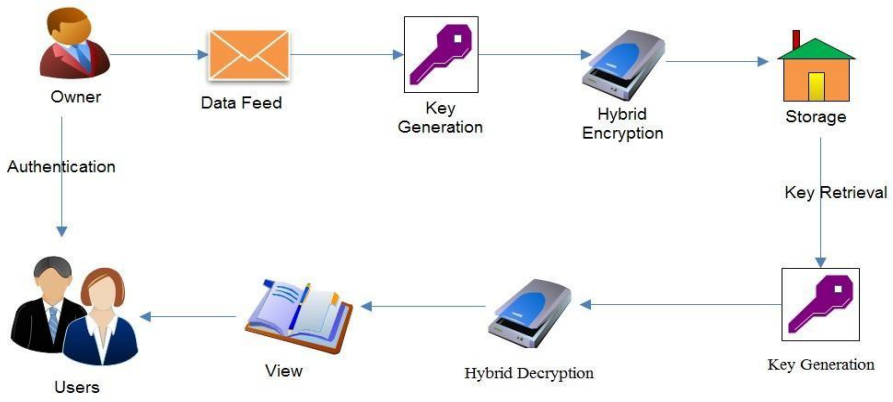
For each legal user in the system, the CA assigns a global unique user identity to it and also generates a global public key for this user. However, the CA is not involved in any attribute management and the creation of secret keys that are associated with attributes. For example, the CA can be the social security administration, an independent agency of the united states government.

Each AA is an independent attribute authority that is responsible for entitling and revoking user's attributes according to their role or identity in its domain. In our scheme, every AA can manage an arbitrary number of attributes.

Each AA has full control over the structure and semantics of its attributes. Each AA is responsible for generating a public attribute key for each attribute it manages and a secret key for each user reflecting his/her attributes.

Each user has a global identity in the system . A user may be entitled a set of attributes which may come from multiple attribute authorities. The user will receive a secret key associated with its attributes entitled by the corresponding attribute authorities.

Architecture Diagram



Each owner first divides the data into several components according to the logic granularities and each data component with different content keys by using hybrid encryption techniques. Then, the owner defines the access policies over attributes from multiple attributes authorities and encrypts the content keys under the policies. Then, the owner sends the encrypted data to the cloud server together with the cipher texts. They do not rely on the server to do data access control. But, the access control happens inside the cryptography. That is only when the user's attributes satisfy the access policy defined in the cipher text, the user is able to decrypt the cipher text. Thus, users with different attributes can decrypt different number of content keys and thus obtain different granularities of information from the same data. And the hybrid encryption

2.2. Frame work

The framework of our data access control scheme is defined as follows.

Definition 1(Framework of multi-authority access control scheme). *The framework of data access control scheme for multi-authority cloud storage system contains the following phases:*

Phase 1:system Initialization. *This phase of CA setup and AA setup with following algorithm*:

CA Setup(1^{λ})→ (GMK, GPP, (GPK uid , GPK' uid), (GSK uid , GSK' uid), *Certificate(uid)*}). The CA setup algorithm is run by the CA. It takes no input other than the implicit security parameter λ. It generates the global master key GMK of the system and the global public parameters GPP. For each user *uid*, it generates the user's global public keys (GPK uid, GPK'uid), the user's global secret keys (GSK uid, GSK' uid) and certificate *certificate(uid)* of the user.

AA Setup(μaid)→ (SK aid, PK aid,{VK Xaid, PK xaid} xaid€μaid). The attribute authority setup algorithm is run by each attribute authority. It takes the attribute universe *μ*aid managed by the AAaid as input. It outputs a secret and public key pair (SKaid, PKaid) of the AAaid and a set of version keys and public attribute keys {VK xaid, PK xaid}xaid€μaid for all the attributes managed by the AAaid.

Phase 2: Secret key generation by AAs.

 SkeyGen(GPP , GPKuid , GPK'uid , GSKuid , SKaid, Suid,aid , {VKxaid, PKxaid}xaid€Suid,aid) →SKuid,aid. The secret key generation algorithm is run by each AA. It takes as input the global public parameters GPP, the global public keys (GPKuid, GPK'uid) and one global secret key GSKuid of the user *uid* , the secret key SKaid of the AAaid, a set of attributes Suid,aid that describes the user *uid* from the AAaid and its corresponding version keys {VKxaid}. It outputs a secret key SKuid,aid for the user *uid* which is used for the decryption.

Phase 3: Data Encryption by Owners. *Owners first encrypt the data m with content key by using hybrid encryption methods, It has two types of algorithm one is AES algorithm and another one is Japsyt algorithm, Which is used to provide the more security in the cloud storage. Then they encrypt the content keys by running the following encryption algorithm:*

 Encrypt(GPP , {PKaidk}aidk€IA , *K*, A)→CT. The encryption algorithm is run by the data owner to encrypt the content keys. It takes as inputs the global public parameters GPP, a set of public keys {PKaidk}aidk€IA for all AAs in the encryption set IA3, the content key k according to the access policy A and outputs a cipher text CT. We will assume that the cipher text implicitly contains the access policy A.

Phase 4:Data Decryption by users. *Users first run the decryption algorithm to get the content keys, and use them to further decrypt the data.*

 Decrypt (CT , GPKuid , GSK'uid, {SKuid,aidk} aidk€IA)→k. The decryption algorithm is run by users to decrypt the cipher text. It takes as input the cipher text CT which contains an access policy A, a global public key GPKuid and a global secret key GSK'uid of the user *uid*, and a set of secret keys {SKuid,aidk}aidk€IA from all the involved AAs . If the attributes {Suid,aidk}aidk€IA of the user *uid* satisfy the access policy A, the algorithm will decrypt the cipher text and return the content key k.

Phase 5: Attribute Revocation. *This phase contains three steps: Update key generation by AAs, secret key Update by Non-revoked Users and Cipher text update by server.*

 UKeyGen(SKaid', xaid', VKxaid)→(VKxaid, UKs,xaid', UKc,xaid'). The update key generation algorithm is run by the corresponding AAaid' that manages the revoked attribute xaid. It takes as input the secret key SKaid' of AAaid', the revoked attribute xaid', and its current version key VKxaid'. It outputs a new version key VKxaid and the update key UKs,xaid' (for secret key update) and the update key UKc,xaid (for cipher text update).

 SKUpdate(SKuid,aid', UKs,xaid)→SKuid,aid'. The secret key update algorithm is run by each non-revoked user *uid.* It takes as input current secret key of the non-revoked user SKuid,aid' and the update key UKs,xaid. It outputs a new secret key SKuid,aid' for each non-revoked ser *uid.*

 CTUpdate(CT, UKc,xaid)→ CT. The cipher text update algorithm is run by the cloud server. It takes as inputs the cipher texts which contain the revoked attribute xaid', and the update key UKc,xaid'. It outputs new cipher text CT which contain the latest version of the revoked attribute xaid'.

2.3.Security Model

In multi-authority cloud storage system , we make the following assumption:

 The CA id fully trusted in the system. It will not collude with any user , but it should be prevented from decrypting any cipher texts by itself.

- Each AA is trusted but can be corrupted by the adversary.
- The server is curious but honest. It is curious about the content of the encrypted data or the received message , but will execute correctly the task assigned by each attribute authority.
- Each user is dishonest and may collude to obtain unauthorized access to data.

We now describe the security model for our revocable multi-authority CP-ABE systems by the following game between a challenger and an adversary. Similar to the identity based encryption schemes, the security model allows the adversary to query for any secret keys and update keys that cannot be used to decrypt the challenge cipher text. We assume that the adversaries can corrupt authorities only statically similar. But key queries are made adaptively. Let SA denote the set or all the attribute authorities. The security game is defined as follows.

Setup: The global public parameters are generated by running CA setup algorithm. The adversary specifies a set of corrupted attributes authorities S'A ç SA. The challenger generates the public keys by running the attributes authority setup algorithm and generates the secret keys by running attribute authorities in SA-S'A' the challenger only sends the public keys to the adversary. The adversary can also get the global public parameter.

Phase 1: The adversary makes secret key queries by submitting pairs (*uid*, S*uid*) to the challenger , where Suid= {Suid,aidk}aidk€sA-s' is a set of attributes belonging to several uncorrupted AA's, and *uid* is a user identifier.

Phase 2: The adversary may query more secret keys and update keys, as long as they do not violate the constraints on the challenge access structure (M*, p*) and the following constraints: None of the updated secret keys is able to decrypt the challenged cipher text.

3. Our data access control scheme:

In this section, we first give an overview of the challenges and techniques. Then, we propose the detailed construction of our access control scheme which consists of five phases: System Initialization, Key Generation, Data Encryption, Data Decryption and Attribute Revocation.

3.1 Overview

To design the data access control scheme for multi-authority cloud storage systems, the main challenging issue is to construct the underlying Revocable Multi-authority CP-ABE protocol. In Chase proposed a multi-authority CP-ABE protocol, however, it cannot be directly applied as the underlying techniques because of two main reasons:

1) Security Issue: Chase's multi-authority CP-ABE protocol allows the central authority to decrypt all the cipher texts, since it holds the master key of the system.

2) Revocation Issue: Chase's protocol does not support attribute revocation.

We propose a new revocable multi-authority CP-ABE protocol based on the single-authority CP-ABE proposed by Lewko and Waters. That is we extend it to multi-authority scenario and make it revocable. We apply the techniques in Chase's multi-authority CP-ABE protocol to tie together the secret keys generated by different authorities for the same user and prevent the collusion attack. Specifically, we separate the functionality of the authority into a global certificate authority (CA) and multiple attribute authorities (AAs). The CA sets up the system and accepts the registration of users and AAs in the system. It assigns a global user identity *uid* to each user and a global authority identity aid to each attribute authority in the system. Because the *uid* is globally unique in the system, secret keys issued by different AAs for the same *uid* can be tied together for decryption. Also, because each AA is associated with an *aid*, every attribute is distinguishable even though some AAs may issue the same attribute. To deal with the security issue, instead of using the system unique public key (generated by the unique master key) to encrypt data, our scheme requires all attribute authorities to generate their own public keys and uses them to encrypt data together with the global public parameters. This prevent the certificate authority in our scheme from decrypting the cipher texts.

To solve the attribute revocation problem, we assign a version number for each attribute. When an attribute revocation happens, only those components associated with the revoked attribute in secret keys and cipher texts need to be updated. When an attribute of a user is revoked from its corresponding AA, the AA generates a new version key for this revoked attribute and generates an update key. With the update key, all the users, except the revoked user, who hold the revoked attributes can update its secret key

By using the update key, the components associated with the revoked attribute in the cipher text can also be updated to the current version. To improve the efficiency, we delegate the workload of cipher text update to the server by using the proxy re encryption method, such that the newly joined user is also able to decrypt the previously published data, which are encrypted with the previous public keys, if they have sufficient attributes (Forward Security). Moreover, by updating the cipher texts, all the users need to hold only the latest secret key, rather than to keep records on all the previous secret keys.

4.Security Analysis:

Backward Security:

During the secret key update phase, the corresponding AA generates an update key for each non-revoked user. Because the update key is associated with the user's global identity *uid*, the revoked user cannot use update keys of other non- revoked users to update its own secret key, even if it can compromise some non-revoked users. Moreover, suppose the revoked user can corrupt some other AAs (not the AA corresponding to the revoked attributes), the item $H(x_{aid})^{r}x_{aid}\beta_{aid}\gamma_{aid}$ in the secret key can prevent users from updating their secret keys with update keys of other users, since γ_{aid} is only known by the AA_{aid} and kept secret to all the users. This guarantees the backward security.

Forward Security:

After each attribute revocation operation, the version of the revoked attribute will be updated.

When new users join the system, their secret keys are associated with attributes with the latest version. However, previously published cipher texts are encrypted under attributes with old version. The cipher text update algorithm in our protocol can update previously published cipher texts into the latest attribute version, such that newly joined users can still decrypt previously published cipher texts, if their attributes can satisfy access policies associated with cipher texts. This guarantees the forward security.

Hybrid Encryption:

Hybrid encryption is a mode of encryption that merges two or more encryption system. It is considered a highly secure type of encryption as long as the public and private key are fully secure. The combination of encryption methods has various advantages. One is that a connection channel is established between two user's sets of equipment. Users then have the ability to communicate through hybrid encryption jasypt is used to slow down the encryption process , but with the simultaneous use of AES encryption, both forms of encryption are enhanced. The result is the added security of the transmittal process along with overall improved system performance.

1).AES Encryption:

- The Advanced Encryption Standard or AES is a symmetric block cipher used by the U.S government to protect classified information and is implemented in software and hardware throughout the world to encrypt sensitive data.
- AES comprises three block cipher, AES-128, AES-192 and AES-256. Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128, 192, 256-bits, respectively.(Rijndael was designed to handle additional block sizes and key lengths, but the functionally was not adopted in AES.) Symmetric or secret-key ciphers use the same key for encrypting and decrypting , so both the sender and the receiver must know and use the same secret key. All key lengths are deemed sufficient to protect classified information up to the "secret" level with "Top secret" information requiring either 192-bits or 256-bits key length. There are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. A round consist of several processing steps that include substitution, transposition and mixing of the input

plaintext and transform it into the final output of cipher text.

2).Jasypt Encryption:

- Jasypt is a java library which allows the developer to add basic encryption capabilities to his/her projects with minimum effort, and without the need of having deep knowledge on how cryptography works.
- High-security, standards-based encryption techniques, both for unidirectional and bidirectional encryption. Encrypt passwords, texts, numbers, binaries...
- Transparent integration with **Hibernate**.
- Suitable for integration into **Spring**-based applications and also transparently integrable with **Spring Security**.
- Integrated capabilities for encrypting the configuration of applications (i.e. datasources).
- Specific features for **high-performance encryption** in multi-processor/multi-core systems.
- Open API for use with any JCE provider.

5.Performance Analysis:

In this section, we analyze the performance of our scheme by comparing with the Ruj's DACC scheme and our previous scheme in the conference version , in terms of storage overhead, communication cost and computation efficiency.

Table

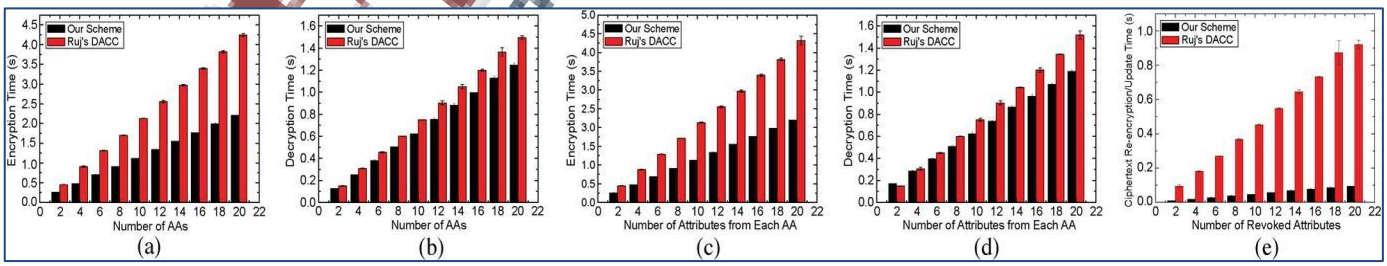
Computation cost for attribute revocation

Operation	In [2013]	In[2014]	Our
Key Update	None	$n_{non,x}\|p\|$	$N_{non,x}\|p\|$
CT Update	$(n_{c,x}, n_{non,x}+1)\|p\|$	$n_{c,aid}\|p\|$	$2\|p\|$

5.1. Storage Over head:

1) Storage Overhead on Each: AA Each AA needs store the information of all the attributes in its domain. Besides in [14], each AA $_{aid}$ also needs to store the secret keys from all the owners, where the storage overhead on each AA is also linear to the total number of owners n_o in

the system. In our scheme, besides the storage of attributes, each AA $_{aid}$ also needs to store a public key and a secret key for each user in the system. Thus, the storage overhead on each AA in our scheme is also linear to the number of users n_u in the system.



2) Storage Overhead on Each Owner: The public parameters contribute the main storage overhead on the owner. Besides the public parameters, in [13] owners are required to re-encrypt the cipher texts and In [14] owners are required to generate the update information during the revocation, where the owner should also hold the encryption secret for every cipher text in the system. This incurs a heavy storage overhead on the owner, especially when the number of cipher text is large in cloud storage systems.

.

3) Storage Overhead on Each User: The storage overhead on each user in our scheme comes from the secret keys issued by all the AAs. However, in [13], the storage overhead on each user consists of both the secret keys issued by all the AAs and the cipher text components that associated with the revoked attribute x, because when the cipher text is re-encrypted, some of its components related to the revoked attributes should be sent to each non-revoked user who holds the revoked attributes. In [14] the user needs to hold multiple secret keys for multiple data owners, which means that the storage overhead on each user is also linear to the number of owners n_O in the system.

4) Storage Overhead on Server: The cipher texts contribute the main storage overhead on the server (here we do not consider the encrypted data which are encrypted by the hybrid content keys).

5.2 Communication Cost

The communication cost of the normal access control is almost the same. Here, we only compare the communication cost of attribute revocation in table. The communication cost of attribute revocation in [13] is linear to the number of cipher texts which contain the revoked attribute. In [14] the communication overhead is linear to the total number of attributes $n_{c,aid}$ belongs to the AA *aid* in all the Cipher texts. It is not difficult to find that our scheme incurs much less communication cost during the attribute revocation.

5.3 Computation Efficiency

We implement our scheme and DACC scheme [13] on a Linux system with an Intel Core 2 Duo CPU at 3.16GHz and 4.00 GB RAM. The code uses the Pairing-Based Cryptography (PBC) library version 0.5.12 to implement the access control schemes. We use a symmetric elliptic curve α-curve, where the base field size is 512-bit and the embedding degree is 2. The α-curve has a 160-bit group order, which means p is a 160-bit length prime. All the simulation results are the mean of 20 trials.

We compare the computation efficiency of both encryption and decryption in two criteria: the number of authorities and the number of attributes per authority. Fig.3a describes the comparison of encryption time versus the number of authorities, where the involved number of attributes per authority is set to be 10. Fig.3c gives the encryption time comparison versus the number of attributes per authority, where the involved number of authority is set to be 10. It is easy to find that our scheme incurs less encryption time than DACC scheme in [13].
Fig.3b shows the comparison of decryption time versus the number of authorities, where the number of attributes the user holds from each authority is set to be 10. Suppose the user has the same number of attributes from each authority, Fig.3d describes the decryption time comparison versus the number of attributes the user holds from each authority. In Fig.3d , the number of authority for the user is fixed to be 10. It is not difficult to see that our scheme incurs less decryption on the user than DACC scheme in [13].

Fig.3.e The time of cipher text update/re encryption versus the number of revoked attributes, and our scheme is more efficient than [13]. The cipher text update/re-encryption contributes the main computation overhead of the attribute revocation. In our conference version [14], when an attribute is revoked from its corresponding authority $AA_{aid'}$, all the cipher texts which are associated with any attributes from $AA_{aid'}$ should be updated. In this paper, however, the attribute revocation method only requires the update of cipher texts which are associated with the revoked attributes.

6.Conclusion

In this paper, we proposed a revocable multi- authority CPABE scheme that can support efficient attribute revocation and hybrid encryption. Then, we constructed a secure and safety and effective data access control scheme for multi-authority cloud storage systems. We also proved that our scheme was revocable multi-authority CPABE and hybrid applied in any remote storage systems and online social networks etc

7. References

[1] kan yang and xiaohua jia "Expressive and efficient and revocable data access control for multi- authority cloud storage" IEEE transaction on parallel and distributed system. July 2014

[2] P. Mell and T. Grance, ''The NIST Definition of Cloud Computing,'' National Institute of Standards and Technology, Gaithersburg, MD, USA, Tech. Rep., 2009.

[3] J. Bethencourt, A. Sahai, and B. Waters, ''Cipher text-Policy Attribute-Based ,'' in Proc. IEEE Symp. Security and privacy (S&P'07), 2007, pp. 321-334.

[4] B. Waters, ''Cipher text-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization,'' in Proc. 4th Int'l Conf. Practice and Theory in Public Key Cryptography (PKC'11), 2011, pp. 53-70.

[5] V. Goyal, A. Jain,O. Pandey, andA. Sahai, ''Bounded Cipher text Policy Attribute Based Encryption,'' in Proc. 35th Int'l Colloquium on Automata, Languages, and Programming (ICALP'08), 2008, pp. 579-591.

[6] A.B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B.Waters, ''Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption,'' in Proc. Advances in Cryptology-EUROCRYPT'10, 2010, pp. 62-91.

[7] M. Chase, ''Multi-Authority Attribute Based Encryption,'' in Proc. 4th Theory of Cryptography Conf. Theory of Cryptography (TCC'07), 2007, pp.

[8] M. Chase and S.S.M. Chow, ''Improving Privacy and Security in Multi-Authority Attribute-Based Encryption,'' in Proc. 16th ACM Conf. Computer and Comm. Security (CCS'09), 2009, pp. 121-130.

[9] A.B. Lewko and B. Waters, ''Decentralizing Attribute-Based Encryption,'' in Proc. Advances in Cryptology-EUROCRYPT'11, 2011, pp. 568-588.

[10] S. Yu, C. Wang, K. Ren, and W. Lou, ''Attribute Based Data Sharing with Attribute Revocation,'' in Proc. 5th ACM Symp. Information, Computer and Comm. Security (ASIACCS'10), 2010, pp. 261-270.

[11] M. Li, S. Yu, Y. Zheng, K. Ren, andW. Lou, ''Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption,'' IEEE Trans. Parallel Distributed Systems, vol. 24, no. 1, pp. 131-143, Jan. 2013.

[12] J. Hur and D.K. Noh, ''Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems,'' IEEE Trans. Parallel Distributed Systems, vol. 22, no. 7, pp. 1214-1221, Julj,2011.

[13] S. Jahid, P. Mittal, and N. Borisov, ''Easier: Encryption-Based Access Control in Social Networks with Efficient Revocation,'' in Proc. 6th ACM Symp. Information, Computer and Comm. Security (ASIACCS'11), 2011, pp. 411-415.

[14] S. Ruj, A. Nayak, and I. Stojmenovic, ''DACC: Distributed Access Control in Clouds,'' in Proc. 10th IEEE Int'l Conf. TrustCom, 2011, pp. 91-98.

[15] K. Yang and X. Jia, ''Attribute-Based Access Control for Multi-Authority Systems in Cloud Storage,'' in Proc. 32th IEEE Int'l Conf. Distributed Computing Systems (ICDCS'12), 2012, pp. 1-10.

[16] D. Boneh and M.K. Franklin, ''Identity-Based Encryption from the Weil Pairing,'' in Proc. 21st

Ann. Int'l Cryptology Conf.: Advances in Cryptology- CRYPTO'01, 2001, pp. 213-229.

[17] A.B. Lewko and B. Waters, ''New Proof Methods for Attribute- Based Encryption: Achieving Full Security through Selective Techniques,'' in Proc. 32st Ann. Int'l Cryptology Conf.: Advances in Cryptology - CRYPTO'12, 2012, pp. 180-198.