International Journal Of Advanced Engineering & Innovative Technology (IJAEIT)
ISSN: 2348 7208
ISRA IMPACT FACTOR: 0.42

Decentralized Access Control Technique with Anonymous Authentication

Kavya Ratna A. & K.C.Thiagarajan

ABSTRACT: In this paper, a new decentralized access control scheme for secure data storage in clouds that supports anonymous authentication is proposed. In the proposed scheme, the cloud verifies the authenticity of the series without knowing the user's identity before storing data. This scheme also has the added feature of access control in which only valid users are able to decrypt the stored information. The scheme prevents replay attacks and supports creation, modification, and reading data stored in the cloud. This paper also address user revocation. Moreover, our authentication and access control scheme is decentralized and robust, unlike other access control schemes designed for clouds which are centralized. The communication, computation, and storage overheads are comparable to centralized approaches.
Index Terms—Access control, authentication, attribute-based signatures, attribute-based encryption, cloud storage.

I INTRODUCTION:

Research in cloud computing is receiving a lot of attention from both academic and industrial worlds. In cloud computing, users can outsource their computation and storage to servers using Internet. This frees users from the hassles of maintaining resources on-site. Clouds can provide several types of services like applications, infrastructures, and platforms to help developers write applications. Much of the data stored in clouds is highly sensitive, for example, medical records and social networks. Security and privacy are, thus, very important issues in cloud computing. In one hand, the user should authenticate itself before initiating any transaction, and on the other hand, it must be ensured that the cloud does not tamper with the data that is outsourced. User privacy is also required so that the cloud or other users do not know the identity of the user. The cloud can hold the user accountable for the data it outsources, and likewise, the cloud is itself accountable for the services it provides. The validity of the user who stores the data is also verified. Apart from the technical solutions to ensure security and privacy, there is also a need for law enforcement.

Cloud servers prone to Byzantine failure, where a storage server can fail in arbitrary ways. The cloud is also prone to data modification and server colluding attacks. In server colluding attack, the adversary can compromise storage servers, so that it can modify data files as long as they are internally consistent. To provide secure data storage, the data needs to be encrypted. However, the data is often modified and this dynamic property needs to be taken into account while designing efficient secure storage techniques. Efficient search on encrypted data is also an important concern in clouds. The clouds should not know the query but should be able to return the records that satisfy the query.

Kavya Ratna A. &K.C.Thiagarajan Research students of SRS College of Engineering and Technology.

This is achieved by means of searchable encryption. The keywords are sent to the cloud encrypted, and the cloud returns the result without knowing the actual keyword for the search. The problem here is that the data records should have keywords associated with them to enable the search. The correct records are returned only when searched with the exact keywords.

Security and privacy protection in clouds are being explored by many researchers. Many homomorphism encryption techniques have been suggested to ensure that the cloud is not able to read the data while performing computations on them. Using homomorphicencryption, the cloud receives cipher text of the data and performs computations on the cipher text and returns the encoded value of the result. The user is able to decode the result, but the cloud does not know what data it has operated on. In such circumstances, it must be possible for the user to verify that the cloud returns correct results. Accountability of clouds is a very challenging task and involves technical issues and law enforcement. Neither clouds nor users should deny any operations performed or requested. It is important to have log of the transactions performed; however, it is an important concern to decide how much information to keep in the log. Considering the following situation: A law student, Alice, wants to send a series of reports about some malpractices by authorities of University X to all the professors of University X, research chairs of universities in the country, and students belonging to Law department in all universities in the province. She wants to remain anonymous while publishing all evidence of malpractice. She stores the information in the cloud. Access control is important in such case, so that only authorized users can access the data. It is also important to verify that the information comes from a reliable source. The problems of access control, authentication, and privacy protection should be solved simultaneously. We address this problem in its entirety in this paper. Access control in clouds is gaining attention because it is important that only authorized users have access to

valid service. A huge amount of information is being stored in the cloud, and much of this is sensitive information. Care should be taken to ensure access control of this sensitive information which can often be related to health, important documents (as in Google Docs or Dropbox) or even personal information (as in social networking). There are broadly three types of access control: user-based access control (UBAC), role-based access control (RBAC), and attribute-based access control (ABAC). In UBAC, the access control list contains the list of users who are authorized to access data. This is not feasible in clouds where there are many users. In RBAC users are classified based on their individual roles. Data can be accessed by users who have matching roles. The roles are defined by the system. For example, only faculty members and senior secretaries might have access to data but not the junior secretaries. ABAC is more extended in scope, in which users are given attributes, and the data has attached access policy. Only users with valid set of attributes, satisfying the access policy, can access the data. For instance, in the above example certain records might be accessible by faculty members with more than 10 years of research experience or by senior secretaries with more than 8 years' experience. An area where access control is widely being used is health care. Clouds are being used to store sensitive information about patients to enable access to medical professionals, hospital staff, researchers, and policy makers. It is important to control the access of data so that only authorized users can access the data. Using ABE, the records are encrypted under some access policy and stored in the cloud. Users are given sets of attributes and corresponding keys. Only when the users have matching set of attributes, can they decrypt the information stored in the cloud.

Access control is also gaining importance in online social networking where users store their personal information, pictures, and videos and share them with selected groups of users or communities they belong to. Such data are being stored in clouds. It is very important that only the authorized users are given access to those information. A similar situation arises when data is storedin clouds, for example, in Dropbox, and shared with certain groups of people. It is just not enough to store the contents securely in the cloud but it might also be necessary to ensure anonymity of the user. For example, a user would like to store some sensitive information but does not want to be recognized. The user might want to post a comment on an article, but does not want his/her identity to be disclosed. However, the user should be able to prove to the other users that he/ she is a valid user who stored the information without revealing the identity. There are cryptographic protocols like ring signatures, mesh signatures, group signatures, which can be used in these situations. Ring signature is not a feasible option for clouds where there are a large number of users. Group signatures assume the preexistence of a group which might not be possible in clouds. Mesh signatures do not ensure if the message is from a single user or many users colluding together. For these reasons, a new protocol known as attribute-based signature (ABS) has been applied. In ABS, users have a claim predicate associated with a message. The claim predicate helps to identify the user as an authorized one, without revealing its identity. Other users or the cloud can verify the user and the validity of the message stored. ABS can be combined with ABE to achieve authenticated access control without disclosing the identity of the user to the cloud. Existing work on access control in cloud are centralized in nature. Most schemes use ABE. Some scheme uses a symmetric key approach and does not support authentication. However, the authors take a centralized approach where a single key distribution center (KDC) distributes secret keys and attributes to all users. Unfortunately, a single KDC is not only a single point of failure but difficult to maintain because of the large number of users that are supported in a cloud environment. We, therefore, emphasize that clouds should take a decentralized approach while distributing secret keys and attributes to users. It is also quite natural for clouds to have many KDCs in different locations in the world. Although a decentralized approach is proposed, their technique does not authenticate users, who want to remain anonymous while accessing the cloud. However, the scheme did not provide user authentication. The other drawback was that a user can create and store a file and other users can only read the file. Write access was not permitted to users other than the creator. In the preliminary version of this paper [1], we extend our previous work with added features that enables to authenticate the validity of the message without revealing the identity of the user who has stored information in the cloud. In this version we also address user revocation. We use ABS scheme to achieve authenticity and privacy. Unlike, our scheme is resistant to replay attacks, in which a user can replace fresh data with stale data from a previous write, even if it no longer has valid claim policy. This is an important property because a user, revoked of its attributes, might no longer be able to write to the cloud. We, therefore, add this extra feature in our scheme and modify appropriately. Our scheme also allows writing multiple times which was not permitted in our earlier work .

OUR CONTRIBUTION:

The main contributions of this paper are the following:

1. Distributed access control of data stored in cloud so that only authorized users with valid attributes can access them.
2. Authentication of users who store and modify their data on the cloud.

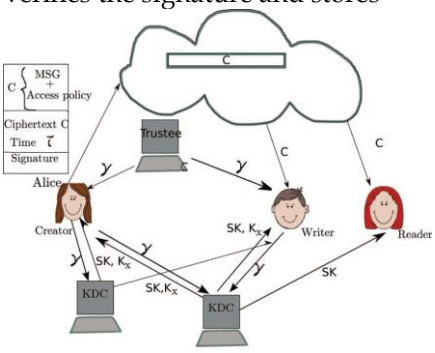
3. The identity of the user is protected from the cloudduring authentication.
4. The architecture is decentralized, meaning that therecan be several KDCs for key Management.
5. The access control and authentication are bothcollusion resistant, meaning that no two users cancollude and access data or authenticate themselves, if they are individually not authorized.
6. Revoked users cannot access data after they havebeen revoked.
7. The proposed scheme is resilient to replay attacks. Awriter whose attributes and keys have been revokedcannot write back stale information.
8. The protocol supports multiple read and write onthe data stored in the cloud.
9. The costs are comparable to the existing centralizedapproaches, and the expensive operations are mostlydone by the cloud.

II RELATED WORK:

In ABE, a userhas a set of attributes in addition to its unique ID. There aretwo classes of ABEs. In key-policy ABE or KP-ABE, the sender has an access policy to encrypt data. Awriter whose attributes and keys have been revoked cannotwrite back stale information. The receiver receives attributes and secret keys from the attribute authority and is able todecrypt information if it has matching attributes. InCiphertext-policy, CP-ABE the receiver has theaccess policy in the form of a tree, with attributes as leavesand monotonic access structure with AND, OR and other thresholdgates. All the approaches take a centralized approach and
allow only one KDC, which is a single point of failure.Chase proposed a multiauthority ABE, in which thereare several KDC authorities (coordinated by a trusted authority) which distribute attributes and secret keys to users. Multiauthority ABE protocol, which required no trusted authority which requiresevery user to have attributes from at all the KDCs. Recently,
Lewko and Waters [35] proposed a fully decentralized ABE where users could have zero or more attributes from each authority and did not require a trusted server. In all these cases, decryption at user's end is computation intensive. So, this technique might be inefficient when users access using their mobile devices. To get over this problem, Green et al. proposed to outsource the decryption task to a proxyserver, so that the user can compute with minimumresources (for example, hand held devices). However, thepresence of one proxy and one KDC makes it less robustthan decentralized approaches. Both these approaches had
no way to authenticate users, anonymously. Yang et al. presented a modification of, authenticate users, who want to remain anonymous while accessing the cloud. Toensure anonymous user authentication ABSs were introduced by Maji et al. This was also a centralizedapproach. A recent scheme by Majietal takes decentralized approach and provides authentication without disclosing the identity of the users. However, as mentionedearlier in the previous section it is prone to replay attack.

III PROPOSED PRIVACY PRESERVING AUTHENTICATED ACCESS CONTROL SCHEME

In this section, we propose our privacy preservingauthenticated access control scheme. According to ourscheme a user can create a file and store it securely in thecloud. This Scheme consists of use of the two protocols ABE and ABS.We will first discuss our scheme in details and then provideaconcrete example to demonstrate how it works. We referto the Fig. 1. There are three users, a creator, a reader, and writer. Creator Alice receives a token _ from the trustee, who is assumed to be honest. A trustee can be someone like the federal government who manages social insurance numbers etc. On presenting her id (like health/social insurance number), the trustee gives her a token _. There are multiple KDCs (here 2), which can be scattered. For example, these can be servers in different parts of the world. A creator on presenting the token to one or more KDCsreceives keys for encryption/decryption and signing. In the Fig. 1, SKs are secret keys given for decryption, Kx are keys for signing. The message MSG is encrypted under the access policy X. The access policy decides who can access the data stored in the cloud. The creator decides on a claimpolicy Y, to prove her authenticity and signs the message under this claim. The ciphertext C with signature is c, and is sent to the cloud. The cloud verifies the signature and stores



thecipher text C. When a reader wants to read, the clouds ends C. If the user has attributes matching with access policy, it can decrypt and get back original message.Write proceeds in the same way as file creation. By designating the verification process to the cloud, it relievesthe individual users from time consuming verifications. When a reader wants to read some data stored in the cloud, it tries to decrypt it using the secret keys it receives from the
KDCs. If it has enough attributes matching with the access policy, then it decrypts the information stored in the cloud.

DATA STORAGE IN CLOUDS

A user $U_u$first registers itself with one or more trustees. For simplicity we assume there is one trustee. The

trustee gives it a token γ=(u,k_{base},k_0,ρ)where ρ *is* the signature onu||K_{base}signed with the trustees private key T_{Sig} (by (6)).The KDCs are given keys $P_k[i]$;$S_k[i]$for encryption decryption and ASK[i]APK[i] for signing/verifying. The user on presenting this token obtains attributes and secretkeys from one or more KDCs. A key for an attribute xbelonging to KDC A_i is calculated $ask_x=k_{base}^{1/(a=bx)}$, where(a,b) €ASK[i]. The user also receives secret keys $sk_{x;u}$ forencrypting messages. The user then creates an access policyX which is a monotone Boolean function. The message is then encrypted under the access policy as

C=ABE.Encrypt(MSG,X)

The user also constructs a claim policy Y to enable the cloud to authenticate the user. The creator does not send themessage MSG as is, but uses the time stamp and creates H(C) iT.This is done to prevent replay attacks. If the timestamp is not sent, then the user can write previous stale message back to the cloud with a valid signature, even when its claim policy and attributes have been revoked. The original work by Majietal.suffers from replay attacks.

In their scheme, a writer can send its message and correct signature even when it no longer has access rights. In our scheme a writer whose rights have been revoked cannot create a new signature with new time stamp and, thus, cannot write back stale information. It then signs the message and calculates the message signature as

ϭ=ABS.Sign(Public key of trustee; Public key of KDC:token; signing key; message; access claim);

The following information is then sent in the cloud

c=(C,T,ϭ,Y).

The cloud on receiving the information verifies the access claim using the algorithm ABS.*verify*. The creatorchecks the value of V=ABS.VERIFY(TPK,ϭ,c,Y).if V=0,then authentication has failed and the message is discarded. Else, the message (C,T) stored in the cloud.

READING FROM THE CLOUD

When a user requests data from the cloud, the cloud sends the ciphertext C using SSH Protocol. Decryption proceeds using algorithm ABE.*DECRYPT*(C,{$SK_{i,U}$ and the messages is calculated

WRITING TO THE CLOUD

To write to an already existing file, the user must send itsmessage with the claim policy as done during file creation. The cloud verifies the claim policy, and only if the user is authentic, is allowed to write on the file.

USER REVOCATION

We have just discussed how to prevent replay attacks. We will now discuss how to handle user revocation. It should be ensured that users must not have the ability to access data, even if they possess matching set of attributes. For this reason, the owners should change the stored data and send updated information to other users. The set of attributes Iupossessed by the revoked user Uu is noted and all users change their stored data that have attributes i 2 Iu. In, revocation involved changing the public and secret keys of the minimal set of attributes which are required to decryptthe data. We do not consider this approach because here different data are encrypted by the same set of attributes, so such a minimal set of attributes is different for different users. Therefore, this does not apply to our model. Once the attributes Iu are identified, all data that possess the attributes are collected. For each such data record, the following steps are then carried out:

1. A new value of s, s,s_{new}€Zqis selected.
2. The first entry of vector v_{new} is changed to new s_{new}.

3$\lambda_x=R_xV_{new}$ is calculated, for each row x corresponding to leaf attributes in Iu.

4. C1,x is recalculated for x.
5. New value of C1,x is securely transmitted to the cloud.
6. New $C_0=Me(g,g)^{snew}$is calculated and stored in the cloud.
7. New value of C1,x is not stored with the data, but is transmitted to users, who wish to decrypt the data.

We note here that the new value of C1,x is not stored in the cloud but transmitted to the no revoked users who have attribute corresponding to x. This prevents a revoked user todecrypt the new value of C0 and get back the message.

IVATTRIBUTE BASED ENCRYPTION

ABE with multiple authorities as proposed by lewko and waters proceed as follows

SYSTEM INITIALIZATION

select a prime generator g of G_0,groups G_0 and G_t of order q, a map e:G_0*G_0->G_T,and a hash function H:{0,1}* -> G_0that maps the identities of users to G0. The hash function used here is SHA-1.Each KDC A_j∈ A has a set of attributes L_J.The attributes disjoint ($Li\cap Lj=\emptyset$ for i≠ j).Each KDC chooses two random exponents α_i,y_i∈ Zq.The secret key of KDC A_j is

SK[j]={αi,yi,i∈ Lj}.

The public key of KDC Aj is published

PK[j]={e(g,g)$^{\alpha i}$,g^{yi},i∈ Lj}

KEY GENERATION AND DISTRIBUTION BY KDCS

User Uu receives a set of attributes I[j, u]from KDC Aj, and corresponding secret key $sk_{i,u}$ for each i∈ $I[j,u]$

$sk_{i,u}=g^{\alpha i}h(u)^{yi}$

where α,y_i∈ $SK[j]$. Note that all keys are delivered to theuser securely using the user's public key, such that only thatuser can decrypt it using its secret key.

ENCRYPTION BY SENDER

The encryption function is ABE.*Encrypt*(MSG,X). Sender decides about the access tree X. LSSS matrix R can be derived. Sender encrypts message MSG as follows:

1. Choose a random seed s∈Zq and a random vector

v∈Z^h_q, with s as its first entry; h is the number of leaves in the access tree (equal to the number of row sin the corresponding matrix R).
2. Calculate λx= Rx .v, where Rx is a row of R.
3. Choose a random vector w ∈$Z^h_q$with 0 as the firstentry.
4. Calculate ω_x¼=Rx .w.
5. For each row Rx of R, choose a random ρ∈Zq.
6. The following parameters are calculated:

c_0=MSGe(g,g)^s,
c1.x=e(g,g)$^{\lambda x}$e(g,g)$^{\alpha}\pi(x)^{\rho x}$,$\forall x$
c2.x=g$^{\rho x}\forall x$,
c3.x=g$^y\pi(x)\rho x g\omega x\forall x$,

where$\pi(x)$ is mapping from Rx to the attribute i thatis located at the corresponding leaf of the access tree.
7. The ciphertext C is sent by the sender (it also includes the access tree via R matrix):
c=<Rπ,c0,{c1.x,c2.x,c3.x,∀x}>

DECRYPTION BY RECEIVER

The decryption function isABE.*Decrypt*(C,{ski;u}), whereC is given by (5). Receiver Uu takes as input ciphertext C,secret keys {ski;u}, group G0, and outputs message *msg*. Itobtains the access matrix R and mapping πfrom C. It thenexecutes the following steps:
1. Uu calculates the set of attributes $\{\pi(x): x \in X\} \cap Iu$that are common to itself and the access matrix. X isthe set of rows of R.
2. For each of these attributes, it checks if there is a subset X0 of rows of R, such that the vector (1.0,.....0) is their linear combination. If not, decryption is impossible. If yes, it calculates constantsCx∈ Zq, such that$\sum x \in X' CxRx$=(i,0.....0)
3. Decryption proceeds as follows:
a.for each x∈ X',dec(x)=$\frac{c1.x(h(u),c3.x)}{e(sk\pi(x)c2.x)}$
b.Uu computes MSG =c0/$\pi x \in X' dec(x)$

V ATTRIBUTE-BASED SIGNATURE SCHEME

ABS scheme [24] has the following steps.

SYSTEM INITIALIZATION

Select a prime q, and groups G1 and G2, which are of orderq. We define the mapping
^e : G1 _ G1->G2.
Let g1; g2 begenerators of G1 and hj be generators of G2, for j∈[tmax],for arbitrary tmax. Let H be a hash function. Let A0 = h_0^{a0},whereas$_0$∈ Zq is chosen at random. (TSig.TVer) meanTSig is the private key with which a message is signed andTVer is the public key used for verification. The secret key
for the trustee is TSK=(a0; TSig) and public key is
TPK =(G1,G2,H, g1,A0,h0,h1,. . . ..,htmax, g2,TV er).

USER REGISTRATION

For a user with identity Uuthe KDC draws at randomKbase∈G. Let K0= k $^{1/a0}_{base}$.The following token γ is output
γ=(u,k_{base},k_0,ρ),
whereρ is signature on u||K_{base}using the signing key TSig.

KDC SETUP

Choose a,b∈ Zq ∗ randomly and compute: A_{ij}=h^a_j , B_{ij}=h^b_j,for Ai∈A, j∈[tmax]. The private key of ith KDC isASK[i] =(a, b)and public key APK[i]=(A_{ij};,B_{ij} |j∈ $[tmax]$).

ATTRIBUTE GENERATION

The token verification algorithm verifies the signaturecontained in γ using the signature verification key TVer inTPK. This algorithm extracts Kbase from γ using (a, b) from ASK[i] and computesk_x=$k_{base}^{1/(a+bx)}$,x∈ $j[i,u]$. Thekey Kx can be checked for consistency using algorithm
ABS.*KeyCheck*(TPK;APK[i],γ,kx), which checks
e^(kx,$A_{ij}B_{ij}^x$)=e^(k_{base},h_j),
for all x∈ $j[i,u] and\ j \in [tmax]$

SIGN

The algorithm

ABS.Sign(TPK,{APK[i] : i∈ $AT[u]$},
γ,{kx:x∈ j_u},MSG,Y),

has input the public key of the trustee, the secret key of the signer, the message to be signed and the policy claim Y. The policy claim is first converted into the span program M∈ Zq. q , with rows labeled with attributes. Mx denotesrow x of M. Let π'denote the mapping from rows to the attributes. So,$\pi'(x)$ is the mapping from Mx to attribute x. A
Vector v is computed that satisfies the assignment {x : x ∈ J[i; u]}. Compute μh(MSG||y).Choose r0∈ Zq and ri∈ Zq,i∈ ju and compute:

y=k_{base}^{ro},si=(k_i^{xi})^{r0}.(g2g1$^{\mu}$)^{ri}($\forall i \in ju$),
w=k0r0,pj=$\pi i \epsilon AT[u](A_{ij}B_{ij}\pi'(i))^{mijri}(\forall j \in [t])$.
the signature is calculated as

$\sigma = (y, w, s1, s2, \ldots . st, p1, p2 \ldots ., pt)$

VERIFY

Algorithm
ABS.*verify*(TPK,σ=(y,w,s1,s2,...st, $p1, p2 \ldots ., pt$),MSG,Y), converts Y to the corresponding monotone program M ∈zq^{l*t}, with rows labeled with attributes.Computeμ=h(MSG||y). If Y=1, ABS.V erify= 0 meaning false.Otherwise, the following constraints are checked

e^(W,A0)=?e^(y.H0),

$\pi i \in e\hat{}(si, AijBij))=?\{\begin{cases} e^{(y,h1)e^{g2g1\mu,p1}}, j=1 \\ e^{g2g1\mu,pj}, j>1, \end{cases}$

where i'=AT[i].

VI SECURITY OF THE PROTOCOL

In this section, we will prove the security of the protocol. We will show that our scheme authenticates a user whoequwants to write to the cloud. A user can only write providedthe cloud is able to validate its access claim. An invalid usercannot receive attributes from a KDC, if it does not have thecredentials from the trustee. If a user's credentials are revoked, then it cannot replace data with previous staledata, thus preventing replay attacks.

Theorem 1.*Our access control scheme is secure (no outsider or cloud can decrypt ciphertext), collusion resistant and allows access only to authorized users.*

Proof. We first show that no unauthorized user can access data from the cloud. We will first prove the validity of our scheme. A user can decrypt data if and only if it has a matching set of attributes. This follows from the fact that access structure S (and hence matrix R) is constructed if and only if there exists a set of rows X0 in R, and linear constants$c_{x\epsilon Z_q}$ such that $\sum x \in X$cxRx-(1,0,.....0).we note that

$$dec(x)=\frac{c1.xe(h(u),c3.x)}{e(sk(x),u,c2.x)}=e(g,g)^{\lambda}e(h(u).g)^{wx}$$

Thus,

$$\pi x \in x'dec(x)$$
$$=\pi x \in x'(e(g,g)\lambda e(h(u),g\omega x$$
$$=e(g,g)^{s}$$

equation above holds because$\lambda_x=R_x.v$ and$\omega x = Rx.\omega$,where v.(1,0,...0)=r and $\omega.(1,0,\ldots.0) = 0c0/\pi x \in X'$dec(x)=c0/e(g,g)s=M.

For an invalid user, there does not exists attributes Corresponding to rows x, such that $\sum x \in 'xCx\ Rx$ (1, 0, . . . , 0). Thus, e(g,g)^s cannot be calculated.

We next show that two or more users cannotcollude and gain access to data that they are not individually supposed to access. Suppose that there exist attributes$\pi(x)$ from the colluders, such that $\sum x \in 'xCx\ Rx$=(1, 0, . . . , 0). However, e(h(u),g)ω needs to be calculatedaccording to (15). Since different users have differentvalues of e(h(u),g) even if they combine their attributes, they cannot decrypt the message.

We next observe that the cloud cannot decode stored data. This is because it does not posses the secret keysski;u (by (3)). Even if it colludes with other users, it cannot decrypt data which the users cannot themselves decrypt, because of the above reason (same as collusion of users). The KDCs are located in different servers and are not owned by the cloud. For this reason, even if some (but not all) KDCs are compromised, the cloud cannot decode data.

Theorem 2.Our authentication scheme is correct, collusion secure, resistant to replay attacks, and protects privacy of the user.

Proof. We first note that only valid users registered with the trustee(s) receive attributes and keys from the KDCs. Auser's token is Kγ=(u,k_{base},k0,ρ)where ρ is signature on ukKbase with TSig belonging to the trustee. An invalid user with a different user-id cannot create the same signature because it does not know TSig.We next show that only a valid user with valid access claim is only able to store the message in the cloud. This follows from the functions Assi*gn* andABS.*Verify*. A user who wants tocreate a file and tries to make a false access claim, cannot do so, because it will not have attribute keys from the related KDCs. At the same time since the message is encrypted, a user without valid accesspolicy cannot decrypt and change the information.

Table 1NOTATIONS

Symbols	Computation
E_π	Exponentiation in group G_π
τ_H	Time to hash using function H
$\tau_{\mathcal{H}}$	Time to hash using function $\mathcal{H}$
$\tau_{l'}/\tau_{\hat{l}'}$	Time taken to perform 1 pairing operation in $e/\hat{e}$
$\lvert G\rvert$	Size of group G
a	Number of KDCs which contribute keys to user

Two users cannot collude and create an access policy consisting of attributes shared between them. Suppose, there are two users A and B who have attributes x_A andx_B, respectively. They have the following informationK_{baseA},$K_{xA}$and K_{baseB},K_{xB} , respectively. A new value of

$$K_{xB}=K_{baseA}^{1/(a+bx')}$$

Cannot be calculated by B, because itdoes not know the values of ða; bÞ. Thus, the authentications collusion secure. Our scheme is resistant to replay attacks. If a writer's access claims are revoked, it cannot replace a data with stale information from previous writes. This is because it has to attach a new time stamp and sign the messageH© it again. Since it does not have attributes, it cannot have a valid signature.

VII CONCLUSION

We have presented a decentralized access control technique with anonymous authentication, which provides user revocation and prevents replay attacks. The cloud does not know the identity of the user who stores information, but only verifies the user's credentials. Key distribution is done in a decentralized way. One limitation is that the cloud knows the access policy for each record stored in the cloud. In future, we would like to hide the attributes and access policy of a user.

VIII REFERENCES

[1] S. Ruj, M. Stojmenovic, and A. Nayak, "Privacy Preserving Access Control with Authentication for Securing Data in Clouds," Proc.IEEE/ACM Int'l Symp.Cluster, Cloud and Grid Computing, pp. 556-563, 2012.

[2] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward Secure and Dependable Storage Services in Cloud Computing,"IEEE Trans. Services Computing, vol. 5, no. 2, pp. 220-232, Apr.-June 2012.

[3] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "FuzzyKeyword Search Over Encrypted Data in Cloud Computing,"Proc. IEEE INFOCOM, pp. 441-445, 2010.

[4] S. Kamara and K. Lauter, "Cryptographic Cloud Storage," Proc.14th Int'l Conf. Financial Cryptography and Data Security, pp. 136-149,2010.

[5] H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-Based Authentication for Cloud Computing," Proc. First Int'l Conf. Cloud Computing(CloudCom), pp. 157-166, 2009.
[6] C. Gentry, "A Fully Homomorphic Encryption Scheme," PhDdissertation, Stanford Univ., http://www.crypto.stanford.edu/craig, 2009.
[7] A.-R. Sadeghi, T. Schneider, and M. Winandy, "Token-BasedCloud Computing," Proc. Third Int'l Conf. Trust and TrustworthyComputing (TRUST), pp. 417-429, 2010.
[8] R.K.L. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M.Kirchberg, Q. Liang, and B.S. Lee, "Trustcloud: A Frameworkfor Accountability and Trust in Cloud Computing," HP TechnicalReport HPL-2011-38, http://www.hpl.hp.com/techreports/
2011/HPL-2011-38.html, 2013.
[9] R. Lu, X. Lin, X. Liang, and X. Shen, "Secure Provenance: TheEssential of Bread and Butter of Data Forensics in CloudComputing," Proc. Fifth ACM Symp. Information, Computer andComm. Security (ASIACCS), pp. 282-292, 2010.
[10] D.F. Ferraiolo and D.R. Kuhn, "Role-Based Access Controls," Proc.15th Nat'l Computer Security Conf., 1992.
[11] D.R. Kuhn, E.J. Coyne, and T.R. Weil, "Adding Attributes to Role-Based Access Control," IEEE Computer, vol. 43, no. 6, pp. 79-81,June 2010.
[12] M. Li, S. Yu, K. Ren, and W. Lou, "Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-Owner Settings," Proc. Sixth Int'l
ICST Conf. Security and Privacy in Comm.Networks (SecureComm),pp. 89-106, 2010.
[13] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation," Proc. ACM Symp. Information,
Computer and Comm. Security (ASIACCS), pp. 261-270, 2010.
[14] G. Wang, Q. Liu, and J. Wu, "Hierarchical Attribute-Based Encryption for Fine-Grained Access Control in Cloud Storage Services," Proc. 17th ACM Conf. Computer and Comm. Security
(CCS), pp. 735-737, 2010.
[15] F. Zhao, T. Nishide, and K. Sakurai, "Realizing Fine-Grained and Flexible Access Control to Outsourced Data with Attribute-Based Cryptosystems," Proc. Seventh Int'l Conf. Information SecurityPractice and Experience (ISPEC), pp. 83-97, 2011.
[16] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed AccessControl in Clouds," Proc. IEEE 10th Int'l Conf. Trust, Security andPrivacy in Computing and Communications (TrustCom), 2011.
[17] http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cs-01-en.pdf, 2013.
[18] http://securesoftwaredev.com/2012/08//xacml-in-the-cloud,2013.
[19] S. Jahid, P. Mittal, and N. Borisov, "EASiER: Encryption-BasedAccess Control in Social networks with Efficient Revocation,"Proc. ACM Symp. Information, Computer and Comm. Security
(ASIACCS), 2011.
[20] R.L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret,"Proc. Seventh Int'l Conf. Theory and Application of Cryptology andInformation Security (ASIACRYPT), pp. 552-565, 2001.
[21] X. Boyen, "Mesh Signatures," Proc. 26th Ann. Int'l Conf. Advancesin Cryptology EUROCRYPT), pp. 210-227, 2007.
[22] D. Chaum and E.V. Heyst, "Group signatures," Proc. Ann. Int'lConf. Advances in Cryptology (EUROCRYPT), pp. 257-265, 1991.
[23] H.K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-BasedSignatures: Achieving Attribute-Privacy and Collusion-Resistance," JIACR Cryptology ePrint Archive, 2008.
[24] H.K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-Based Signatures," Topics in Cryptology - CT-RSA, vol. 6558, pp. 376-392, 2011.
[25] A. Beimel, "Secure Schemes for Secret Sharing and Key Distribution, "PhD thesis, Technion, Haifa, 1996.
[26] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," Proc.Ann. Int'l Conf. Advances in Cryptology (EUROCRYPT), pp. 457-473,2005.
[27] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data,"Proc. ACM Conf. Computer and Comm. Security, pp. 89-98, 2006.
[28] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," Proc. IEEE Symp.Security and Privacy, pp. 321-334, 2007.
[29] X. Liang, Z. Cao, H. Lin, and D. Xing, "Provably Secure andEfficient Bounded Ciphertext Policy Attribute Based Encryption,"Proc. ACM Symp. Information, Computer and Comm. Security
(ASIACCS), pp 343-352, 2009.
[30] M. Chase, "Multi-Authority Attribute Based Encryption," Proc.Fourth Conf. Theory of Cryptography (TCC), pp. 515-534, 2007.