ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds

Ms.Priya Bhashini.K
M.Tech, Ph.D,
Department of Computer Science and Engineering,
Krishna Murthy Institute of Technology & Engineering, Ghatkesar, Hyderabad.

P.Mamatha
Department of Computer Science and Engineering,
Krishna Murthy Institute of Technology & Engineering, Ghatkesar, Hyderabad.

Abstract :

With cloud data services, it is commonplace for data to be not only stored in the cloud, but also shared across multiple users. Unfortunately, the integrity of cloud data is subject to skepticism due to the existence of hardware/ software failures and human errors. Several mechanisms have been designed to allow both data owners and public verifiers to efficiently audit cloud data integrity without retrieving the entire data from the cloud server. However, public auditing on the integrity of shared data with these existing mechanisms will inevitably reveal confidential information—identity privacy—to public verifiers. In this paper, we propose a novel privacy-preserving mechanism that supports public auditing on shared data stored in the cloud. In particular, we exploit ring signatures to compute verification metadata needed to audit the correctness of shared data. With our mechanism, the identity of the signer on each block in shared data is kept private from public verifiers, who are able to efficiently verify shared data integrity without retrieving the entire file. In addition, our mechanism is able to perform multiple auditing tasks simultaneously instead of verifying them one by one. Our experimental results demonstrate the effectiveness and efficiency of our mechanism when auditing shared data integrity.

Index Terms:

Access control, authentication, attribute-based signatures, attribute-based encryption, cloud storage.

1 INTRODUCTION:

Nowadays 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.The privacy protection and Security in clouds are being explored by many researchers. Wang et al, addressed storage security using Reed-Solomon erasure-correcting codes. Authentication of users using public key cryptographic techniques are studied in [5] and many homomorphic encryption techniques are suggested [6], [7] to ensure that the cloud is not able to read the data while performing computations on them.

With the help of homomorphic encryption, the cloud gets ciphertext of the data and performs computations on the ciphertext 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 user to ensure that the data returned from cloud is correct output.Take 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 does store the information in the cloud. Here only autherized users should access thje data. And is also manditory to know that the information comes from authenticate and reliable person. The problems of access control, authentication, and privacy protection should be solved simultaneously. As only authorized users have access to valid service, Access control in clouds is gaining attention.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A huge amount of sensitive information is being stored in the cloud, care should be taken to make sure 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 (introduced by Ferraiolo and Kuhn), 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 authentication, 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.Apart from storing the content in the cloud, it is also 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.

2 RELATED WORK:

In Attribute Based Encryption (ABE), a user has a set of attributes in addition to its unique ID. There are two classes of ABEs. In key-policy ABE or KP-ABE , the sender has an access policy to encrypt data. A writer whose attributes and keys have been revoked cannot write back stale information. The receiver receives attributes and secret keys from the attribute authority and is able to decrypt information if it has matching attributes. In Ciphertext-policy, CP-ABE, the receiver has the access policy in the form of a tree, with attributes as leaves and monotonic access structure with AND, OR and other threshold gates. 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 there are several KDC authorities (coordinated by a trusted authority) which distribute attributes and secret keys to users. Multiauthority ABE protocol required no trusted authority which requires every user to have attributes from at all the KDCs. Lewko and Waters 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 is inefficient when users access using their mobile devices. To get over this problem, Green et al. proposed to outsource the decryption task to a proxy server, so that the user can compute with minimum resources (for example, hand held devices). However, the presence of one proxy and one KDC makes it less robust than 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.

3 BACKGROUND:

Here we present our cloud storage model, adversary model and the assumptions we have made in the paper. The notations used throughout the paper is represented in table 1.

3.1 Assumptions:

We make the following assumptions in our work:
1.The cloud is honest-but-curious, which means that the cloud administrators can be interested in viewing user's content, but cannot modify it. This is a valid assumption that has been made in [12] and [13].

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Honest-but-curious model of adversaries do not tamper with data so that they can keep the system functioning normally and remain undetected.
2.Users can have either read or write or both accesses to a file stored in the cloud
3.All communications between users/clouds are secured by secure shell protocol, SSH.

3.2.Formats of Access Policies:

The format of Access policies can be in any of the following:

1) Boolean functions of attributes,
2) linear secret sharing scheme (LSSS) matrix, or
3) monotone span programs.

Any access structure can be converted into a Boolean function An example of a Boolean function is ((a1∧a2∧a3) ∨ (a4∧a5) ∨ (a6∧a7)) where a1, a2, a3, . . are attributes.

Let Y : {0, 1}n → (0, 1) be a monotone Boolean function. A monotone span program for Y over a field F is an l x t matrix M with entries in F, along with a labeling function a : [l] → [n] that associates each row of M with an input variable of Y, such that, for every (x1; x2 . . . ; xn) belongs to (0, 1)n, the following condition is satisfied:

$$\begin{aligned} \mathcal{Y}(x_1, x_2, \ldots, x_n) = 1 \Leftrightarrow \exists v \in \mathbb{F}^{1\times l} : vM \\ = [1,0,0,\ldots,0] \text{ and } (\forall i : x_{a(i)} \\ = 0 \Rightarrow v_i = 0). \end{aligned}$$

3.3.Attribute-Based Encryption:
3.3.1.System Initialization:

Select a prime q, generator g of G0, groups G0 and GT of order q, a map e : G0 × G0 → GT , and a hash function H : (0; 1)x → G0 that maps the identities of users to G0. The hash function used here is SHA-1. Each KDC Aj belongs to A has a set of attributes Lj. The attributes disjoint (Li ∩ Lj = ϕ_ for i ≠ j). Each KDC also chooses two random exponents ai; ,yi belongs to Zq. The secret key of KDC Aj is

$$SK[j] = \{\alpha_i, y_i, i \in L_j\}. \quad (1)$$

The public key of KDC $\mathcal{A}_j$ is published

$$PK[j] = \{e(g,g)^{\alpha_i}, g^{y_i}, i \in L_j\}. \quad (2)$$

TABLE 1
Notations

Symbols	Meanings
U_u	u-th User/Owner
$\mathcal{A}_j$	j-th KDC
$\mathbb{A}$	Set of KDCs
L_j	Set of attributes that KDC $\mathcal{A}_j$ possesses
$l_j = \lvert L_j \rvert$	Number of attributes that KDC $\mathcal{A}_j$ possesses
$I[j,u]$	Set of attributes that $\mathcal{A}_j$ gives to user U_u for encryption/decryption
I_u	Set of attributes that user U_u possesses
$J[j,u]$	Set of attributes that $\mathcal{A}_j$ gives to user U_u for claim attributes
J_u	Set of attributes that user U_u possesses as claim attributes
$AT[j]$	KDC which has attribute j
$PK[j]/SK[j]$	Public key/secret key of KDC $\mathcal{A}_j$ for encryption/decryption
$sk_{i,u}$	Secret key given by $\mathcal{A}_j$ corresponding to attribute i given to user U_u
TPK/PSK	Trustee public key/secret key
$APK[j]/ASK[j]$	Public key/secret key of KDC $\mathcal{A}_j$ for verifying claim
$\mathcal{X}$	Boolean access structure
$\mathcal{Y}$	Claim policy
τ	Time instant
R	Access matrix of dimension $m \times h$
M	Matrix of dimension $l \times t$ corresponding to the claim predicate
MSG	Message
$\lvert MSG \rvert$	Size of message MSG
C	Ciphertext
$H, \mathcal{H}$	Hash functions, example SHA-1

3.3.2.Key Generation and Distribution by KDCs:

User Uu receives a set of attributes I[j, u] from KDC Aj, and corresponding secret key ski,u for each i belongs to I[j, u].

$$sk_{i,u} = g^{\alpha_i} H(u)^{y_i}, \quad (3)$$

where $\alpha_i, y_i \in SK[j]$.

3.3.3.Encryption by Sender:

The encryption function is ABE:Encrypt(MSG, X). Sender encrypts message MSG as follows:

1.Choose a random seed s belongs to Zq and a random vector v belongs to Zhq' with s as its first entry h is the number of leaves in the access tree (equal to the number of rows in the corresponding matrix R).

2. Calculate λx = Rx. v, where Rx is a row of R.
3. Choose a random vector $w \in \mathbb{Z}_q^h$ with 0 as the first entry.
4. Calculate $\omega_x = R_x \cdot w$.
5. For each row Rx of R, choose a random $\rho_x \in \mathbb{Z}_q$.
6. The following parameters are calculated:

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

$$
\begin{aligned}
C_0 &= MSGe(g,g)^s, \\
C_{1,x} &= e(g,g)^{\lambda_x} e(g,g)^{\alpha_{\pi(x)}\rho_x}, \forall x, \\
C_{2,x} &= g^{\rho_x} \forall x, \\
C_{3,x} &= g^{y_{\pi(x)}\rho_x} g^{\omega_x} \forall x,
\end{aligned} \qquad (4)
$$

where ∏(x) is mapping from Rx to the attribute i that is located at the corresponding leaf of the access tree The ciphertext C is sent by the sender (it also

7.includes the access tree via R matrix):

$$C = \langle R, \pi, C_0, \{C_{1,x}, C_{2,x}, C_{3,x}, \forall x\}\rangle. \qquad (5)$$

3.3.4.Decryption by Receiver:

The decrypt function is ABE:Decrypt(C, { ski,u }). Where C is given by (5). Receiver Uu takes as input ciphertext C, secret keys { ski,u } group G0, and outputs message msg. It obtains the access matrix R and mapping ∏ from C. It then executes the following steps:

1. **Uu** calculates the set of attributes $\{\pi(x) : x \in X\} \bigcap I_u$ that are common to itself and the access matrix. X is the set of rows of R.
2. For each of these attributes, it checks if there is a subset X' of rows of R, such that the vector (1, 0 . . . 0) is their linear combination. If not, decryption is impossible. If yes, it calculates constants $c_x \in \mathbb{Z}_q$, such that $\sum_{x \in X'} c_x R_x = (1, 0, \ldots, 0)$.
3. Decryption proceeds as follows:

a. For each $x \in X'$, $dec(x) = \frac{C_{1,x} e(H(u), C_{3,x})}{e(sk_{\pi(x),u}, C_{2,x})}$.

b. U_u computes $MSG = C_0 / \Pi_{x \in X'} dec(x)$.

4 PROPOSED PRIVACY PRESERVING AUTHENTICATED ACCESS CONTROL SCHEME

Here we propose privacy preserving authenticated access control scheme, According to our scheme a user can create a file and store it securely in the cloud. This scheme consists of use of the two protocols ABE and ABS. In Fig. 1, there are three users, a creator, a reader, and writer. Creator Alice receives a token ɣ 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 ɣ. 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 KDCs receives 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 claim policy 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 the ciphertext C. When a reader wants to read, the cloud sends C. If the user has attributes matching with access policy, it can decrypt and get back original message.

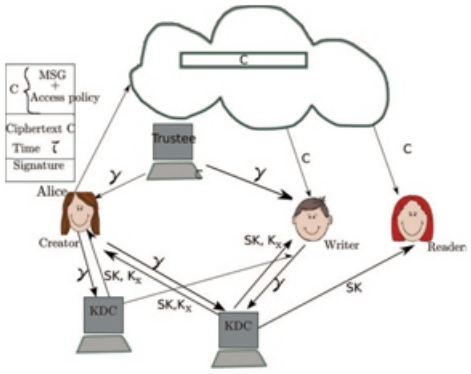

Fig. 1. Our secure cloud storage model.

4.1 Writing to the Cloud:

To write to an already existing file, the user must send its message 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.

4.2 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, {ski,u}) and the message MSG is calculated .

4.4 User Revocation:

Herer, we described 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.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

For this reason, the owners should change the stored data and send updated information to other users. The set of attributes Iu possessed by the revoked user Uu is noted and all users change their stored data that have attributes . In [13], revocation involved changing the public and secret keys of the minimal set of attributes which are required to decrypt the 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 $s, s_{new} \in \mathbb{Z}_q$ is selected.
2. The first entry of vector v_{new} is changed to new s_{new}.
3. $\lambda_x = R_x v_{new}$ is calculated , for each row x, corresponding to leaf attribute in $\mathbf{I_u}$
4. $\mathbf{C_{1,x}}$ is calculated for x.
5. New value of C1;x is securely transmitted to the cloud
6. New $C_0 = Me(g,g)^{s_{new}}$ 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.

5 REAL LIFE EXAMPLE:

The problem, that we left in introduction, is here. Suppose Alice is a law student and wants to send a series of reports about malpractices by authorities of University X to all the professors of University X, Research chairs of universities X; Y ;Z and students belonging to Law department in university X. She wants to remain anonymous, while publishing all evidence. All information is stored in the cloud. It is important that users should not be able to know her identity, but must trust that the information is from avalid source. For this reason she also sends a claim message which states that she "Is a law student" or "Is a student counselor" or "Professor at university X." The tree corresponding to the claim policy is shown in Fig. 2. The leaves of the tree consists of attributes and the intermediary nodes consists of Boolean operators. In this example the attributes are "Student," "Prof," "Dept Law," "Uni X," "Counselor." The above claim policy can be written as a Boolean function of attributes as((Student AND Dept Law) OR (Prof AND Uni X)) OR (Student Counselor).

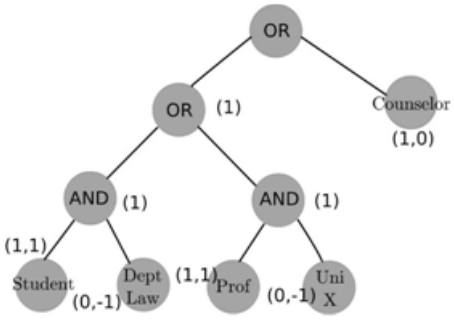

Fig. 2. Example of claim policy

6 SECURITY OF THE PROTOCOL:

Theorem : 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. A user's token is

$\gamma = (u, K_{base}, K_0, \rho)$, Where ρ is signature on $u \| K_{base}$ with $TSig$ belonging to the trustee. An invalid user with a different user-id cannot create the same signature because it does not know $\mathbf{TS_{ig}}$.

TABLE 2
Notations

Symbols	Computation
E_x	Exponentiation in group G_x
τ_H	Time to hash using function H
$\tau_{\mathcal{H}}$	Time to hash using function $\mathcal{H}$
$\tau_P/\tau_{\hat{P}}$	Time taken to perform 1 pairing operation in $e/\hat{e}$
$\|G\|$	Size of group G
a	Number of KDCs which contribute keys to user

7.COMPARISON WITH OTHER ACCESS CONTROL SCHEMES IN CLOUD

We compare our scheme with other access control schemes (in Table 3) and show that our scheme supports many features that the other schemes did not

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

TABLE 3
Comparison of Our Scheme with Existing Access Control Schemes

Schemes	Fine-grained access control	Centralized/ Decentralized	Write/read access	Type of access control	Privacy preserving authentication	User revocation?
[38]	Yes	Centralized	1-W-M-R	Symmetric key cryptography	No authentication	No
[12]	Yes	Centralized	1-W-M-R	ABE	No authentication	No
[13]	Yes	Centralized	1-W-M-R	ABE	No authentication	No
[16]	Yes	Decentralized	1-W-M-R	ABE	No authentication	Yes
[33]	Yes	Centralized	1-W-M-R	ABE	No authentication	No
[34]	Yes	Decentralized	1-W-M-R	ABE	Not privacy preserving	Yes
[15]	Yes	Centralized	M-W-M-R	ABE	Authentication	No
Ours	Yes	Decentralized	M-W-M-R	ABE	Authentication	Yes

TABLE 4
Comparison of Computation and Size of Ciphertext While Creating a File

Schemes	Computation by creator	Computation by cloud	Size of ciphertext
[12]	$(m+2)E_0$	0	$m\log\|G_0\| + \|G_T\| + m\log m + \|MSG\|$
[13]	$(m+2)E_0$	0	$m\log\|G_0\| + \|G_1\| + \|MSG\|$
[16]	$(3m+1)E_0 + 2mE_T + \tau_P$ (encrypt)	0	$2m\|G_0\| + m\|G_T\| + m^2 + \|MSG\|$
[33]	$(2m+1)E_0 + E_T + \tau_P$ (encrypt)	$mE_0 + mE_T + (m+1)T_{\hat{p}}$	$(2m+1)\|G_0\| + \|G_T\| + m^2 + \|MSG\|$
[34]	$(4m+1)E_0 + 2aE_T + \tau_P$ (encrypt)	$3maE_T + (3ma+1)\tau_P$	$(3m+1)\|G_0\| + \|G_T\| + m^2 + \|MSG\|$
[15]	$E_1 + (2m+1)E_0 + m\tau_H$ (encrypt) $(2l+2)E_1 + 2tE_2 + \tau_{\mathcal{H}}$ (sign)	$(l+2t)\tau_{\hat{p}} + l(E_1+E_2) + \tau_{\mathcal{H}}$ (verify)	$\|G_2\| + (2m+1)\|G_1\| + \|MSG\|$ $+(l+t+2)\|G_1\| + m^2$
Our approach	$(3m+1)E_0 + 2mE_T + \tau_P$ (encrypt) $(2l+2)E_1 + 2tE_2 + \tau_{\mathcal{H}}$ (sign)	$(l+2t)\tau_{\hat{p}} + l(E_1+E_2) + \tau_{\mathcal{H}}$ (verify)	$2m\|G_0\| + m\|G_T\| + m^2 + \|MSG\|$ $+(l+t+2)\|G_1\|$

TABLE 5
Comparison of Computation during Read and Write by User and Cloud

Schemes	Computation by user while write	Computation by user while read	Computation by cloud while write
[12]	No write access	$m\tau_P$	No write access
[13]	No write access	$m\tau_P$	No write access
[16]	No write access	$2m\tau_P + \tau_H + O(mh)$	No write access
[33]	No write access	$E_0 + \tau_H + O(mh)$	No write access
[34]	No write access	$E_0 + \tau_H + O(mh)$	No write access
[15]	$E_1 + (2m+1)E_0 + m\tau_H$ (encrypt) $(2l+2)E_1 + 2tE_2 + \tau_{\mathcal{H}}$ (sign)	$(2m+1)\tau_P$ (decrypt)	$(l+2t)\tau_{\hat{p}} + l(E_1+E_2) + \tau_{\mathcal{H}}$ (verify)
Our approach	$(3m+1)E_0 + 2mE_T + \tau_P$ (encrypt) $(2l+2)E_1 + 2tE_2 + \tau_{\mathcal{H}}$ (sign)	$2m\tau_P + \tau_H + O(mh)$ (decrypt)	$(l+2t)\tau_{\hat{p}} + l(E_1+E_2) + \tau_{\mathcal{H}}$ (verify)

support. 1-W-M-R means that only one user can write while many users can read. M-W-M-R means that many users can write and read. We see that most schemes do not support many writes which is supported by our scheme. Our scheme is robust and decentralized, most of the others are centralized. Our scheme also supports privacy preserving authentication, which is not supported by others. Most of the schemes do not support user revocation, which our scheme does.

In Tables 4 and 5, we compare the computation and communication costs incurred by the users and clouds and show that our distributed approach has comparable costs to centralized approaches. The most expensive operations involving pairings and is done by the cloud. If we compare the computation load of user during read we see that our scheme has comparable costs.

8 CONCLUSION:

In this paper, 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.

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.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[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, "Fuzzy Keyword Search Over Encrypted Data in Cloud Computing," Proc. IEEE INFOCOM, pp. 441-445, 2010.

[4] S. Kamara and K. Lauter, "Cryptographic Cloud Storage," roc. 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," PhD dissertation, Stanford Univ., http://www.crypto.stanford.edu/ craig, 2009.

[7] A.-R. Sadeghi, T. Schneider, and M. Winandy, "Token-Based Cloud Computing," Proc. Third Int'l Conf. Trust and Trustworthy Computing (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 Framework for Accountability and Trust in Cloud Computing," HP Technical Report 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: The Essential of Bread and Butter of Data Forensics in Cloud Computing," Proc. Fifth ACM Symp. Information, Computer and Comm. 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] [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.