EasyChair Preprint

№ 4367

An Expressive Access Control Mechanism for Cloud Data Outsourcing

Somen Debnath and Bubu Bhuyan

October 11, 2020

An Expressive Access Control Mechanism with user Revocation for Cloud Data Outsourcing

Somen Debnath
Information Technology
Mizoram University
Aizawl, India.
Email:somen@mzu.edu.in,

Dr. Bubu Bhuyan
Information Technology
North East Hill University
Shillong,India.
Email:bbhuyan@nehu.ac.in

***Abstract*—Now a days, intension of people are hosting all kind of information into the cloud to get benefitted in various way without considering security and privacy. Though cloud is providing wonderful services with lot of utilities, fully trust on third party cloud server tends to a lot of risk, especially data storage and outsourcing. In order to build the trust level of users of outsourcing sensitive information maintained by third party un-trusted cloud provider, the user should have partial or full control on security of their own data rather than relying completely on the security provided by the cloud provider. The user to set the access polices on their outsourced sensitive information to decide which user can access what information. The user should be authenticated before accessing any service from the cloud without compromising their privacy. In this paper, we explained various issues of cloud data outsourcing and also propose a dynamic access control mechanism for outsourced cloud data which will give more efficiency and reliability.**

Keywords: Cloud Computing, Dynamic Access Control, Homomorphic Encryption, Attribute Based Encryption, Attribute based Signature.

I. Introduction

Recently, a lot of research has been going on the security and privacy aspects of cloud computing [1], [4]. These researches are primarily aimed at developing scalable, reliable, secure and privacy preserving techniques in order to build up trust of cloud users to outsource their computation and sensitive information through Internet[3]. Cloud computing is an open standard model, which can enable ubiquitous computing and offers on-demand network access to a shared pool of configurable computing resources [5]. It is a promising computing paradigm which provides different types of services on demand basis like applications as a service (e.g. Microsoft online, Google Apps etc.), platform as a service for developing applications(e.g. Windows Azure, Amazon's S3 etc.), and infrastructure as a service (e.g. Amazon's E2, Nimbus etc.). Cloud is a multi-tenant environment, where a large number of users can use it as data storage and access it anytime and anywhere. Cloud user requires minimal management effort and receives scalable and dynamic services, global/remote access and usage control with commercially low cost [4]. Besides these, the cloud provides many more advantages that attract many organizations and individual user to store and outsource their resources from local to remote cloud server[20].

The security and privacy issues [1], [3] on outsourced data in cloud are different from the other paradigm due to the following reasons:

- Cloud servers are usually executed by third party providers which are likely to be outside of users' trusted domain [7]. There is enhanced risk of confidentiality, integrity on outsourced cloud data. While authenticating the user to access outsourced cloud data, the identity of users should remain undisclosed from the cloud provider. Auditing , accountability [20] on outsourced cloud data are very important for the user of the services.
- Cloud server is a multi-tenant environment [1]. Many users are outsourcing their data on the cloud. The adversary and other user may attack and access that data from the cloud if there is no security and privacy.

In order to build up the trust level of users of outsourcing sensitive information maintained by third party un-trusted cloud provider [8], the user should have partial or full control on security [3] of their own data rather than relying completely on the security provided by the cloud provider. The user to set the access polices on their outsourced sensitive information to decide which user can access what information. The user should be authenticated before allowing to access any service from the cloud without compromising their privacy. It is also important to confirm the integrity of cloud data with reliable audit information.

Conventional public/private key cryptographic techniques and access control mechanisms do not scale up in the cloud environment primarily due to following reasons :

- Key distribution: In cloud computing, the data creator will store his outsourced data in encrypted form. The creator is not aware of who will be the potential users at the time of encrypting the data stored in the cloud. So, for allowing to user access the encrypted data, the creator and the user need to be online for key distribution if they used private key techniques. Moreover, using conventional private key techniques, setting up of access control and distribution of required key will be very complex and inefficient due to a large number of users on cloud [8].
- Conventional access control mechanisms [9] usually assume that storage servers and data owners are in the same trusted domain. In conventional access control

TABLE I
COMPARISON OF DIFFERENT ACCESS CONTROL SCHEMES.

Scheme	Security Assumption	Access Structure	Architecture	User Revocation	Attribute Revocation	Revocation Controller	Revocation Method
Yang[6]	q-BDHE	Monotone	Distributed	Yes	Yes	KDC Center	Key Update
Ruj[20]	SEDH	LSSS	Distributed	Yes	No	DO	LSSS Matrix
Hur [2]	DBDH	KEK tree	Centralized	Yes	No	DO	Re-encryption
Xin[21]	CDHP	VSS	Distributed	Yes	Yes	KDC Center	Key Update
Zu[23]	s-BDHE	LSSS	Distributed	Yes	Yes	Cloud Server	Key Update

TABLE II
NOTATIONS

Notations	Meanings
DO	Data Owner
DU	Data User
TCA	Trusted certified Authority
CSP	Cloud Service Provider
KDC	Key Distribution Center
GID	Global Identity
LSSS	Linear secret sharing scheme
MSG	Message
CT	CipherText
H, h	Hash functions,example SHA-1
M	Matrix of dimension $l \times t$ corresponding to the claim predicate
xid	User Identity
PK_{kid}	Public Key of kid KDC
$Cert(xid)$	User certificate issued by TCA
I_A	Encryption Set
SK^*_{xid}	Secret key of xid
PK^*_{xid}	Public key of xid
KDC_{kid}	kidth KDC
S_{kid}	Set of Credentials/Attributes
SK_{kid}	Secret key of kid KDC
$PK_{x_{kid}}$	Public key of credential x by Kid
$SK_{xid,kid}$	User Secret key for decryption

mechanisms, the storage servers are fully trusted and responsible for enforcing and defining of control policies. Unfortunately, this assumption is not applicable in cloud environment due to the cloud storage servers and the data owners are likely to be in two different trust domains. The conventional access control mechanisms [3], [9] like Role Based Access Control(RBAC) and User Based Access Control (UBAC) do not scale up in the cloud environment due to the presence of multiple user. The user revocation is also an issue for conventional access control mechanisms in the cloud.

In this paper, we expressively discuss a access control scheme for data store and outsource with efficient user revocation in cloud. Here, the DO provide the data accessability by enforce access policies on DU's credentials. In the Table 1, explain comparison of different recent access control mechanisms. The symbolic notations with their meaning used in the paper are shown in table 2.

Our contribution in this paper are:

- We proposed dynamic access control scheme for cloud data outsourcing which is Multi-authority in architecture means many KDC centers can generate secret key for DO and DU. Our scheme also support large universal credentials for user which is use in access policy implementation.
- We improve the user revocation by revoking the attributes. In our revocation method, secret key and ciphertext both need to be updated. These updation can be done by corresponding attribute authority but not by DO.
- We reviewed previous several access control mechanisms based on attribute based encryption with user revocation in cloud environment to compare with our scheme and find research gap and future direction.

The remaining of this paper is organized as follows. We explained the related works and background for system design under Section 2 and Section 3 respectively. Sections 4 and 5 give the detail proposed access control mechanism and Performance analysis based on storage overhead, simulation, security analysis of the model. Last section gives conclusion.

II. RELATED WORKS

Recently, a lot of researches are going on exploring the possibility to use attribute based encryption (ABE) technique for designing secure and privacy preserved access control in cloud. In 2005, Sahai and Water [10] proposed ABE, where a user identified uniquely by a set of credentials or attributes. Next, in 2007, Bethencourt et al. [11] addressed the first CP-ABE based on monotonic access structure. Later on, Bethencourt et al.'s [11] security proof improved, but then also it is not sufficiently expressive due to logical conjunction based policy used in the scheme. Moreover, the length of the secret key and ciphertext size increases linearly in this scheme with the total number of attributes. Nishide et al. [12] and Emura et al. [13] proposed improved CP-ABE based on Cheung and Newport's technique. Nishide et al. [12] addressed a technique which access policy is designed with AND gates on multi-value attributes. Emura et al. [13] proposed an improved technique using the same access policy. All those approaches are based on single authority. In 2011, Lewko et al. [17] proposed first a fully decentralized ABE scheme. This scheme is most suitable for cloud environment to provide confidentiality. The major drawback of this scheme is that scalability and efficiency with no user revocation facility. In 2015, Rouselakis[19] proposed the improvement of Lewko scheme with large universe Multi-authority CP-ABE. Yang et al. [6] suggested a scheme introducing user revocation with [17] for legitimate users where the user hides its own identity at the time of accessing the cloud. Ruj et al. [20] addressed a distributed access control mechanism based on ABE and attributed based signature which include user authentication. Recently, Yangli et al. [24] proposed constant cipher text attributed based encryption on multi-authority however not include user revocation with the scheme.

III. BACKGROUND

A. System Model

The system architecture for developing our scheme shown in figure 1. similar to [22]. This system model has five types of entities: DO, DU, TCA, CSP, KDC. The TCA works as a trustee for registration of all users and KDCs. TCA assigns a global Identity GID uniquely and generate public key for each authorized users.

Every KDCs is an Attribute authority assigned user attributes/credentials with corresponding decryption Key. Defin-

ing and managing of each attributes done by KDC with generating of public and private key for each attribute for the users. Every user have its global identity GID in this system which is generated by TCA and verifiable of this GID by the KDC. Each user may be hold multiple attributes based on their role received from different KDCs and also received Secret key for decryption of ciphertext from the KDC for each attributes.

The function of data owner is generation of ciphertext for the message which he/she wants to upload to Cloud server by dividing into several components and encrypts each plain text components using symmetric key encryption technique and marge each cipher text content key with own access policy. The owner do not depends on the access policy of CSP. The users, whose are having different attribute secret keys under the policy defined by data owner, allowed to decrypt the specific cipher Text.

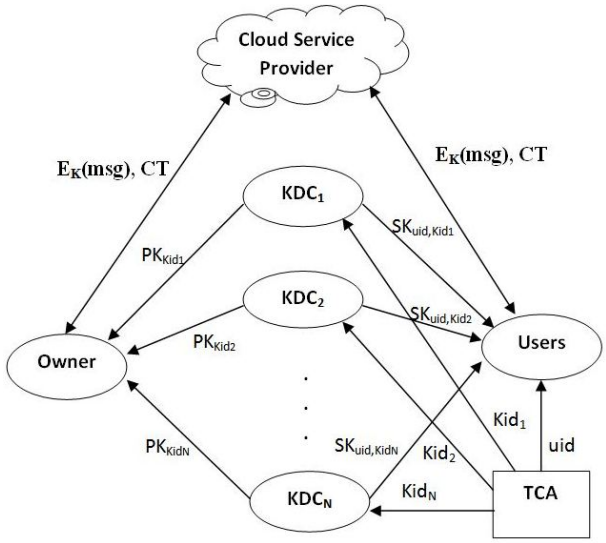

Fig. 1. System model for Access Control

B. Bilinear Pairing

Let G_1 and G_2 are cyclic group of prime order p generated by g. Let G_T be a group of order q. We can define the map $e : G_1 \times G_2 \rightarrow G_T$. The map satisfies the following properties:

1) $e(u^a, v^b) = e(u, v)^{ab}$; $u \in G_1$ and $v \in G_2$ $a, b \in Z_q, Z_q = \{0, 1, 2, ..., q1\}$;
2) Non-degenerate: $e(g, g) \neq 1$;
3) Computable: Efficient computability for any input pair.

Construction of bilinear maps can be done from certain elliptic curves. The choice of the curve is an important consideration because curve determines the complexity of pairing operation[20]. We used here symmetric pairing for development of scheme[17].

C. Complexity Assumption

The vary important feature for developing a security scheme is define security parameter. The security assumption for our scheme is decisional q-BDHE problem[19], [22]. (Decisional q-BDHE Assumption) Let G_1 be a cyclic group of order p and $e : G_1 \times G_1 \rightarrow G_T$. The decisional q-BDHE assumption is q a problem, for given a $(2q + 1)$ tuple $(g, h, g^{\alpha}, g^{\alpha^2}, ..., g^{\alpha^q}, ..., g^{\alpha^{q+2}}, ..., g^{\alpha^{2q}}) \in G_1$.(where α is random, $\alpha \in Z_p^*$) and a random element $T \in G_T$ to decide $T = e(g, h)^{\alpha^{q+1}}$ or not.

D. Access Structure and LSSS

Defining access policy and secrecy sharing are another important issues for development of security scheme. We are not develop any access structure and secret sharing scheme. We used this feature similar to [18] Let $P = \{p_1, p_2, ..., p_n\}$ be the attribute universe. An Access structure on P is a collection A of non-empty set of attributes, i.e. $\mathbb{A} \subseteq 2^U \backslash \{\}$. A (monotone) access structure is a (monotone) collection $\mathbb{A}$ of non-empty subsets of $\{p_1, p_2, ..., p_n\}$. The sets in A are called authorized sets and sets not in $\mathbb{A}$ are called unauthorized sets. Additionally, an access structure is called monotone if $\forall B, C \in \mathbb{A} : if B \in \mathbb{A}$ and $B \subseteq C$. then $C \in \mathbb{A}$. (Linear Secret Sharing Schemes(LSSS))[18]. A secret sharing scheme $\prod$ over a set of parties P is called linear over Z_p if

1) The shares for each party form a vector over Z_p.
2) There exists a matrix $M \in Z_p^{l \times n}$, with row labels $\rho(i) \in P, \forall i \in [l]$.

The column vector $v = (s, r_2, ..., r_n)$, where $s \in Zp$ is the secret to be shared, $r_2, ..., r_n$ are randomly select in Z_p. M_v is the vector of shares.

Every LSSS enjoys the linear reconstruction property : Let $S \in A$ be an authorized set, where A is the access structure. Let $I \subseteq \{1, 2, ..., l\}$ and the label function is $\rho(i) \in S$. There exist constants$\{\omega_i \in Zp\}_{i \in I}$, if λ_i are valid shares of any secret s according to $\prod$, then $\sum_{i \in I}^{\omega_i \lambda_i} = s$. But for any unauthorized set, the secret s is hidden.

IV. Proposed Access Control System for Cloud Data Outsourcing

We design our access control system with multiple key distributed center and one global trusted center. We have used attribute/credential revocation for user revoke with backward and forward security. We use Rouselakis's large universe multi-authority ABE scheme for security development. We modified this scheme and introduced user revocation with it. In this model the global trusted center are responsible for generation of user key with user identity as initial setup. It is also responsible for generation of global setup for all key distribution center. It register each key distribution center which can generate various credentials/attributes for the user with their keys. KDC also generate user secret key for decryption the ciphertext which is stored in cloud by DO. For encryption the data creator use different credentials with their keys which is independent of user identity. Privacy also achieved here from the cloud as the encrypted data in independent of user's identity. The user revocation is done very efficiently here. The whole process explained in details as follows.

A. System Setup

The system setup contains TCA Setup and KDC Setup.

TCA-GlobalSetup The initialization of system done by $TCASetup$. The algorithm takes security parameter 1^{λ} as input. TCA chooses two multiplicative groups G and G_T of prime order p and construct a bilinear function $e : G \times G \rightarrow G_T$ .There is a random hash function is $H : \{0,1\}^* \rightarrow G$. TCA chooses two numbers randomly $a, b \in Z_p$. Then the system's master key will be $MK = \{a, b\}$ and the global parameters , $GP = (g, g^a, g^b, H)$.
The TCA is responsible for initial DO, DU and KDCs registration.

1) User Registration: All DO, DUs need to register under TCA at the time of system setup. For legitimate user in system , the TCA assign GID as xid to each user. The TCA will generate for each user xid two random numbers $(a_{xid}, \bar{a}_{xid}) \in Z_p$ and generate secret keys for globally used are $SK_{xid} = a_{xid}, SK^*_{xid} = \bar{a}_{xid}$. It also generates user's public keys for globally used are $PK_{xid} = g^{a_{xid}}, PK^*_{xid} = g^{\bar{a}_{xid}}$
The TCA produce a certificate $Cert(xid)$ for the user xid. After that, TCA send to the user xid. as user identity one of the user's PK_{xid} and SK^*_{xid} with Certificate $Cert(xid)$.

2) **KDC Registration**: Every KDC need to register itself under the TCA at the time of the system setup. Each registered KDC will receive global Key distribution identity kid from the TCA . To verify the registered user certificate, TCA sends verification key which include another set of user global public/ secret key PK^*_{xid} , SK_{xid} to the KDC KDC_{kid}

3) **KDCSetup for User's Credential Generation**
Let S_{kid} , set of credentials managed by all key distribution center KDC_{kid} . It chooses randomly three numbers $\alpha_{kid}, \beta_{kid}, \gamma_{kid} \in Z_p$ as the KDC secret key

$$SK_{kid} = (\alpha_{kid}, \beta_{kid}, \gamma_{kid}),$$

where α_{kid} is used for data encryption,γ_{kid} is used for attribute revocation and β_{kid} is used to differentiate credentials from various KDCs.

Each KDC publish public key PK_{kid}

$$PK_{kid} = (e(g,g)^{\alpha_{kid}}, g^{\beta_{kid}}, g^{\frac{1}{\beta_{kid}}}),$$

For each credential $x_{kid} \in S_{kid}$, the KDC_kid announced a public credential key as

$$PK_{x_{kid}} = (PK_{1,x_{kid}} = H(x_{kid})^{v_{x_{kid}}}, \\ PK_{2,x_{kid}} = H(x_{kid})^{v_{x_{kid}}\gamma_{kid}})$$

The KDC also choose here version key for the attribute/user credential $VK_{x_{kid}} = v_{x_{kid}}$. All the public credential keys $\{PK_{x_{kid}}\}_{x_{kid} \in S_{kid}}$ are announced publicly of the KDC_{kid} with the public key PK_{kid} of its own.

B. Decryption Key Generation

After checking of user authentication of each xid , KDC_{kid} will assign some credentials to the user xid. The authenticity of user done by KDC_{kid} after checking user certificate $Cert(xid)$ with a verification key received from TCA.

For a legitimate user xid, KDC KDC_{kid} entitles a set of credential $S_{xid,kid}$ based on user identity or role. Then, the KDC_{kid} produced user's decryption key $SK_{xid,kid}$ after randomly choose a number $t_{xid,kid} \in Z_p$. The User's Decryption Key is

$$SK_{xid,kid} = (K_{xid,kid} = g^{\alpha_{kid}} g^{au_{xid}} g^{bt_{xid,kid}}, \\ \bar{K}_{xid,kid} = g^{t_{xid,kid}}, \forall x_{kid} \in S_{kid,xid} : K_{x_{kid},xid} = \\ g^{\bar{u}_{xid} t_{xid,kid} \beta_{kid}} H(x_{kid})^{v_{x_{kid}} \beta_{kid}(u_{xid}+\gamma_{kid})}$$

C. Ciphertext Generation for hosting in Cloud

The data owner can use symmetric key encryption for large size of data to increase efficiency of encipherment dividing in several components of whole message msg and use symmetric key encryption for each of components. If the msg divided in several components then each key content can use this algorithm for generation CT otherwise for smaller msg directly can apply this algorithm as follows.
This algorithm takes as inputs GP, $\{PK_{kid_i}\}_{kid_i \in I_A}$ (set of public key of KDCs in a encryption set I_A, the msg (if applied symmetric key encryption the content key as msg) and overall invoked credentials. It also takes access structure M, where M is $l \times n$ matrix. Here l denotes the total number credentials.The function ρ maps each row of M to an credential.
To encrypt the msg, this algorithm first randomly chooses a encryption secret $s \in Z_p$ and also chooses a random vector $\vec{v} = (s, y_2, .., y_n) \in Z^n_p$, where secret s is share among $(y_2, .., y_n)$ by secret share scheme. For $j = 1, ..., l$, where M_j it computes $\lambda_j = \vec{v}.M_j$, corresponding to the j-th row of M. Then, it randomly chooses $r_1, r_2, ..., r_l \in Z_p$ and computes the ciphertext as

$$\mathbf{CT} = (C = msg.(\prod_{kid_i \in I_A} PK_{kid_i})^s, C^* = g^s, C^{**} = g^{bs}, \\ \forall 1 \leq j \leq l, \rho(j) \in S_{kid_i} : C_j = g^{a\lambda_j}.(PK_{1,\rho(j)})^{-r_j}, C^*_j = g^{r_j} \\ D_j = g^{\frac{r_j}{\beta_{kid_i}}}, D^*_j = (PK_{2,\rho(j)})^{r_j}).$$

D. Cloud Cipher Data Decryption

The legitimate users can decrypt the encrypted data or encrypted content key after receiving different secret keys from different KDC center. The satisfied credential user can decrypt by decryption keys of legitimate user.

The decryption algorithm **Decrypt** $(\mathbf{CT}, PK_{xid}, SK^*_{xid}, \{SK_{kid_k}\}_{kid_k \in I_A})$
$\rightarrow C_k$. Here inputs are ciphertext $\mathbf{CT}$ which contains an access policy $(\mathcal{A}, \rho)$, a global secret key SK^*_{xid} , a global public key $PK_{xid}$of the user xid and a set of secret keys $\{SK_{kid_i}\}_{kid_i \in I_A}$ of involved $KDCs$. If the user's(xid's) credentials/attributes $\{SK_{kid_i}\}_{kid_i \in I_A}$ satisfy the access policy $(\mathcal{A}, \rho)$, the algorithm allows to decrypt the ciphertext as follows.

Let I be involved KDC $\{I_{kid_i}\}_{kid_i \in I_A}$, where $I_{kid_i} \subset \{1, 2, ..., l\}$ is define as $I_{kid_i} = \{j : \rho(j) \in S_{kid_i}\}$. Let the number of KDCs included in ciphertext be $\eta_A = |I_A|$. The algorithm choose $\{\omega_j \in Z_p\}_{j \in I}$ to reconstruct encryption secret $s = \sum_{j \in I} \omega_j \lambda_j$ if $\{\lambda_j\}$ are valid shares of secret s according to access structure M. It computes

$$\prod_{kid_i \in I_A} e(C^*, \bar{K}_{xid,kid_i}) e(C^{**}, K_{xid,kid_i})^{-1} = e(g,g)^{au_{xid}\eta_A s}. \prod_{kid_i \in I_A} e(g,g)^{s}\alpha_{kid}.$$

For each $j \in I$ suppose $\rho(i) \in S_{kid_i}$, it computes

$$e(C_j, PK_{xid}) e(D_j, K_{\rho(j),xid}) e(C_j^*, \bar{K}_{xid,kid_i}^{-SK^*_{xid}}) e(g, D_j^*)^{-1} = e(g,g)^{au_{xid}\lambda_j}$$

Then

$$\prod_{kid_i \in I_A} \prod_{j \in I_{kid_i}} (e(g,g)^{au_{xid}\lambda_j})^{\omega_j \eta_A} = e(g,g)^{au_{xid}\eta_A s}$$

then the user will obtains $\prod_{i \in I_A} e(g,g)^{\alpha_{kid_i} s}$ and use it for decrypt

$$msg = C / \prod_{i \in I_A} e(g,g)^{\alpha_{kid_i} s}$$

E. Attribute Revocation

Attribute/credential revocation is important for redocked user(Backward Security) and the newly joined user(Forward Security). For achieving the attribute revocation we mainly focus on three very important matter of new legitimate user key generation, update ciphertext generation and also update key generation for the key distribution center are shown below.

- **UpdateKeyGen**$(SK_{k\tilde{id}}, \tilde{x}_{k\tilde{id}}, VK_{\tilde{x}_{k\tilde{id}}}) \rightarrow (\bar{VK}_{\tilde{x}_{k\tilde{id}}}, UK_{s,\tilde{x}_{k\tilde{id}}}, UK_{c,\tilde{x}_{k\tilde{id}}})$. To handle the revoked credential $\tilde{x}_{k\tilde{id}}$, this algorithm is executed by the corrosponding $KDC_{k\tilde{id}}$. It takes current version key $VK_{\tilde{x}_{k\tilde{id}}}$, the secret key $SK_{k\tilde{id}}$ of $KDC_{k\tilde{id}}$ and the revoked credential $\tilde{x}_{k\tilde{id}}$ as inputs. It outputs a new $\bar{VK}_{\tilde{x}_{k\tilde{id}}}$, the update key $UK_{c,\tilde{x}_{k\tilde{id}}}$ (for ciphertext update) and the update key $UK_{s,\tilde{x}_{k\tilde{id}}}$ (for secret key update).
- **SKUpdate** $(SK_{uid,k\tilde{id}}, UK_{s,\tilde{x}_{k\tilde{id}}}) \rightarrow \bar{SK}_{uid,k\tilde{id}}$. The unrevoked user uid executed this algorithm. The inputs of the algorithm are the recent secret key of the unrevoked user $SK_{uid,k\tilde{id}}$ and the update key $UK_{s,\tilde{x}_{k\tilde{id}}}$. The output of the algorithm is a new secret key $\bar{SK}_{uid,k\tilde{id}}$ of every unrevoked user uid.
- **CTUpdate**$(\mathbf{CT}, UK_{c,\tilde{x}_{k\tilde{id}}}) \rightarrow \bar{\mathbf{CT}}$. The cloud server executed this algorithm. The inputs are previous ciphertext which contain revoked attribute $\tilde{x}_{k\tilde{id}}$ and the update key $UK_{c,\tilde{x}_{k\tilde{id}}}$. The generated output is new ciphertext $\bar{\mathbf{CT}}$ contained with latest version of revoked credentials $\tilde{x}_{k\tilde{id}}$.

V. Performance Analysis

A. Storage Overhead

We will analyze storage overhead for our scheme by comparing with Yang's DAC-MAC scheme [6]and Ruj's DACC scheme[20]. This is vary significance issue of access control scheme. We are comparing other schemes with our scheme based on some security level. The performance of the schemes are shown in table 3. The notation used inside the table with their significance as follows: $|q|$ denote element size used in bilinear group; N_A denote KDC center ; N_c denote number of ciphertext stored on cloud; N_U denote number of users in the system; N_{kid} denoted the number of credential under N_A KDC; N_o denote number of DO; $N_{a,xid}$ denote number of credentials hold by user xid and l denotes the number of credentials in ciphertext.

B. Simulation

To simulate our system, we use Python based scripting tools Charm-Crypto[15]. It is a framework to simulate modern cryptosystem similar to theoretical implementations. It binds with PBC library for efficient group operations. Charm-Crypto also provides predefined linear secret sharing scheme(LSSS) routines to use, which is very useful to implement Attribute-Based systems. Charm-crypto provided several elliptic curve based on bilinear pairing group i.e.three MNT asymmetric EC groups and two super-singular(SS) symmetric EC groups. Some of the utilized EC groups are "SS512" provides 80 bits in security level, "MNT201" provides 90 bits in security level, "MNT224" and " SS1024" provides 100 and 120 bits in security level respectively.
We have tested our access control model on Intel® $Core^{TM}$ i3-5005M CPU @ 2.00GHz × 2 with 4.0 GB RAM running Ubuntu 16.04 LTS and Python 3.5. We have use symmetric pairing for developing access control model as security assumption.
After simulating we got the timing result in our system for symmetric pairing system based on "SS512" security level as shown in table 4. Timing of each algorithm shown in milliseconds.

C. Security Analysis

For Security analysis of the system we are playing a game between an adversary and challenger. We are assuming that the adversary can statically corrupt the KDCs but adaptively make the key queries. Let S_{KDC} are the set of KDCs. Now the security game is define as follows:
Setup. TCA generate first GP. The adversary specifies $\acute{S}_{KDC} \in S_{KDC}$. The challenger generates the public keys and the secret keys form the $\acute{S}_{KDC}$. For corrupted KDC in $\acute{S}_{K\acute{D}C}$ and secret keys to the adversary. For uncorrupted(KDC) in $S_{KDC} - \acute{S}_{K\acute{D}C}$ sends the public keys to the adversary.
Phase 1. The adversary enquire secret key by submitting pairs (xid, S_{xid}) to the challenger, where

Scheme	KDC($\|q\|$)	Data Owner($\|q\|$)	Data User($\|q\|$))	Server($\|q\|$)
DAC-MACS [6]	$(2N_{kid} + 2N_o + 1)$	$(N_c + N_A + N_a + 2)$	$(N_O(N_A + N_{a,xid}))$	$(l + 2)$
DACC[20]	$(2N_{kid})$	$(N_c + 2N_a)$	$(N_{c,x} + N_{a,xid})$	$(3l + 1)$
Our Scheme	$(N_{kid} + 2N_u + 3)$	$(3N_A + N_a + 3)$	$(N_A + N_{a,xid})$	$(4l + 3)$

TABLE III
COMPARISON OF STORAGE OVERHEAD

No. of Credentials	Initial Setup	Authority Setup	Secret KeyGen	Encryption	Decryption
2	0.0577	0.2647	0.1352	2.1577	1.4685
4	0.0655	0.4290	0.2588	3.9795	2.5017
6	0.0706	0.5910	0.3707	5.6791	3.5271
8	0.0646	0.7399	0.4981	7.4440	4.5499
10	0.0608	0.8643	0.5514	9.2255	5.5854

TABLE IV
OUR ACCESS CONTROL SCHEME(COMPUTATION TIME WITH NO. OF CREDENTIALS)

$S_{xid} = \{S_{xid,kid_k}\}_{kid_k \in S_{KDC} - \acute{S}_{KDC}}$ is a set of credential belonging to KDC several uncorrupted KDCs, and xid is a user identifier. A set of secret keys $\{SK_{xid,kid_k}\}$ gives by challenger to the adversary. By submitting a set of credential $\acute{S}_{kid}$ adversary makes key queries to challenger for corresponding update keys.

Challenge. The adversary takes two messages m_0 and m_1 of equal length and challenges the access structure (M^*, ρ^*). Here access structure need to satisfy the some constraints. The corrupted KDC labeled by V of rows of M^*. The credential x from credential sets labeled for each user xid by V_{xid} for the subset of rows of M^*. For each xid, we require that the subspace spanned by $V \bigcup V_{xid}$ must not include $(1, 0, ..., 0)$. This means combining with the keys from corrupted KDCs the adversary can not ask the keys for decryption the cipher text. Then, under the access structure (M^*, ρ^*), the challenger encrypts m_c by flip a random coin c.The adversary receive new resultant ciphertext CT^*.

Phase 2. Without violating the challenge access structure (M^*, ρ^*), The adversary may enquire multiple set of updated secret key. Interesting things is that none of the updated secret key set able to decrypt the challenged ciphertext.

Guess. The adversary outputs a guess $\acute{c}$ of $c \in \{0, 1\}$.

The advantage of an adversary KDC in this game is defined as $Pr[\acute{c} = c] - \frac{1}{2}$.

VI. CONCLUSION

We have presented here a dynamic access control mechanism to ensure secure outsourcing in cloud environment with attribute revocation. Our scheme is efficient and secure under selective security. The access policy is open for cloud and users inside the ciphertext in this proposed scheme. This work is expandable with hidden access policy. Key distribution is done here dynamically in distributed manner. This work provides an expressive, and secure over encrypted data in cloud environment and also applicable for social networking site, remote storage etc.

REFERENCES

[1] A. Sharma, " Privacy and Security issues in Cloud Computing", Journal of Global Research in Computer Science, vol. 4, no. 9, pp. 15–17, 2013.

[2] Hur, Junbeom, and Dong Kun Noh. "Attribute-based access control with efficient revocation in data outsourcing systems." IEEE Transactions on Parallel and Distributed Systems 22.7 (2011): 1214-1221.

[3] S. Debnath, S.Sahana, B. Bhuyan " A Distributed Fine - grained Access Control Mechanism for Cloud Data Outsourcing", SCience & Technology Journal (pp. 78-85). 2015 vol 3.

[4] D. Catteddu, " Cloud Computing: Benefits, Risks and Recommendations for Information Security", *Web Application Security*, Springer Berlin Heidelberg, pp. 11–17, 2010.

[5] P. Mell and T. Grance, " The NIST definition of Cloud Computing", *Computer Security Division*, Information Technology Laboratory, NIST Gaithersburg, 2011.

[6] Yang, K., and Jia, X. (2014). " DAC-MACS: Effective data access control for multi-authority cloud storage systems". In Security for Cloud Storage Systems (pp. 59-83). Springer New York.

[7] D. Zissis and D. Lekkas, " Addressing Cloud Computing Security issues", *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012.

[8] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, " Toward Secure and Dependable Storage Services in Cloud Computing", *IEEE Transactions on Services Computing*, vol. 5, no. 2, pp. 220–232, 2012.

[9] Y. A. Younis, K. Kifayat, and M. Merabti, " An Access Control Model for Cloud Computing", *Journal of Information Security and Applications*, vol. 19, no. 1, pp. 45–60, 2014.

[10] A. Sahai and B. Waters, " Fuzzy Identity-Based Encryption", *EUROCRYPT. Lecture Notes in Computer Science*, R. Cramer, Ed, vol. 3494, pp. 457–473, 2005.

[11] J. Bethencourt, A. Sahai, and B. Waters, " Ciphertext-policy Attribute-based Encryption", *SP'07. IEEE Symposium on Security and Privacy*, IEEE, pp. 321–334, 2007.

[12] T. Nishide, K. Yoneyama, and K. Ohta, " Attribute-based Encryption with Partially Hidden Encryptor specified Access Structures", *Applied Cryptography and Network Security, Springer Berlin Heidelberg*, pp. 111–129, 2008.

[13] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, " A Ciphertext-Policy Attribute Based Encryption Scheme with Constant Ciphertext Length", *Information Security Practice and Experience*, Springer Berlin Heidelberg, pp. 13–23, 2009.

[14] X. Liang, Z. Cao, H. Lin, and J. Shao, " Attribute Based Proxy Re-encryption with Delegating Capabilities", *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, ACM, New York, pp. 276–286, 2009.

[15] Joseph A. Akinyele, Matthew Green, and Avi Rubin. " Charm: A framework for rapidly prototyping cryptosystems". *Cryptology ePrint Archive*, Report 2011/617, 2011. http://eprint.iacr.org/.

[16] M. Chase and S. S. Chow, " Improving Privacy and Security in Multi-authority Attribute Based Encryption", *Proceedings of the 16th ACM conference on Computer and Communications Security*, Chicago, ACM, pp. 121–130, 2009.

[17] A. Lewko and B.. Waters. " Decentralizing attribute-based encryption". *In Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 568–588. Springer, 2011.

[18] Z. Liu, Z. Cao, and D. . Wong." Efficient generation of linear secret sharing scheme matrices from threshold access trees". *Technical report, IACR Cryptology* ePrint Archive, 2010.

[19] Y. Rouselakis and B. Waters. " Efficient statically-secure large-universe multi-authority attribute-based encryption". *In International Conference on Financial Cryptography and Data Security*, pages 315–332.Springer, 2015.

[20] S. Ruj and M. Stojmenovic. " Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds".*IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, VOL. 25, NO.2, FEBRUARY 2014, 25(2):384–394, 2014.

[21] L. Xin, X. Sun, Z. Fu, L. Zhang, and J. Xi. " Effective and secure access control for multi-authority cloud storage systems". *International Journal of Security and Its Applications*, 10(2):217–236, 2016.

[22] K. Yang and X. Jia. " Expressive, efficient, and revocable data access control for multi-authority cloud storage".*IEEE transactions on parallel and distributed systems*, 25(7):1735–1744, 2014.

[23] Zu, L., Liu, Z., and Li, J. (2014, September). New ciphertext-policy attribute-based encryption with efficient revocation. In Computer and Information Technology (CIT), 2014 IEEE International Conference on (pp. 281-287). IEEE.

[24] C. Yanli, S. Lingling, and Y. Geng. " Attribute-based access control for multi-authority systems with constant size ciphertext in cloud computing". *China Communications*, 13(2):146–162, 2016.