ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 2, Issue 4, April 2013

Amplifying Security for Cipher-Text Policy Attribute-Based Data sharing

Lekshmy Aravind * and M.Victor Jose #
- **Student of M.E(CSE), Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu, India**
 # Department of CSE, Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu, India

***Abstract*— Several data hiding techniques have been freshly figure out as they could help to manipulate as part of the security benefits. The randomization is expected to increase the security of the system and also increase the capacity. Concern to data security also arises. One of the most auspicious cryptographic solutions to this issue is Cipher text policy attribute-based encryption (CP-ABE).Ciphertext policy attribute based encryption is an approach in which a user with a enigmatical key (containing attributes) is able to decrypt a message only if the attributes in the policy meet those defined in the secret key.CP-ABE is used to control outsourced data sharing, it confronts two obstacles. Key generation is one of the main factors in datasharing.In key generation the major drawback which is called as the key escrow problem and in my proposed system it is overcome by twin calculation protocol between the key generation center and the data storing center. In proposed system by applying encryption in the data sharing system introduces another challenge with regard to the user revocation and it is resolving using also resolving the issue of revocation using technique Alternate dynamic revocation of cipher policy attribute based encryption mechanism. It is possible to remove access from a user without issuing new keys to other users or re-encrypting existing cipher texts. In this proposed system individually make a motion to newer approach to add, update, modify or delete the value of peculiar attribute freely without knowledge of other attribute.**

***Index Terms*— CP-ABE, Removing escrow, Revocation, Data privacy, Fine-grained access control**

I. INTRODUCTION

Data sharing is the proceedings of making data used for learned research available to other pragmatists. Rapid flowering of technology in network and computing give power to many people for sharing their data with external storages easily. People can contribute their enthusiasm with friends by uploading their personal things like data, pictures or messages into the online social networks such as Face book and LinkedIn; or upload highly sensitive personal health records into online data servers such as Microsoft Health Vault, Google Health for ease of sharing and cost saving. The security management in Personal Health Record is shown in the Fig. 1.In the midst of the multiplicity of the users in a typical distributed system, devising a befitting mechanism to control the access to the resources is non-trivial. Over the years, numerous, complex access-control mechanisms have been proposed to ensure the authorized access. These approaches are based either on a cryptographic technique or on any non-cryptographic mechanism. Amongst the cryptographic approaches, one of the prevalent and popular approach is ABE Eminently Ciphertext-Policy Attribute-Based Encryption give power to the people in such a way that an encryptor to define the attribute set over a universe of attributes that a decryptor needs to possess in order to decrypt the cipher text and enforce it on the contents. In Ciphertext Policy ABE [1][2], a user's secret key is based on a defined set of attributes (i.e. credentials of that user) and the ciphertexts is generated based on the defined policy. Thus, a user is able to decrypt the ciphertext only if the attributes in the secret key of user satisfy the policy defined in the ciphertext. CP-ABE in data sharing system has many confrontations. In CP-ABE, the key generation center (KGC) generates private keys of users by applying the KGC's master secret keys to users' associated set of attributes. The main profit of this approach is to largely reduce the need for processing and storing public key certificates under traditional public key infrastructure (PKI). Moreover, CP-ABE facing two stumbling blocks. Firstly key escrow problem [1], which means the data owner, must trust the third party authorities. Secondly, the issue of attribute revocation of CP-ABE schemes [1], which suffers from such problems revocation, low scalability and high calculative complexity is cumbersome. Since some users may change their associate attributes at some point of time, or private keys might be negotiate, key revocation or refurbish for each attribute is necessary in order to make systems secure. This issue is even more difficult conspicuously in ABE, since each attribute is conceivably shared by multiple users (henceforth, we refer to such a set of users as an attribute group). In the existing works another limitations catched out is there is no pillar for charismatic attributes.Inspirated by the above limitation in this work we proposed refined approach for supporting charismatic attributes. Dynamic characteristics includes add, delete, update and modify. One must be able to assign desired value to chosen a dynamic attribute. The modification of one attribute value must be independent of the same to the other. A user must not be compelled to re-produce the proof of old attributes and their values when updating the same.

Manuscript received Mar, 2013.

***Lekshmy Aravind**, Student of M.E(CSE), Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu, India,Tel: +91 9633160990, Fax: +91 4651 257266*

***M.Victor Jose**, Department of M.E(CSE), Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu, India,Tel: +91 9633160990, Fax: +91 4651 257266.*

All Rights Reserved © 2013 IJSETR

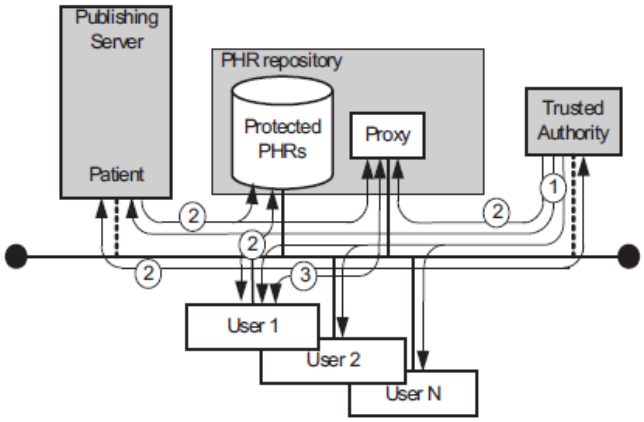


Fig. 1 Security Management of PHRs

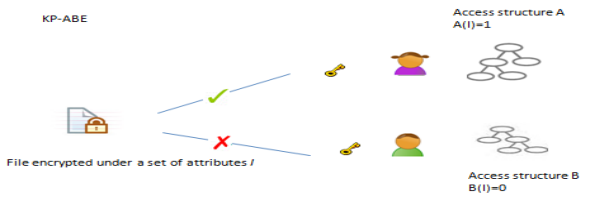
II.BACKGROUND

In general public key cryptosystem there are two keys, one is public key and another one is private key. For example Smith uses Helen public key to encrypt a message to Helen. Helen uses her private key to decrypt the message. How does Smith know the public key of Helen? Helen may have given the public key using a secure medium. This works only if there is already some trust/familiarity between both Smith and Helen. If Smith and Helen do not know each other, the above method fails. In a strange open system, we need a trusted third party to uniquely bind public keys to users. In such situations we need a public key cryptosystem. A public key infrastructure has one more trusted entities called Certification Authorities. For example, VeriSign is a Certification Authorities. Certification Authorities issues Helen a certificate (which contains the public key of Helen) signed by the Certification Authorities's public key after verifying Helen's credentials. Smith can now retrieve Helen's certificate and verify the authenticity by checking the signature. Due to various reasons certificates may have to be revoked. For example, if Helen's private key is robbed, she will have to ask the Certification Authorities to revoke its certificate. Then how does Smith know if a certificate is revoked? The Certification Authorities maintains a revocation list which allows Smith to verify if a given certificate is revoked or not. In general public key cryptosystem is somewhat cumbersome as one need to retrieve certificates, inspect revocation list, and then encrypt. A number of brilliant researchers came up with the idea of using user identities (for example, your email address) as public keys [2, 3]. Such systems are called identity based encryption. How an identity based encryption works? The idea to eliminate the need of certificate is mentioned below. A trusted third party called Key Generation Server. Key Generation Center generates a private key and the identity acts as the public key. For example, Helen's identity is her email address Helen@example.com. Helen uses this identity to obtain a private key from the Key Generation Server. Now Smith encrypts a message using Helen's email. Only Helen can decrypt the message since the identity, the public key, Helen@example.com belongs to Helen and only she can obtain the private from the Key Generation Server. Notice that there is a huge trust placed on the Key Generation Server. The security of the whole system relies on the security of the Key Generation Center and how well the Key Generation Center authenticates users before issuing private keys. The view point of identification based encryption was further improved to support much preferred systems. The approach of attribute-based encryption (ABE) has been introduced by Sahai and Waters [3]. Attribute-based encryption can be considered as a generalization of identity based encryption [2, 3] (IBE), the encryption is based on some identity. Thus, ABE is more clear cut than IBE. In an attribute-based encryption system, the plaintext is encrypted with a set of attributes. The Key Generation Center, which generates the master key, distributes different private keys to users after authenticating the attributes they bear. Thus, these private keys are associated with the set of attributes each user maintains. In its basic form, a user can decrypt a cipher text if and only if it meets the conditions between the attributes of the ciphertext and the user's key. For example, Helen carry the attributes "role = doc" and "age > 20". Now Smith encrypts a message using the attributes ("role = student" AND "age > 20"). Helen can decrypt the message as she satisfies both attributes. Smith encrypts another message using the attributes ("role = professor" OR "role = staff"). Helen cannot decrypt the message as she does not satisfy the policy. The initial ABE system is limited only to threshold policies where there should be at least k out of n attributes common between the attributes used to encrypt the plaintext and the attributes users possess. Pirretti et al. [5] gave an implementation of such a threshold ABE system using a variant of the Sahai-Waters Large Universe construction [3]. For example, Smith encrypts a message for any 3 attributes out of the 5 attributes {a1, a2, a3, a4, a5}. Helen has the attributes {a1, a2, a4, a5} and Eve has {a1, a2}. While Helen can decrypt Smith's message, Eve cannot as she does not satisfy the threshold policy. Many versions have been introduced to provide more expressive attribute based encryption systems. Out of them there are two important variants.

1. Key Policy ABE
2. Cipher Policy ABE

The idea behind KP-ABE and CP-ABE are explained using diagram.

In the below diagram, Key Policy ABE Smith encrypts a file using some set of attributes. Using access structure it is defining. Helen and Tim try to decrypt the message. Helen's attribute has to satisfy with access structure, and then only she can decrypt the key and the file. But Tim's attribute is not satisfying the access structure; hence he can't decrypt the key and the document



. **Fig. 2 KP-ABE**

All Rights Reserved © 2013 IJSETR

In the below diagram, Cipher Policy ABE reverses the execution of Key Policy ABE. Here the encryption is related to access structure. The key generation center distributes the private keys for the users. Users can decrypt only if attributes satisfy the access structure. Here the full control goes to the data owner.

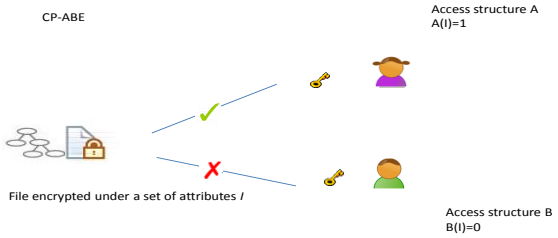


Fig. 3 CP-ABE

III. RELATED WORK

Cipher text-Policy Attribute-Based Encryption (CP-ABE), a user secret key is associated with a set of attributes, and the encrypted data is correlated to access policy. The user can decrypt the encrypted data if and only if the attribute set satisfies the access policy of her secret key. In many distributed systems a user can access data if a user acquires a certain set of attestation or attributes. Presently, one of the method for demand such policies is to employ a trusted server to save the data and mediate access control [2]. They are created public key revocation encryption systems with small cryptographic private and public keys. Their systems have two important features related to size of public and private key. Initially, public keys are short and enable a user to create an encrypted message that revokes an unbounded number of users. Secondly, the cryptographic key that must be saved securely on the receiving devices is small. Maintaining the size of private key storage will often be stored in tamper-resistant memory and more costly. This can be critical in small devices such as sensor nodes, in which cost is high .Identity-based encryption is an exciting alternative to public-key encryption, as it eliminates the need for a Public Key Infrastructure. The senders using an identity-based encryption do not need to look up the public keys and the corresponding certificates of the receivers, the identities (e.g. emails or IP addresses) of the latter are sufficient to encrypt. Whatever change in the settings of public key infrastructure and identity based may result to user revocation. Using time periods while encrypting is one of the solution for this needs .Cipher text-Policy Attribute Based Encryption (CP-ABE) is a promising cryptographic primitive for fine-grained access control of shared data. In CP-ABE, each user is related with a set of attributes and data are encrypted with access structures on attributes. It is possible for the user to decrypt a encrypted message if and only if his attributes meets the access structure. The communication model is one-to-one.

A. Removing Escrow

In the case of most attribute based encryption schemes are constructed based on single trusted authority and they have the power to generate the whole private keys of users with its master secret information [1], [3], [2], [6], [13], [14], [15]. Thus, the key escrow problem is inherent such that the single trusted authority can decrypt every encrypted message addressed to users in the system by generating their secret keys at any time. Chase et al. [11] presented a distributed KP-ABE scheme that solves the key escrow problem in a multi-authority system. In this approach, all attribute which are not joined, authorities are participating in the key generation protocol in a distributed way such that they cannot pool their data and link multiple attribute sets belonging to the same user. Major disadvantage here is the performance degradation, because there is no centralized authority with master secret information, all attribute authorities should communicate with the other authorities in the system to generate a user's secret key. This results in $O(N^2)$ communication overhead on the system setup phase and on any rekeying phase, and requires each user to store $O(N^2)$ additional auxiliary key components besides the attributes keys, where N is the number of authorities in the system. Recently, Chow [10] proposed an anonymous private key generation protocol in identity-based literature such that the KGC can issue a private key to an authenticated user without knowing the list of users' identities. It seems that this anonymous private key generation protocol works properly in ABE systems when we treat an attribute as an identity in this construction. However, we found that this cannot be adapted to ABE systems due to mainly two reasons. First, in Chow's protocol, identities of users are not public anymore, at least to the KGC, because the KGC can generate users' secret keys otherwise. Since public keys (attributes in the ABE setting) are no longer 'public', it needs additional secure protocols for users to obtain the attribute information from attribute authorities. Second, since the collusion attack between users is the main security threat in ABE, the KGC issues different personalized key components to users by blinding them with a random secret even if they are associated with the same set of attributes. The random secret is unique and should be consistent with the same user for any possible attribute change (such as adding some attributes) of the user. However, it is impossible for the KGC to issue a personalized key component with the same random secret as that of attribute key components to a user, since the KGC can by no means know which random secrets (used to issue a set of attributes key components) are assigned to which users in the Chow's key issuing protocol.

B. Revocation

Bethencourt et al. [2] and Boldyreva et al. [16] proposed first key revocation mechanisms in CP-ABE and KPABE settings respectively. These schemes enable an attribute key revocation by encrypting the message to the attribute set with its validation time. These attribute-revocable ABE schemes [2], [16], [5] have the security degradation problem in terms of the backward and forward secrecy [17]. They revoke attribute itself using timed rekeying mechanism, which is realized by setting expiration time on each attribute. In ABE systems, it is a considerable scenario that membership may change frequently in the attribute group. Then, a new user

All Rights Reserved © 2013 IJSETR

might be able to access the previous data encrypted before his joining until the data is re-encrypted with the newly updated attribute keys by periodic rekeying which is called backward secrecy. On the other hand, a revoked user would still be able to access the encrypted data even if he does not hold the attribute any more until the next expiration time which is called forward secrecy. This is called the window of vulnerability. Today, the importance of immediate user revocation (rather than attribute revocation) have been taken notice of in many practical ABE-based systems [19], [18].The user revocation can be done by using ABE that supports negative clauses, proposed by Ostrovsky et al. [6]. To do so, one just adds conjunctively the AND of negation of revoked user identities (where each is considered as an attribute here). The major flaw of this scheme is that the size of private key incremented by a multiplicative factor of log *n*, here *n* indicates maximum number of attributes. Lewkoet al. [18] proposed more efficient instantiations of the Ostrovsky et al.'s framework [6] for non monotonic ABE, where public parameters is only $O(1)$ and private keys for access structures includes t leaf attributes is of size $O(t)$. However, these user-revocable schemes also have a limitation with regard to the availability. When a user is revoked even from a single attribute group, he loses all the access rights to the system, which is not desirable in many pragmatic scenarios since the other attributes may be still valid. Attrapadung et al. suggested other user revocable ABE schemes addressing this problem by combining broadcast encryption schemes with attribute based encryption schemes. In this scheme, the data owner should take full charge of maintaining all the membership lists for each attribute group to enable the direct user revocation. It is not applicable to the data sharing system, because the data owners will no longer be directly in control of data after storing their data to the external storage server. Yu et al. also recently addressed the user revocation in the ABE based data sharing system. In this scheme, the user revocation is realized using proxy re-encryption by the data server. However, in order to revoke users, the KGC should generate all secret keys including the proxy key on behalf of the data server. Then, the server would re-encrypt the ciphertext under the proxy key received from the KGC to prevent revoked users from decrypting the ciphertext. Thus, the key escrow problem is also inherent in this scheme, since the KGC manages all secret keys of users as wells as the proxy keys of the data server.

C. STATIC ATTRIBUTES

In [7] authors proposed the approach based on AND gate only where attributes have negative and positive values. In [8] authors support the non-monotonic access structure. In [9] authors give the construction of bounded CP-ABE scheme. In [10] authors the construction of hidden access structure where no one can able to see the policy attached which ciphertext. Till now all the approaches dealing with static attributes. Static attributes means it is impossible to add, delete, modify or updating the attributes. In this work we are proposing updating the attributes. Updating the attribute included adding, modifying, deleting, updating.Here we can change the value in secret key.

IV. CONTRIBUTION

In this work, we are suggesting a new CP-ABE scheme for a secure data sharing system, which features the following achievements.

A. REMOVE KEY ESCROW SCHEME

Key escrow is an inherent property in the current proposed attribute based encryption. In this paper, a scheme which removes the key escrow and maintaining some important properties of the ABE [3][4][2]. Also some cryptosystems are introduced based on variant including an authenticated key agreement. The KGC and the data storing center are involved in the user key issuing protocol. In the protocol, a user is required to contact the two parties before getting a set of keys. The key generation center is responsible for authenticating a user and issuing attribute keys to him if the user is entitled to the attributes. For the given password, hashing technique is applied to generate the Key. Here for hashing technique, to generate the key SHA1 algorithm is used.

A.1.1. TWIN CALCULATION PROTOCOL FOR CP-ABE

The KGC and the data storing center are involved in the user key issuing protocol. In the protocol, a user is required to contact the two parties before getting a set of keys. The KGC is responsible for authenticating a user and issuing attribute keys to him if the user is entitled to the attributes. The secret key is generated through the secure 2PC protocol between the KGC and the data storing center. They engage in the arithmetic secure 2PC protocol with master secret keys and issue independent key components to a user. Then, the user is able to generate the whole secret keys with the key components separately received from the two authorities. The secure 2PC protocol deters them from knowing each other's master secrets so that none of them can generate the whole secret keys of a user alone. The data storing center probabilistically outputs the public and private key pair. The KGC and the data storing center are involved in the key generation protocol. The value should be unique and secret to the user, which should be consistent .Then, the KGC and the data storing center engage in a secure 2PC protocol. When one member is compromised, the group can still continue with its secure communication by excluding the compromised member. The last characteristics is the dynamic compromised property, means the group key agreement scheme property, retains both accuracy and efficiency even if the group key retains agreement scheme. To offer data privacy, an effective approach is to require all group members to establish a common secret group key, which is held only by group members, but not outsiders, for encrypting the transmitted data.

A.1.2. KEY REVOCATION

The immediate user revocation can be done via the alternate dynamic revocation of cipher policy attribute based encryption mechanism. Access control to data in general affair is typically enforced through connecting monitors. Since more and more resources are outsourcing the data the major challenge comes trusts the third party. Because of this popularity to cryptography also increases. The major threat to attribute based encryption is revocation. To forward this threat, we propose alternate dynamic revocation of cipher policy attribute based encryption mechanism, an architecture

All Rights Reserved © 2013 IJSETR

that supports fine grained access control policies and dynamic group membership. It is built using attribute-based encryption; a key and new element of architecture. Using this work it is possible to remove access from a user without issuing new keys to other users or re-encrypting existing ciphertexts as seen in the existing system. We achieve this by introducing a proxy that participates in the decryption process and impel revocation conditions. The proxy that we are using is minimally trusted and cannot decrypt ciphertexts or provide access to previously revoked users. This design makes use of a minimally trusted proxy, which handles revoked users and attributes. Upon revocation, no new key is generated for any user, neither is the existing data re-encrypted.

As we observed that the approaches till now has some kind of limitation with the static and dynamic attribute. So that they are not trustworthy as well as not applicable in real life. To deal with this problem we proposed a new approach in which CA extract the old values from the secret key and change the value of required attribute and replace the new value with the old value and give secret key to user.

V. DATA SHARING ARCHITECTURE

As shown in Fig. 3, the architecture of data sharing system consists of the following entities:

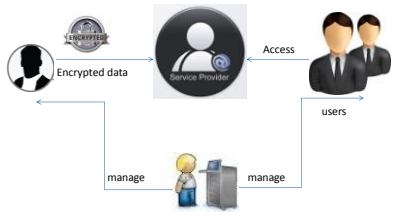


Fig. 3. Architecture of a Data Sharing System

A. Data Owner

It is a client who owns data, and wishes to upload it into the external data storing centre for ease of sharing or for cost saving. It is the responsibility of a data owner to defining (attribute based) access policy, and make sure on its own data by encrypting the data under the policy before distributing it. Data Owner encrypt the file which he wants to send using the key which is generated by key generation center.

B. Data Storing Centre

This is an entity provides data sharing services. It is used to controlling the accesses from outside users. The data storing centre is another key authority that generates personalized user key along with key generation center and distributes or revokes attribute group keys to valid users.. Data storing centre store the data. Data Storage Centers provides offsite record

C. User

This is an entity who needs permission to access the data encrypted by data owner. User can decrypt data if and only if he should satisfy the access polices of attributes defined by the data owner, and it is impossible to revoke in any of the attribute groups, then he will be able to decrypt the cipher text and obtain the data.

D. Key Generation Center

This is a key authority responsible to generate public and secret parameters for Cipher Policy ABE. Main role of key generation center are issuing, revoking, and updating attribute keys for users. It grants access rights to individual users on the basis of their attributes. Key generation is the process of generating keys for encryption and decryption purposes. A key is used to encrypt and decrypt whatever data is being encrypted or decrypted.

VI. CONSTRUCTION

A. ASSUMPTIONS AND BASICS

Before get into the technicalities of the construction, here providing some basic mathematical assumptions and details of CP-ABE and revocation scheme used in alternate dynamic revocation of cipher policy attribute based encryption mechanism.

Bilinear Pairing

Let G1, G2, and G_T be multiplicative cyclic groups of prime order p, and e a map $(G1 \times G2 \rightarrow G_T)$. Let g1and g2be generators of G1and G2respectively ($G_i = \langle g_i \rangle$).If $\forall u \in G1$, $v \in G2$and a, b ∈ Zp, $e(u, v)^{ab}$ and $e(g_1, g_2) \neq 1$, then e is called a bilinear pairing. If G1= G2, it is called a symmetric pairing, otherwise the pairing is asymmetric.

Access Structure

Let $\{P_1, P_2 \ldots Pn\}$ be a set of parties. A collection $A \subseteq 2\{P_1, P_2 \ldots Pn\}$.is monotone if $\forall$ B, C: if B ∈ A and B ⊆ C then *C* ∈ A. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) A of non-empty subsets of $\{P_1, P_2 \ldots Pn\}$.The sets in A are called the authorized sets, and the sets not in A are called the unauthorized sets. In CP-ABE schemes, the role of the parties is taken by the attributes. Thus, the access structure A will contain the authorized sets of attributes. From now on, by an access structure we mean a monotone access structure. The algorithms in CP-ABE due to the Bethencourt et al. are described below. Though CP-ABE uses symmetric pairing, it can be implemented using an asymmetric pairing as well.

- SETUP**:** The key authority KA generates a public key PK, and a master secret key MK:

$$PK = G1, g, h = g^{\beta}; \quad e(g,g)^{\beta}$$
$$MK = (\beta, g^{\alpha})$$

where random α, β∈Zp,G1 = <g> |G1|= p

The PK also contains an extra component $f = g^{1/\beta}$ to support attribute delegation.

ENCRYPT (PK, M, and τ): A policy is represented as an access tree structure τ with the attributes at leaves and threshold k-of gates at intermediate nodes. Each node is associated with a polynomial qx of degree d_x, where d_x is 1 less than the threshold value k of that node. The polynomials are of degree 0 for OR gates and leaves. The secret s (random s ∈ Zp) to blind the data M is associated with the polynomial at the root of the tree, i.e., $q_R(0) = s$.The sharing works in a top down manner: for all other nodes, $q_x(0) = q$ parent(x) (index(x)). Index(x) returns a number between 1 and num associated with x where num is the number of children of parent(x). Let Y be the set of leaf nodes in τ. The ciphertext CT is:

All Rights Reserved © 2013 IJSETR

CT = (τ,˜C = Me(g,g)α^s,C = h^s,∀y ∈ Y: Cy= $g^q{}_y(0)$,Cy= H(att(y))^{qy}(0))

Here, H: {0, 1} ∗→ G1is a hash function, modeled as random oracle, that maps string attribute to random element of G1.

- KEYGEN(MK, S): The secret key SK corresponding to a set of attributes S is (random r,rj∈ Zp):

SK =(D = g(α+r)/β,
∀j ∈ S: Dj= g^rH(j)$^r{}_j$,Dj= gr_j)

Dj, Dj for each attribute is blinded by rj, and all the components are tied together using r in D. This prevents attributes of different users from being combined together and provides collusion resistance.

- DECRYPT (CT, SK): The goal of decryption algorithm is to find out e (g, g) $^{\alpha s}$. It finds out the secret q_x (0) at each node x blinded by the random value r. A secret key SK that achieves d_R such secrets at the root R can solve the polynomial qRand decrypt the ciphertext. A recursive algorithm DecryptNode pairs Di and Di (from SK) with Cx and Cx (from CT) respectively and return e (g, g)^{rqx}(0)for each leaf node x in the τ in CT, iff i = attr(x). i ∈ Sis the set of attributes for which a user is assigned SK.

At each non-leaf node, Lagrange interpolation is used on at least k (the threshold value of the node) such e (g, g) rqz received from its children z, to calculate (g, g) ^{rqx} (0). Let A = e (g, g) rqR (0) = e (g, g)^{rs}. Then˜C, C, D and A are used in bilinear mapping to cancel out e (g, g) ^{rs}., and retrieve M. Further details can be found in [4].

- DELEGATE (SK,˜S). The delegate algorithm re-randomizes the relevant set of attributes˜S ⊆ S of a secret key SK assigned for some set of attributes S.It outputs a secret key ˜SK for the set of attributes ˜S.˜SK =(˜D = Df˜ r,∀k ∈˜S :˜Dk= Dkg˜ rH(k)˜ rk,˜Dk= Dkg˜ ^{rk}).

VII. PROPOSED CONSTRUCTION

In this proposed work five algorithms

SETUP: It will take implicit security parameter and output public parameter PK and master key MK.

KEYGEN (MK, L): The key generation algorithm runs by CA. It takes as input the master key of CA and the set of attributes L for user, then generate the secret key SK.

KEYUPDATE (MK, SK, old value, new value): The key updation algorithm runs by CA. It takes as input the master key of CA, old SK and old attribute value *old value*, and then updates the secret key SK by updating (add/delete/update) *old value* with *new value*.

ENCRYPT (PK, M, A): The encryption algorithm takes as input the message M, public parameter PK and access structure A over the universe of attributes. Generate the output CT such that only those users who had valid set of attributes that satisfy the access policy can only able to decrypt. Assume that the CT implicitly contains access structure A.

DECRYPT(PK,CT,SK) : The decrypt algorithm run by user takes input the public parameter, the ciphertext CT contains access structure A and the secret key SK contain of user attribute set S. If S satisfies the access tree then algorithm decrypt the CT and give M otherwise gives "ϕ".

A. *Alternate dynamic complete key revocation*

In this section we describe how to completely revoke keys from parties. That means, all the privileges granted by the key authority are revoked from one or more contact(s). This construction allows revocation of up to t users at a time since it is based on the scheme in [10] described before.

Intuition:

The master key MK contains a polynomial P of degree t. P(0) is used to blind users' secret keys. Each user u also gets a random share P(u) of P(0) in her key. The proxy key consists of t such shares and is used to convert a part of the ciphertext for decryption. Whenever access is revoked from someone, her share becomes a part of the proxy key, and eventually the converted ciphertext. Therefore, the revoked user does not have enough points, i.e. (t + 1) points to unblind her key and the ciphertext and decrypt it. However, non-revoked users can always combine their secret keys with the ciphertext and hence decrypt it. When no one is revoked, the proxy key consists of t random P(u) points. Since the revocation is based on polynomial secret sharing, and the degree of the polynomial is t, the scheme is limited to maximum t revocations. Though each time t different users can be revoked, the total number of users in the system is not limited.

- SETUP: The key authority KA randomly generates a polynomial P of degree t (the maximum number of revoked users) over Zp, sets the broadcast secret P (0) to be used after revocation, and randomly chooses α, β ∈ Zp. She generates PK and MK as follows:

PK = G1,G2,g1,g2,h =gβ1,e(g1,g2)α,MK = β,gα,2,P.

- ENCRYPT (PK, M, τ): Let Y be the set of leaf nodes in τ. Data M is encrypted to get the ciphertext CT. Other than the asymmetric groups, this algorithm works exactly the same as in BSW CP-ABE. CT = (τ,˜C = Me(g1,g2)αs,C = hs= gβs1,∀y ∈ Y : Cy= gqy(0)1, Cy= H(att(y))qy(0)= ghyqy(0)2) where H : {0,1}∗→ G2 and hy = logg2H(att(y))(used for notational convenience only).
- KEYGEN(MK, S): The algorithm KeyGen outputs the secret key corresponding to the set of attributes S, blinded by P(0) from MK. W e introduce an extra component— Dj—that in addition to attribute information contains user information. Without loss of generality, we assume use u^k receives this key.SK = (D,∀j ∈ S :D_j),where D = g(α+r)/β2,Dj= gr 2H(j)r^jP(0)= gr+hj r^jP (0) , D_j = gr1,D_j= (D_j)P(u^k)= gr^jP(u^k)
- PROXYREKEY (PK, MK, RL): Whenever the KA wants to revoke keys from social contacts, she creates a list of revoked users RL with the i identities u_i, i ∈ {1... t}, and evaluates the corresponding P (u_i) using MK. She gives the proxy key PXK to the proxy. In case of no or fewer than t revocations, the KA generates random x,P(x) other than the actual user identities, to make PXK of length t. PXK = ∀ui∈ RL :u_i,P(ui)
- CONVERT(PXK, ∀y ∈ Y : Cy, uk): The proxy uses its key PXK and the decryptor's identity uk to calculate Cy as follows: ∀i,j ∈ {1,...,t}, k / ∈ {1,...,t}, λi= u^k − u_{ij}=I uj (uj− ui), ∀y ∈ Y : Cy= (Cy)ti=1λiP(ui)= ghyqy(0)t i=1λiP(ui) 2 Since the user secret key SK is blinded by P(0), she needs Cy in addition to Cy and Cy for decryption. The proxy also calculates λk and gives it to the user u^k.
- (CT, SK): The decryption steps involve one extra pairing than BSW CP-ABE at each leaf node of the policy. For each leaf node x where i = attr(x), if i ∈ S, (S is the set of attributes

All Rights Reserved © 2013 IJSETR

for which SK is issued) then, Decrypt Node(CT,SK,x) = e(Cx,Di)e(i,Cx)λke(Di,Cx)=e g1,g2)rqx(0)+hiriP(0)qx(0) e(g1,g2)rihiqx(0)λkP(uk)e(g1,g2)$r^i h^i$qx(0)j=1λjP(uj) = e(g1,g2)rqx(0)+$h^i r^i$P(0)qx(0) e(g1,g2)rihiqx(0)(j=1λjP(uj)+λkP(uk))=e(g1,g2) qx(0)+hiriP(0)qx(0) e(g1,g2)rihiq$_x$(0)P(0),k∈ {1,2,...,t}= e(g1,g2)rq^x(0).

B. ALTERNATE DYNAMIC ATTRIBUTE REVOCATION

Here we are describing how to revoke one or more attributes from a given secret key. This is useful since often the data owner may want to merely revoke a few attributes from her contacts instead of the whole key. For instance, user A might want to remove friend attribute from B, but B still remains in her colleague group.

Intuition:
The idea is basically the same as complete key revocation. The master key contains one polynomial Pi of degree ti for each possible attribute i that the data owner can assign. Any attribute can be introduced later by introducing a new polynomial in the MK. Pi (0) is used to blind the corresponding attribute in the secret keys. Each user u also gets a random share Pi (u) of Pi (0) in her key. The proxy key consists of ti such shares for each attribute in the policy used in the ciphertext. Whenever some attribute is revoked from some user, that share becomes a part of the proxy key, and hence the converted ciphertext. Therefore, the revoked user does not have enough points, i.e. (t_i +1) points for that specific attribute to unblind her key and the ciphertext and decrypt it. However, non-revoked users can always combine their secret key with the ciphertext and hence decrypt it. As before, when no attribute is revoked, the proxy key consists of t_i random points for each attribute i.

- SETUP: The KA generates one polynomial P_y randomly over Zp for each attribute y ∈ Y where Y is an initial set of attributes in the system, and sets Py (0) as the secret to be used to revoke the attribute. To revoke an attribute from t users at a time, the degree of the polynomials is chosen to be t. New attributes can be introduced later by randomly generating polynomials for them. Finally, she randomly chooses α, β ∈ Zp.

PK = G1,G2,g1,g2,h = g^β 1,e(g1,g2)α MK = β,gα,2,∀y∈Y: Py

- KEYGEN (MK, S): The components of the secret key are similar as before except that the polynomial in each is specific to the attribute represented by the component. Again, without loss of generality, we assume user u_k receives this key.

SK = (D,∀j ∈ S :Dj,Dj,D j), where D = g(α+r)/β 2 ,Dj= g_r 2· H(j)r_jPj(0)= gr+hjrjPj(0) 2,Dj= grj 1, Dj= (Dj)Pj(u^k)= grjPj(u^k)

- ENCRYPT (PK, M, τ): Encryption is similar as incomplete key revocation.
- PROXYREKEY (PK, MK, ∀y ∈ Y: RL_y): To revoke an attribute y ∈ Y from t contacts, the KA creates a t-sized list RL_y = { u_i }, i ∈ {1... t} of revoked users for that attribute, and evaluates Py (u_i) using MK. In case of no or less than t revocations, she generates random x, Py(x) to make RL_y of length t. The set of users from whom different attributes are revoked, may or may not overlap. Without loss of generality we assume that the sets of revoked users don't overlap. The proxy key PXK is constructed as follows: PXK = ∀y ∈ Y, ∀ u_i ∈ RLy: u_i, Py (u_i)).
- Convert(PXK, ∀y ∈ Y : $_{Cy}$): The proxy uses its key PXK to convert the attribute relevant components $_{Cy}$ receive from user u_k to $_{Cy}$ as follows:

$^{\lambda}{}_{y}{}^{i}$= u_k − u_i,j =I uj (uj− ui), ∀ui,uj∈ RLy, u^k / ∈ RLy, RLy∈ PXK

∀y ∈ Y: Cy= (Cy) t
i=1λyiPy (ui) =ghyqy (0) I 1λyiPy (ui) 2∀y ∈ Y the proxy also calculates and gives λyk, to u^k.

- Decrypt(CT, SK): For each leaf node x where i = attr(x), if i ∈ S (S is the set of attributes for which SK is issued), and i is not revoked from uk then,

Decrypt Node (CT, SK, x)
=e (Cx, Di)/e (Di, Cx) λik e (Di, Cx)
=e(g1,g2)rqx(0)+$h_i r_i P_i$(0)qx(0)/e(g1,g2) $r_i$hiqx(0)λi kP_i (uk)e(g1,g2) $r_i h_i$qx(0)t
j=1λi jP_i (u_j)
= e (g1, g2) rqx (0)

Otherwise Decrypt Node returns ⊥. The rest of the decryption is as before. In summary, if an attribute i is revoked from user u, he cannot do pairing on Cx and Di. He can continue to use components related to his other unrevoked attributes. Therefore, some of his attributes are revoked whereas some continue to be active.

VIII. PERFORMANCE ANALYSIS

We are providing some information on the performance evaluation of proposed work, and compare it with CP-ABE. This provides security by preventing key and ciphertext components exchange. The results are shown in Figure 4.

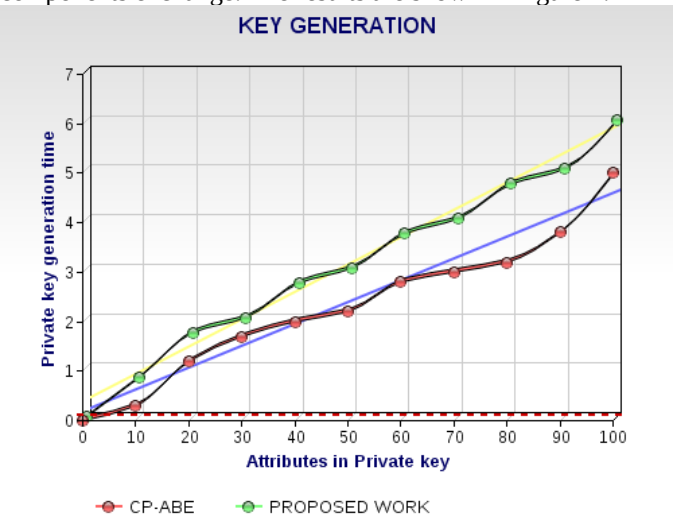


Fig: 4. a) Key Generation

All Rights Reserved © 2013 IJSETR

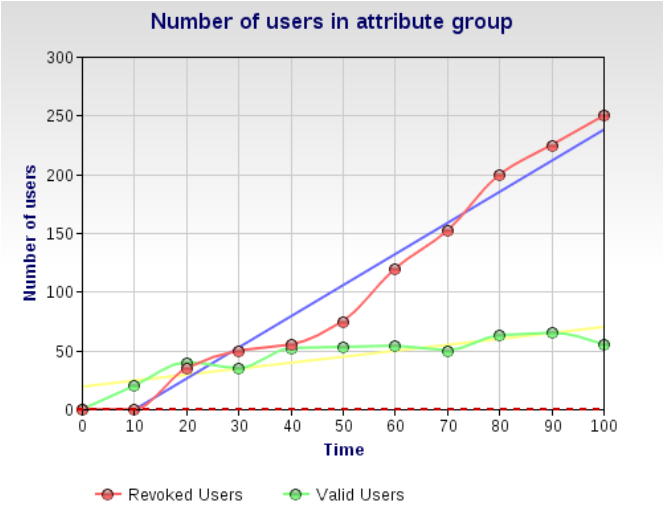


Fig: 4. b) the number of users in an attribute groups

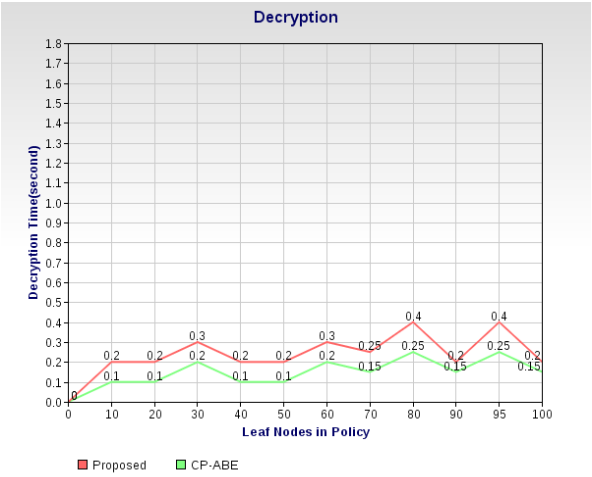


Fig: 4. c) Decryption

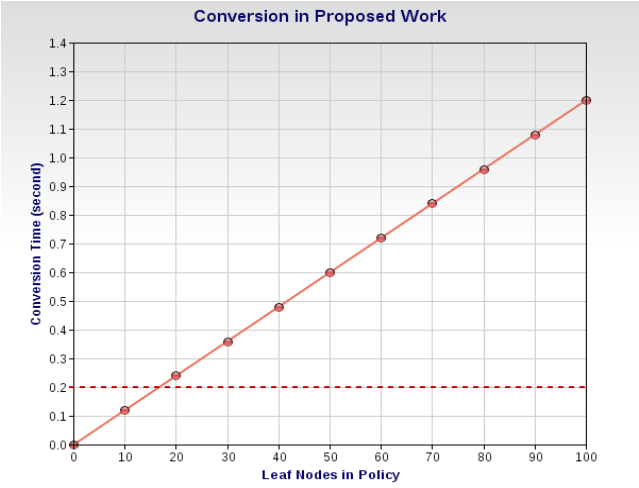


Fig: 4. d) Conversion in Proposed works

Key generation time is linear with number of attributes both in CP-ABE and proposed work. Since it does an extra exponentiation, and generates an extra component for each attribute in and proposed work, the result is justified. To test cryptography, we randomly generated 10 different policies for each of the desired number of leaves (1, 5, 10 . . . 100). Encryption (not shown in the Figure) is also linear with respect to the number of leaf nodes in the policy. Since no change was made to CP-ABE encryption, both take the same amount of time. Decryption (Figure 4c shown with 95% confidence interval) depends on the policy used in encryption and the attributes involved. We generated a decryption key with 100 attributes that satisfies all the policies used. The lines marked proposed work and CP-ABE show the results when an optimization implemented in CP-ABE was used to ensure that the minimum number of leaves were used in DecryptNode. The required time is still below 1 second though recursive DecryptNode was used. We expect better results with further optimization.

Proposed work involves two extra costs before decryption: rekeying the proxy and converting the ciphertext components specific to the leaves in the policy. We perform an optimization by allowing the proxy to pre-calculate a portion of the λ's in PROXYREKEY. The re-keying results (not shown in Figure) show that, even for 500 revoked users, the time required is about 1.4 seconds. This should be compared with the time required to rekey the rest of a group, i.e., generate a new key for everyone, when even one person in the group is revoked.

The time to compute the exponentiations dominates the time to do t multiplications; hence the results are essentially linear in the number of leaf nodes. Figure 4d shows the conversion time for 500 revoked users. We expect the proxy to be more powerful in terms of computing, and hence rekeying, and conversion should be faster in practice. A user requesting decryption only faces the conversion time shown in Figure 4c along with the decryption time mentioned earlier.

A. Component Size and Communication overhead

Component	Proposed work(bytes)	CP-ABE(bytes)
Public key	1316	1316
Master key	152+(t+1)24	148
Private key	128+(a+212)n	128+(a+168)n
Cipher text	168+8i+(176+a)l	168+8i+(176+a)l
Proxy key	24t	NA

Table 1 shows the sizes of the components involved in the system, calculated based on group members. Elements consist of t is the degree of polynomial, n is the number of attributes in private key, and a is the string length of an attribute. To represent bytes G0, G1, G2, and Zp require 44, 124, 124, and 24 bytes respectively. Public Key includes a string describing the pairing used (980 bytes). Users communicate with the proxy for conversion by sending Cy, and receiving Cy. These are represented using elements from G1. This requires 124 bytes to represent (120 for the actual data and 4 for the variable size). Hence, conversion of a ciphertext with l leaf nodes in the policy will need to transfer 124l bytes each way. The user also sends u_k, and receives back. These are represented using Zp which requires 24 bytes. Therefore, we conclude that the communication overhead is reasonable for OSN users.

All Rights Reserved © 2013 IJSETR

IX. CONCLUSION

To achieves more secure and fine-grained data access control in the data sharing system. We demonstrated that the proposed scheme is efficient and scalable to securely manage user data in the data sharing system also we achieved revocation scheme by introducing a semi-trusted proxy, leveraging ideas from a group communication scheme, and combining it with ABE. Data privacy and confidentiality in the data sharing system against any system managers as well as adversarial outsiders without corresponding (enough) credentials. We implemented the scheme and compared it with Bethencourt et al.'s CP-ABE. Our results show that proposed work is scalable in terms of computation and communication. In the future, it would be interesting to consider attribute-based encryption systems by applying advanced cryptosystem for data sharing. In future, we are expecting to encrypt multimedia files, Solve the performance degradation of fully distributed approach, avoid key expired time, we can use multi Data Storing Centre, Proxy servers to update user secret key without disclosing user attribute information.

REFERENCES

[1] J. Hur, "Improving Security and Efficiency in Attribute-Based Data Sharing," IEEE TKDE, 2011.
[2] J. Bethencourt, A. Sahai, B. Waters, "Ciphertext-Policy Attribute-Based Encryption," Proc. IEEE Symposium on Security and Privacy 2007, pp. 321–334, 2007.
[3] A. Sahai and B. Waters. Fuzzy identity-based encryption. In Eurocrypt 2005, LNCS 3494, pages 457–473. Springer-Verlag, 2005.
[4] A. Sahai, B. Waters, "Fuzzy Identity-Based Encryption," Proc. Eurocrypt 2005, pp. 457–473, 2005.
[5] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters. Secure attribute-based systems. In CCS '06: Proceedings of the 13th ACM conference on Computer and communications security, pages 99–112, New York, NY, USA, 2006. ACM.
[6] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In CCS '06: Proceedings of the 13th ACM conference on Computer and communications security, pages 89–98, New York, NY, USA, 2006. ACMS.
[7] Alexandra Boldyreva, Vipul Goyal, Virendra Kumar, ―Identity-based encryption with efficient revocation‖ , Proceedings of the 15th ACM conference on Computer and communications security, ISBN: 978-1-59593-810-7, pp 417-426, 2008.
[8] Shucheng Yu, Cong Wang, Kui Ren, Wenjing Lou, ―"Attribute based data sharing with attribute revocation", Proceedings of the 5th ACM Symposium on Information, ISBN: 978-1-60558-936-7, pp 261-270, 2010
[9] Ling Cheung, Calvin Newport," Provably secure cipher text policy ABE", Proceedings of the 14th ACM conference on Computer and communications security, ISBN: 978-1-59593-703-2, pp 456-465, 2007
[10] S.S.M. Chow, "Removing Escrow from Identity-Based Encryption," Proc. PKC 2009, LNCS 5443, pp. 256–276, 2009.
[11] M. Chase, S.S.M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," Proc. ACM Conference on Computer and Communications Security, pp. 121–130, 2009
[12] S. Yu, C. Wang, K. Ren, W. Lou, "Attribute Based Data Sharing with Attribute Revocation," Proc. ASIACCS '10, 2010.
[13] L. Cheung, C. Newport, "Provably Secure Ciphertext Policy ABE," ACM Conference on Computer and Communications Security, pp. 456–465, 2007.
[14] V. Goyal, A. Jain, O. Pandey, A. Sahai,"Bounded Ciphertext Policy Attribute-Based Encryption," Proc. ICALP, pp. 579–591, 2008
[15] X. Liang, Z. Cao, H. Lin, D. Xing, "Provably Secure and Efficient Bounded Ciphertext Policy Attribute Based Encryption,"Proc. ASIACCS, pp. 343–352, 2009
[16] A. Boldyreva, V. Goyal, V. Kumar, "Identity-Based Encryption with Efficient Revocation," Proc. ACM Conference on Computer and Communications Security 2008, pp. 417–426, 2008.
[17] S. Rafaeli, D. Hutchison, "A Survey of Key Management for Secure Group Communications," ACM Computing Surveys, vol. 35, no 3, pp. 309–329, 2003
[18] A. Lewko, A. Sahai, B. Waters, "Revocation Systems with Very Small Private Keys," Proc. IEEE Symposium on Security and Privacy 2010, pp. 273–285, 2010
[19] R. Ostrovsky, A. Sahai, B. Waters, "Attribute-Based Encryption with Non-Monotonic Access Structures," Proc. ACM Conference on Computer and Communications Security 2007, pp. 195– 203, 2007

AUTHOR BIOGRAPHY

Lekshmy Aravind was born in Trivandrum in 1987. She graduated in B-Tech Computer Science and engineering from Lourdes Matha College of Science & Technology, Kuttichal under University of Kerala in 2009. Currently she is pursuing M.E Computer Science and Engineering in Noorul Islam University, Thuckalay. Her research interest includes Wireless Communication, Data mining, Image processing. She was a member in the Computer Society of India (CSI) in 2005-2009

M.Victor Jose received the BE degree in Computer Science and Engineering from Manonmaniam Sundaranar University, India and ME degree in Computer Science and Engineering from Madurai Kamaraj University, Tamil Nadu, India. He is currently working towards the PhD degree at the Department of Computer Science and Engineering, Anna University, India. His research interests include Grid computing, Network security, Data base security, and Multimedia wireless communications. Email_id: mvictorjose@yahoo.com.

All Rights Reserved © 2013 IJSETR