Attribute-Based Encryption: An Efficient Way to Secure Cloud Storage

P. R. Jaiswal[*], A. W. Rohankar[#]

[*]Sinhgad College of Engg., I.T. Department, Pune University
[#]Sinhgad College of Engg, I.T. Department, Pune University

[*]pavan.jaiswal85@gmail.com
[#]rohankar@sinhgad.edu

Abstract - Cloud computing is an emerging computing paradigm that enables users to remotely store their data on a cloud so as to enjoy scalable services on-demand. It has been found that maintaining the privacy of data from an unauthorized user is really a major challenge. This is a major concern as such data is outsourced to keep storage at third parties - say cloud service providers. It is very much essential to use promising method, which performs encryption on such data before it actually get outsource to the third parties. Besides this, other key issues like – efficient user revocation, scalability in key management; must have to get address while achieving fine grained and scalable access control to cloud storage. This paper ensures effective dealing with above mentioned security concerns by making use of Attribute-Based Encryption technique. Besides this paper presents how cloud storage security is achieved by designing efficient security framework, which is capable of dealing with multiple data owner scenario, handling multiple security domains and dynamic modifications of access policies.

Keywords - cloud computing, attribute-based encryption, data privacy, fine-grained access control, user revocation, trusted authority.

1. INTRODUCTION

In the recent years, cloud computing paradigm has reached high. Many people now prefer to use this feature for computation, processing and storage of their data. Building and maintaining specialized data centers is a quite costly. So usually data storage is kept with third party cloud service provider which manages this cost. In such scenario, maintaining the privacy of user's data from unauthorized users is not an easy task. Another main concern is how user could actually control sharing of data which is kept on semi trusted server. Here the need comes for having a mechanism of fine-grained data access control that could work efficiently with semi trusted server. A promising approach would be to encrypt the data before it is outsourced to the semi trusted server. The best way to do so is to let allow data owner to decide with encryption and access mechanism. Access to original data should only be given to those users who hold the proper decryption key. Besides this, data owner should have right to grant and revoke access privileges whenever necessary [3].

We have chosen ABE as most impressive encryption technique in this paper. ABE allows access policies to be expressed based on attributes. Number of attributes involved decides the complexity per encryption, decryption & key generation, and it is linear. However on demand revocation, dynamic access policies and key management scalability are the important issues which remain open when we integrate ABE with large scale application. To this end we propose ABE-based framework for securing and sharing data in cloud environment under the multiple data owner scenario.

1.1 Motivation

This is fact that in un-trusted storage data servers are neither can be relied to enforce data access policies nor allowed to learn contents of sensitive data. So following best practice, data owner encrypts the data before outsourcing it on storage server. This helps to preserve data confidentiality. The user who holds decryption key is granted to get access to encrypted data. There are few challenges pertained to this type of access control mechanism. We summarize them as below:

Fine-grained access control vs. scalability: Fine-grained access control is required whenever sensitive data needs to be disclosed. Traditionally access control is achieved by ACL-based access control [20], capability-based access control [21] and role-based access control [22]. Making use of ACL based and capability-based cryptographic method leads towards the scalability issue. When ACL's are used as cryptographic method, complexity for each data object in terms of data encryption operation or cipher text size is linear to the number of system users. This results in less scalable system. Capability-based access control do have the same system scalability issue. In role-based access control, remembering the authorized users list is not expected as access to data is granted on the basis of user's role [22]. Several recent work [8],[9],[16],[17] in the areas of "access control of outsourced data" and "shared cryptographic file systems" do address the data access control issue with conventional symmetric and asymmetric cryptography.

To overcome this problem, ABE [2,9,15] have been recently invented which is public-key cryptography. ABE has potential of enforcing access policies to large scale systems. Basic functionalities like data encryption/decryption, collusion resistance have been focused in ABE's existing construction [2],[9],[15]. Several security issues needs to be addressed before ABE is applied in practical systems.

User Dynamics: In real time systems, user may join or leave system at any time. So granting and revoking use access privileges, there should be efficient user management in place. In cryptography, user key revocation is always a matter of concern. User key revocation in ABE is found challenge issue. Efficient revocation scheme for IBE that also applies to ABE is proposed in [4].

User Accountability: In cryptographic-based data access control, user who hold proper decryption key, is allowed to access encrypted data. But sometime this leads to sharing of decryption key to unauthorized user by the authorized user. Copyright sensitive application can be harmed more by such attack. A novel solution is needed to avoid key abuse attack

Privacy Preservation: It is good practice to disclose as less user privacy information to un-trusted servers. Even data owner would always prefer not to disclose his access policy information to server. It is necessary to provide new construction to ABE than the existing one [2],[9],[15] that takes care of privacy preservation policy.

Efficiency: Efficiency of ABE is application dependent. It varies depending on kind of application. Some application expects it by reducing cost of operations like bilinear pairings on encryptor and/or decryptor, where as other applications expects it by delegating computation-intensive operations to more powerful devices.

1.2 Contributions

In this paper we make several major contributions as below.

1.2.1 Security Enhancement to ABE

ABE is known as Public Key Cryptography (PKC) primitive and useful in cryptography-based fine-grained data access control for un-trusted storage. It is very important to address several security issues discussed in section 1.1 before we use ABE in real system. In this paper we propose several security enhancements to address these issues.

User Revocation: To facilitate user revocation, we propose a novel scheme in which revocation is performed by both data owner and TA. The proposed scheme allows data owner to delegate user revocation responsibility to TA in simplest manner.

Encryption with Hidden Policy: In CP-ABE construction [2],[7] user decryption is facilitated by attaching access policy in plaintext to data ciphertext. This discloses the data owner's access policy to the user harming privacy concern of data. In order to make better privacy protection, we have hided access policy information from user.

1.2.2 Secure Data Sharing Scheme for Cloud Computing

Cloud computing is referred as promising next-generation IT architecture. In this cloud providers and cloud users belongs to different trust domains. So there is necessity of secure user-enforced access control mechanism before an outsourcing of confidential data on un-trusted storage. In this paper we use ABE to propose cryptographic-based data access control mechanism in which we enable data owner to take full control over data access. We provide better scalability as compared to previous work [8],[11],[16],[17] because most system operations in our scheme are linear to number of attribute rather than data files or users. Moreover to reduce computation load on data owner, we delegates offload computation intensive tasks to TA.

1.3 Roadmap

In section 2 we focus on technical details of ABE along with its types like: KP-ABE, CP-ABE. Section 3 focuses on usage of ABE to revoke user which in details talks about algorithm used for this purpose. Section 4 concentrates on ABE for preserving privacy of data owner's data and access structure for each user. Section 5 helps to build framework for secure data sharing in cloud computing using ABE along with its security and performance analysis. Section 6 concludes this paper.

2. TECHNICAL PREPARATION

2.1 Attribute-Based Encryption

Definition: For cryptographically enforced access control, Sahai and Waters [15] first introduced Public-Key Cryptography (PKC) ABE. In ABE user private key and ciphertext are associated with set of attributes. Decryption of ciphertext by user is possible only when at least threshold number of attributes overlap between user private key and ciphertext. Different from traditional PKC IBE [5], ABE is intended for one - many encryption means, ciphertext is not necessarily encrypted to one particular user. In Sahai-Waters threshold value is not expressive. To get more general access control, Goyal et al. [9] proposed variant of ABE that is Key- Policy-ABE (KP-ABE). In this variant, ciphertext is associated with set of attributes and each user secret key is embedded in access structure. Bacause of this user can decrypt a ciphertext only if ciphertext attribute satisfy access structure embedded in his private key. In [9] Goyal et al introduced another variant of ABE: Ciphertext-Policy ABE (CP-ABE). CP-ABE is reverse way of KP-ABE. This means ciphertext is associated with access structure and user secret key is embedded in set of attributes. Formally KP-ABE and CP-ABE can be defined as below:

. ***Key-Policy Attribute-Based Encryption***: KP-ABE consists of following four algorithms:

Setup: This algorithm takes as an input security parameter k, and returns public key PK which is used for encryption by sender and secret master key MK which is used by TA to generate user secret keys.

Encryption: This algorithm takes as inputs message M, public key PK, set of attributes γ and outputs ciphertext E.

Key Generation: This algorithm takes as input access structure T and MK. It outputs secret key SK that enables users to decrypt to a message encrypted under set of attributes γ if and only if γ matches T.

Decryption: It takes as input SK for T & E which was encrypted under γ and it outputs M if and only if γ satisfies user's structure T.

. ***Ciphertext-Policy Attribute-Based Encryption***: CP-ABE also consists of four algorithms:

Setup: This algorithm takes as an input security parameter k, and returns public key PK which is used for encryption by sender and secret master key MK which is used by TA to generate user secret keys.

Encrypt: This algorithm takes as input PK, M & T; and outputs ciphertext CT.

Key Generation: It takes as an input γ associated with user & MK. It outputs SK used to decrypt message encrypted under T if and only if γ matches T.

Decrypt: It takes as input CT`, SK for γ. It outputs M if and only if γ satisfies access structure associated with CT.

Beside primitive functionalities for ABE, many works have been proposed to provide better privacy protection for ABE which includes ABE with user accountability [6],[12],[28], ABE with attribute hierarchy [110], CP-ABE with hidden policy [10],[13],[19],[25] and many more.

In this paper we consider a case multiple-writer-and-multiple-reader in un-trusted storage. Who should be writer and/or reader is solely be decided by data owner and TA.

3. USER REVOCATION FOR ABE

User revocation is challenging issue in ABE as each attribute is shared by many users. It might be possible that revoking single user may affect other user who share the same attribute. In this paper we focus on practical application scenarios like data storage and sharing as shown in fig 1. in which proxy servers are always available to provide data services. As mentioned in [8] these servers are assumed curious-but-honest instead of totally un-trusted. It means they will execute the task given honestly, but they do have incentive to learn as much as encrypted data they can. Keeping this assumption in mind, solution in terms of proxy re-encryption technique [3] with ABE is given. This solution allows data owner to delegate user revocation task to TA without even leaking confidential information to them. On each revocation event, TA generates proxy re-encryption keys and transmits them to server. Later on server update secret keys of all other except user to be revoked. This helps to reduce data owner load.

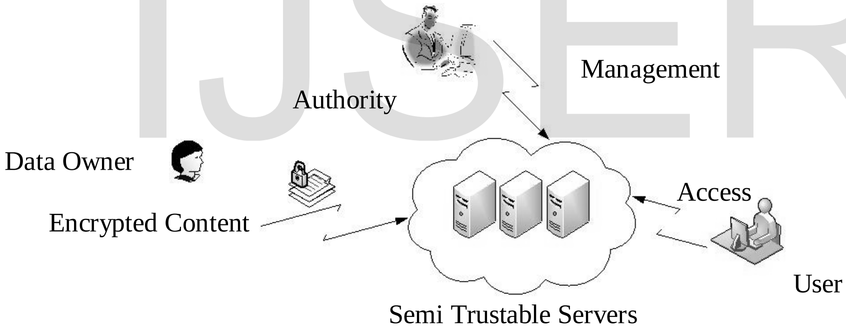

Figure 1. An example application scenario of data sharing

Existing work [2],[14] suggested one approach in which user secret key can be associated with expiration time attributes. But this approach just revoked users at predefined time but cannot revoke user attributes on the fly. In our proposed scheme at any time, data owner and/or TA is freely able to revoke any attribute of user. Our technique, particularly designed for data storage can be used for both KP-ABE & CP-ABE.

3.1 Algorithm Definition

We use following algorithms to provide solution to user revocation problem. Setup, Enc, KeyGen, ReKeyGen, ReEnc, ReKey, Dec. Out of these Setup, KeyGen & ReKeyGen are performed by TA, while ReEnc, ReKey are by server. Encryptor and Decryptor calls Enc and Dec respectively. The operations of Setup, Enc, KeyGen, Dec are same as of CP-ABE scheme [2],[7]. TA defines ReKeyGen to generate proxy re-keys, whereas re-encrypting data and updating user secret key is performed by server that makes use of ReEnc & ReKey respectively. The algorithms are defined as below:

Setup*(1^λ)*: It takes input security parameter 1^λ and outputs system master key MK and public key PK.

Enc*(M,AS,PK)*: It takes as input message M, access structure AS & PK; and outputs ciphertext CT.

KeyGen*(MK,S)*: It takes as input MK & set of attribute S that describes the key. It returns users secret key SK in the form of (S,D, $\bar{D}$= {Di, Fi}iεs)

ReKeyGen*(γ, MK)*: It takes input attribute set γ and MK and outputs new master key MK′, new public key PK′ and set of proxy rekeys rk for all attributes in the attribute universe U.

ReEnc*(CT,rk,β)*: It takes input CT, rk and a set of attributes β which includes all the attributes in *CT*'s access structure with proxy re-key not being 1 in *rk*. It outputs a re-encrypted ciphertext CT′ with the same access structure as *CT*.

ReKey($\bar{D}$*,rk,θ*): It takes as input $\bar{D}$ of SK, rk as set of attributes θ which includes all the attributes in SK with proxy re-key not being 1 in rk. It returns updates user secret key components $\bar{D}'$.

Dec(*CT,PK,SK*): It takes as input CT, PK, SK same version of CT. It returns M if attribute set of SK satisfies the ciphertext access structure. Otherwise it returns ⊥ with an overwhelming probability.

3.2 Basic Construction

The basic idea of construction is to combine proxy re-encryption technique with CP-ABE. Rather than building new CP-ABE scheme, we enhanced existing construction by extending it with capabilities of proxy re-encryption of ciphertext and proxy update of secret key. This construction is based on Cheung et al construction of CP-ABE [7].

Attribute & Access Structure: In this construction, attributes and attribute universe are represented by their index values and U={1,2,..n} for certain natural number n, respectively. We consider three occurrences of each attribute: positive, negative and "don't care". Single AND gate is considered in access structure. "Don't care" occurrence is considered when attribute do not appears in AND gate.

4. PRIVACY PRESERVING ABE

In existing construction of ABE [2],[7],[9],[23],[24], it facilitates decryption by attaching plain text of data attributes (in KP-ABE) or data access structures (in CP-ABE) to ciphertext. This construction reveals data access policies to un-trusted server. So in order to provide better security protection, we provide CP-ABE constructions under given below security model. In our scheme, from data owner point of view, irrespective of whether user is authorized or unauthorized, we have kept access structure hidden to un-trusted server & user.

4.1 Construction Under XDH Assumption

This section focus on construction under External Diffie-Hellman (XDH) assumption which holds for some MNT elliptic curves [26],[27].

Description: We constructed privacy-preserving CP-ABE by enhancing Cheung's construction [7]. Notations followed are same as in [7]. The set of attributes are defined as N:= 1,..n for some natural number n. Literals are referred by attribute i and their negation ¬ i. Let set of attributes needed for decryption are denoted by I. Scheme considers access structures that consists of single AND gate whose inputs are literals, denoted by $\wedge_{i\varepsilon I}$ _I where every _i is literal (ie i or ¬ i).

SETUP: Here bilinear groups G_1 and G_2 of prime order p with generator g1 and g2 respectively. A bilinear map e : $G_1 \times G_2 \rightarrow G_T$ is defined on them. Next it chooses random exponents y, $t_{1,..,}t_{2n} \in Z_p$ The public key published as:
PK = (e, g_1, g_2, Y, T_1, … T_{2n}).
where Y = $e(g_1, g_2)^{y,}$ ∀i ∈ Z_{2n} : Ti = $g^{t}{}_{i}{}^{1}$. The MK = (y, t_1,..t_{2n})

In our construction each attribute has two occurrences: positive and negative. "don't care" is discarded keeping it key element in original construction.

ENCRYPT: Given a message M ∈ G_T and AND gate W = $\wedge_{i\in I}$ _I , the ciphertext is output as CT = (~C, ∧C, {C_{i0}, C_{i1} | i ∈ N }), where ~C = M.$Y^{s,}$ ∧C = g^s , and s is a random number in Zp.

For each i ∈ I, $C_{i,0}$ and $C_{i,1}$ are computed as follows.

. If _i = i, $C_{i,0} = T^s{}_i$, $C_{i,1} = T^x{}_{n+i}$

. If _i = ¬ i, $C_{i,0} = T^x{}_i$, $C_{i,1} = T^s{}_{n+i}$

x is random number in Zp.

For each i ∉ I , $C_{i,0} = T^s{}_i$, $C_{i,1} = T^s{}_{n+i}$

KEYGEN: Let S denote input attribute set. Every i ∉S is considered a negative attribute. The SK = (∧D, {D_i| i ∈N}), where ∧D = $g^{y\text{-}r}{}_2$, r = $\Sigma^n{}_{i=1}$ r_i , r_i is randomly selected from Zp. For each i ∈ N, Di = $g^{ri/ti}{}_2$ if i ∈ S; otherwise, Di = $g^{ri\,/\,tn+i}{}_2$

DECRYPT: Suppose CT = (~C, ∧C, {C_{i0}, C_{i1} | i ∈ N }) . Let SK = (∧D, {D_i| i ∈N}). For each i ∈ N, if users attribute is positive then

Fi = e($C_{i,0,}$ D_i) = e($g^{ti.s}{}_1$, $g^{ri/ti}{}_2$ = e (g_1, g_2)$^{ri.s}$)

If users attribute is negative, then

Fi = e($C_{i,1,}$ D_i) = e($g^{tn+i.s}{}_1$, $g^{ri/tn+i}{}_2$ = e (g_1, g_2)$^{ri.s}$)

Decrypt finishes as follows: M = ~c / Y^s = ~c / e(g_1, g_2)$^{y.s}$), where

e(g_1, g_2)$^{y.s}$ = e($g^s{}_1$, $g^{y\text{-}r}{}_2$) . e(g_1, g_2)$^{r.s}$ = e(∧C, ∧D) . $\prod^n{}_{i=1}$ $F_{i.}$

Above equations demonstrates how any intended user decrypt the ciphertext. The user who is not intended recipient, there at-least one attribute for which the user gets Fi with the form e(g_1, g_2)$^{ri.x}$. Therefore user cannot calculate e (g_1, g_2)$^{y.s}$ as shown above.

4.2 Scheme Analysis

Ciphertext in this discussion does not include access policy. In DECRYPT, to decrypt ciphertext, user uses all his attributes. The user can not decrypt ciphertext if i^{th} attribute is positive while C_{i0} has the form $T^x{}_{i.}$ User is not able to know which attributes are desired by the encryptor as user cannot distinguish between $T^x{}_i$ and $T^s{}_i$ according to XDH assumption. It means user cannot drive any information related to the access policy. This is the reason where an intended user only knows if he can decrypt the ciphertext despite of any idea that which attribute has granted him access. Therefore access policy is hidden to all users.

5. SECURE DATA SHARING WITH ABE IN CLOUD COMPUTING

As mentioned earlier, Cloud computing is referred as promising next-generation IT architecture. In this cloud providers and cloud users belongs to different trust domains. So there is necessity of secure user-enforced access control mechanism before an outsourcing of confidential data on un-trusted storage. Besides this, issues like fine-grained access control with scalability, user dynamics, system efficiency, etc needs to get resolve. It is quite possible that end user may use any low-end device such as mobile phone which has less computation power. This makes it mandatory that data access mechanism should be designed in such way that load addressed on both data owner and consumer must be affordable to these low-end devices.

By considering these many issues, we propose cryptographic-based data access control mechanism. We addressed issue of user revocation by the said method in above section. We reduce computation load on data owner by using technique of lazy re-encryption [11]. We do provide better scalability as complexity of most system operations in our scheme is linear to number of attributes and not to number of users.

5.1 Models and Assumption

5.1.1 System Models

As mentioned in [18], we do assume that parties involved in our system are: cloud users, cloud server, trusted authority, etc. Data owner keeps his data in encrypted form. Encrypted data can be used by him personally or he can make it available for sharing. It is quite possible that cloud users come online just as and when required. So there is TA who works on behalf of data owner to delegate the access rights to sharer. We assume that users in personal domain (PSD) is decided by the data owner where as users in public domain (PUD) is decided by TA.

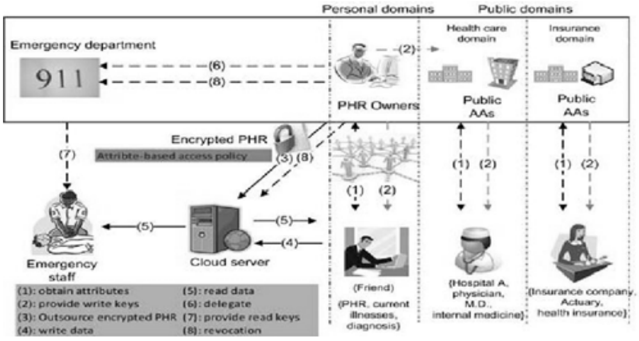

Figure 2. An example case in healthcare scenario [1]

5.2.2 Scheme Description

In this work, we focused operations like: system setup, user grant, user revoke, managing access structure and so on. Below given are notations used in scheme description.

TABLE 1 Notation Used

Notation	Description
PK, MK	system public key and master key
t_i	master key component for attribute i
Ti	public key component for attribute i
SK	user secret key
sk_i	user secret key component for attribute i
Ei	ciphertext component for attribute i
I	attribute set assigned to data file
DEK	symmetric data encryption key of data file
P	user access structure
Lp	set of attributes attached to leaf nodes of P
UL	the system user list

5.1.2 Security Models

In this work, we assumed that other than secret information, cloud servers are more interested in get to know about user access privilege information and data stored on them. We assume that communication channel between cloud server and client is secured by means of some existing protocols, say SSL. Irrespective of this, unauthorized user will try to access data files and will perform some malicious job. In addition to this, each party is loaded with public-private key pair.

5.1.3 Design Goals

Our main design goal is to unable cloud user to gain fine-grained access control to his data kept at cloud server. Fine-grained access control here means data owner should be able to enforce grant, update & revoke to access structure on each user. In addition to this, scheme should support user dynamics, user accountability. Another goal is that computation load should be efficiently handled.

5.2 Our Proposed Scheme

5.2.1 Main Idea

In this work we combine and utilize cryptographic techniques: KP-ABE & Proxy Re-Encryption (PRE) to secure, scalable and fine-grained data sharing in cloud. We provided each user with data attributes and access structure. As discussed earlier, KP-ABE plays important role in fine-grained access control. Proposed scheme takes care of user revocation. Data owner can himself perform this or delegate this task to TA.

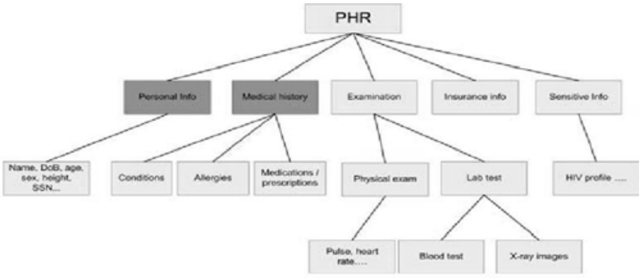

Figure 3. Attribute Hierarchy

Algorithms used in this scheme are: ASetup, AEncrypt, AKeyGen, ADecrypt, AUpdateAtt and AUpdateSK. First four algorithms are similar to Setup, Encryption, KeyGeneration and Decryption of standard KP-ABE respectively. In this section we focus on AUpdateAtt algorithm. Below fig 5 shows pseudo code of it.

```
AUpdateAtt(I, MK)
  // assume current version of attribute i is k-1
randomly pick $t'_i \xleftarrow{R} Z_p$
compute $T'_i \leftarrow g^{t'_I}$ and $rk_i^{(k)}$
```

```
output $t'_i$ , $T'_I$ and $rk_i^{(k)}$
```

Figure 5. Pseudo Code

AUpdateAtt algorithm updates an attribute. For this it redefines system master key and public key component. It outputs proxy re-encryption key between the old version of new version of the attribute.

AUpdateSK translates secret key component of attribute i in the user secret key SK from old version to its latest version.

5.3 Analysis of Proposed Scheme

5.3.1 Security Analysis

We analyze security properties starting with the following:

Fine-Grained Access Control: This scheme allows data owner to define and enforce access structure for each user. Basically access structure represents any desired data file set. In this we define access structure as a logical formula.

Data confidentiality: We analyze data confidentiality by comparing our scheme with intuitive scheme in which encryption on data files is performed using DEKs and DEKs are encrypted directly using standard KP-ABE. As standard KP-ABE is secure under attribute-based selective model, [9] intuitive scheme is also secure under same model. Our goal is to show security of our scheme which is same as intuitive scheme's security.

5.3.2 Performance Analysis

Here we evaluate performance in terms of computation overhead introduced by operations performed and ciphertext size.

Computation Complexity: We analyze computation complexity for following operations.

System Setup: In this data owner generates PK and MK, and defines underlying bilinear groups. In case of KP-ABE, computation overhead is introduced by N group multiplication operation on G_1.

New File: *Creation* Main computation overhead of this operation is using symmetric DEK for encryption of data file as well as using KP-ABE for encryption of DEK. Size of data file decides complexity of former. The computation overhead of later consists of | *I* | multiplication operations on G_1 and 1 multiplication operation on G_2. In this *I* denote attribute set *I* of data file. All these operations are considered for data owner.

New User Grant: In this operation data owner and cloud server is involved. In this computation overhead for data owner is for generation of user secret key and encryption of user secret key with user's pubic key. The former account for | *L* | multiplication operations on G_1, *L* denotes set of leaf node of access tree. The later account for one PKC operation.

User Revocation: This operation holds two stages. The second stage can be amortized as the file access operation. First stage occurs between data owner and cloud server.

Operation	**Complexity**
File Creation	O (| *I* |)
User Grant	O (| *L* |)
User Revocation	O (| *N* |)

Figure 6. Complexity of proposed scheme

6. CONCLUSION

In this paper we addressed an important issue of secure data sharing on un-trusted storage. We revealed challenges to this problem and exploited PKC ABE to enjoy fine-grained access control on outsourced data. In order to gain full fledged cryptographic basis for secure data sharing on un-trusted storage we proposed two schemes using ABE. First scheme is given for efficient user revocation using ABE and second scheme is provided to gain better privacy preservation in terms of access structure protection using ABE. With these enhanced schemes we presented solution to secure data sharing in cloud computing for health care application.

REFERENCES

[1] M. Li, S. Yu, K. Ren, and W. Lou, "*Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute Based Encryption,*" in IEEE Transaction on Parallel and Distributed Systems, pp. 99. 2012.

[2] J. Bethencourt, A. Sahai, and B. Waters. "*Ciphertext-Policy Attribute-Based Encryption*". In *Proc. of SP'07*, Washington, DC, USA, 2007.

[3] M. Blaze, G. Bleumer, and M. Strauss. "*Divertible Protocols and Atomic Proxy Cryptography*". In *Proc. of EUROCRYPT '98*, Espoo, Finland, 1998.

[4] A. Boldyreva, V. Goyal, and V. Kumar. "*Identity-based Encryption with Efficient Revocation*". In *Proc. of CCS'08*, Alexandria, Virginia, USA, 2008.

[5] D. Boneh and M. Franklin. "*Identity-Based Encryption from The Weil Pairing*". In *Proc. of CRYPTO'01*, Santa Barbara, California, USA, 2001.

[6] S. Yu, K. Ren, W. Lou, and J. Li. "*Defending Against Key Abuse Attacks in KP-ABE Enabled Broadcast Systems*". In *Proc. of Securecomm'09*, Athens, Greece, 2009.

[7] L. Cheung and C. Newport. "*Provably Secure Ciphertext Policy ABE*". In *Proc. of CCS'07*, New York, NY, USA, 2007.

[8] S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. "*Over-encryption: Management of Access Control Evolution on Outsourced Data*". In *Proc. of VLDB'07*, Vienna, Austria, 2007.

[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters. "*Attribute-Based Encryption for Fine-grained Access Control of Encrypted Data*". In *Proc. of CCS'06*, Alexandria, Virginia, USA, 2006.

[10] S. Yu, K. Ren, and W. Lou. "*Attribute-Based On-Demand Multicast Group Setup with Membership Anonymity*". In *Proc. of SecureComm'08*, Istanbul, Turkey, 2008.

[11] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu. Plutus: "*Scalable Secure File Sharing on Untrusted Storage*". In *Proc. of FAST'03*, Berkeley, California, USA, 2003.

[12] J. Li, K. Ren, B. Zhu, and Z. Wan." *Privacy-Aware Attribute-Based Encryption with User Accountability*". In *Proc. of ISC'09*, Pisa, Italy, 2009.

[13] S. Yu, K. Ren, and W. Lou. "*Attribute-Based Content Distribution with Hidden Policy*". In *Proc. of NPSEC'08*, Orlando, Florida, USA, 2008.

[14] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters."*Secure Atrribute-Based Systems*". In *Proc. of CCS'06*, New York, NY, USA, 2006.

[15] A. Sahai and B. Waters. "Fuzzy *Identity-Based Encryption*". In *Proc. of EUROCRYPT' 05*, Aarhus, enmark, 2005.

[16] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, "*Sirius: Securing remote untrusted storage*," in *Proc. of NDSS'03*, 2003.

[17] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "*Improved proxy reencryption schemes with applications to secure distributed storage*," in *Proc. of NDSS'05*, 2005.

[18] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "*Enabling public verifiability and data dynamics for storage security in cloud computing,*" in *Proc. Of ESORICS '09*, 2009.

[19] T. Nishide, K. Yoneyama, and K. Ohta, "*Attribute-based encryption with partially hidden encryptor-specified access structures*," in *ACNS'08*. LNCS 5037, 2008, pp. 111–129.

[20] ACL. http://en.wikipedia.org/wiki/Access control list

[21] H. M. Levy, "*Capability-Based Computer Systems*", Digital Equipment Corporation 1984. ISBN 0-932376-22-3.

[22] NIST. "*Role Based Access Control (RBAC) and Role Based Security*". http://csrc.nist.gov/groups/SNS/rbac/

[23] B. Waters, "Ciphertext-*Policy Attribute-Based Encryption: An Expressive,Efficient, and Provably Secure Realization*", http://eprint.iacr.org/2008/290.

[24] V. Goyal, A. Jain, O. Pandey and A. Sahai, "Bounded *Ciphertext-Policy Attribute based Encryption*", In *Proc. of ICALP'08*, Reykjavik, Iceland, 2008

[25] A. Kapadia , P. Tsang, and S. W. Smith, "*Attribute-Based Publishing with Hidden Credentials and Hidden Policies*," In *Proc. of NDSS'07*, San Diego, CA, 2007.

[26] A. Miyaji, M. Nakabayashi, and S. Takano. "*New explicit conditions of elliptic curve traces for FR-reduction*". IEICE Trans. Fundamentals, E84- A(5):1234C43, May 2001.

[27] D. Boneh, B. Lynn, and H. Shacham. "*Short signatures from theWeil pairing*". In *Proc. of Asiacrypt'01*, Gold Coast, Australia, 2001.

[28] M. J. Hinek, S. Jiang, R. Safavi-Naini, and S. F. Shahandashti, "*Attribute-Based Encryption with Key Cloning Protection*", http://eprint.iacr.org/2008/478.