Attribute based Encryption and Key Distribution for Secure Storage in Clouds

[1]R.Vaishali , [2] M.Menaka
[1]Student, 2Assistant Professor
Kingston Engineering College, Vellore (India)

***Abstract*- A new decentralized access control scheme is used for secure data storage in clouds that supports anonymous authentication. According to this scheme a user can create a file and store it securely in the cloud. Decryption will only work if the attributes associated with the decryption key match the policy used to encrypt the message. The cloud verifies the authenticity of the users 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. This scheme is resilient to replay attacks and supports creation, modification and reading data stored in the cloud. The proposed scheme is resilient to replay attacks. In this scheme Secure Hash algorithm is used for authentication purpose, SHA is the one of several cryptographic hash functions, most often used to verify that a file has been unaltered. The Paillier cryptosystem is a probabilistic asymmetric algorithm for public key cryptography. Paillier algorithm is used for creation of access policy, file accessing and file restoring process.**

***Index Terms*- Access Control, Authentication, Secure Hash Algorithm, Paillier Algorithm, Replay Attacks**

I. INTRODUCTION

The mainstay of this is to propose a new decentralized access control scheme for secure data storage in clouds that supports anonymous authentication. The proposed scheme is resilient to replay attacks. A writer whose attributes and keys have been revoked cannot write back stale information. Distributed access control of data stored in cloud so that only authorized users with valid attributes can access them. Authentication of users is also verified who store and modify their data on the cloud. The identity of the user is protected from the cloud during authentication. The architecture is decentralized, meaning that there can be several KDCs for key management. The access control and authentication are both collusion resistant, meaning that no two users can collude and access data or authenticate themselves, if they are individually not authorized. Revoked users cannot access data after they have been revoked. The proposed scheme is resilient to replay attacks. A writer whose attributes and keys have been revoked cannot write back stale information. The protocol supports multiple read and writes on the data stored in the cloud. The costs are comparable to the existing centralized approaches and the expensive operations are mostly done by the cloud. Proposing 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. The cloud verifies the authenticity of the user without knowing the user's identity before storing data. The 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.

II. PROPOSED WORK

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. The identity of the user is protected from the cloud during authentication.
3. The architecture is decentralized, meaning that there can be several KDCs for key management.
4. The access control and authentication are both collusion resistant, meaning that no two users can collude and access data or authenticate themselves, if they are individually not authorized.
5. Revoked users cannot access data after they have been revoked.
6. The proposed scheme is resilient to replay attacks. A writer whose attributes and keys have been revoked cannot write back stale information.
7. This protocol supports multiple read and writes on the data stored in the cloud.
8. The costs are comparable to the existing centralized approaches and the expensive operations are mostly done by the cloud.

III. PROPOSED PRIVACY PRESERVING AUTHENTICATED ACCESS CONTROL SCHEME

According to this scheme users can create a file and store it securely in the cloud. This scheme consists of use of the two protocols ABE and ABS. 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, 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. 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. Write proceeds in the same way as file creation. By designating the verification process to the cloud, it relieves the 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. SKs are secret keys given for decryption, K_x are keys for signing. The message MSG is encrypted under the access policy. 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 cipher text C with signature is c and is sent to the cloud. The cloud verifies the signature and stores the cipher text 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. Write proceeds in the same way as file creation. By designating the verification process to the cloud, it relieves the 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 key distribution center (KDC). If it has enough attributes matching with the access policy, then it decrypts the information stored in the cloud.

IV. PAILLIER ALGORITHM

The Paillier cryptosystem is a probabilistic asymmetric algorithm for public key cryptography. This scheme is an additive homomorphic cryptosystem; given only the public-key and the encryption of m1 and m2, the encryption of m1+m2 is computed. This converts an alphanumeric message into a purely numeric message, which can be broken into blocks, m_i, such that, for each i, $0 < mi < n$, for a predetermined value, n. It provides semantic security against chosen-plaintext attacks. Self-blinding is ability to change one cipher text into another without changing the content of its decryption.

Key generation

- Choose two large prime number p and q randomly and independently of each other such that gcd (pq,(p-1)(q-1))=1
- Compute n=pq and λ=lcm(p-1,q-1)
- Select random integer g where $g \in Z^*_{n2}$
- Compute $\mu = (L(g^{\lambda} \bmod n^2))^{-1} \bmod n$
- public (encryption) key is (n,g)
- private (decryption) key is (λ,μ)

Encryption

Let m be a message to be encrypted where m belongs to Z^*_n.Select random where $r \in Z^*_{n..}$Compute cipher text as: $\mathbf{c = g^m .r^n \bmod n^2}$

Decryption

Cipher text: c belongs to Z^*_{n2}.Compute message: $\mathbf{m = L(c^{\lambda} \bmod n^2).\ \mu \bmod n}$

4.1 SECURE HASH ALGORITHM

SHA-1 is one of several cryptographic hash functions, most often used to verify that a file has been unaltered. SHA is short for Secure Hash Algorithm. File verification using SHA-1 is accomplished by comparing the checksums created after running the algorithm on the two files need to be compared. SHA-1 is the second iteration of this cryptographic hash function, replacing the previous SHA-0. In the SHA-1 iteration A, B, C, D and E are 32bit words of the state. F is a nonlinear function that varies.n denotes a left bit rotation by n places. n varies for each operation. W_t is the expanded message word of round t. K_t is the round constant of round t. F denotes addition modulo 232.

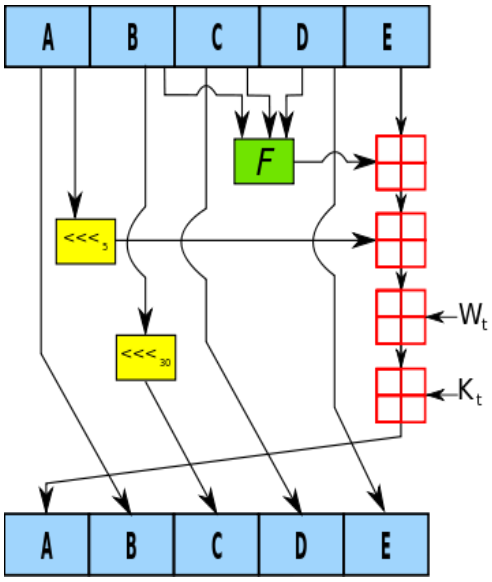

Figure 1 SHA Round Function

V. FUNCTIONAL MODULES

5.1 Creation of KDC

Different numbers of KDCs are created to register the user details.KDC name; KDC id and KDC password are given as input to create each KDC. Inputs will be saved in the database and the new users are registered first in the KDC by providing details such as username, user id and type are given as inputs. The user will enroll their personal details to KDC. KDC will verify the user details and it will save it in the database.

5.1.1 KDC Authentication

After KDC gives a user id to the user, the new user will enroll their personal details to the database by giving inputs such as user name, user id, password, mail id, phone number, university, type, etc. The key distribution center will verify the user details and if the user details are valid, their details will be stored in the database. The Key distribution center mainly verifies the user type and university name with its database to authenticate the users. Each key distribution center has a set of attributes Lj.

5.2Trustee and User Accessibility

Users receive 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 to the trustee, trustee gives a token. There are multiple KDCs, which can be scattered. Users on presenting the token to KDC receive keys for encryption/decryption and signing. SK are secret keys given for decryption, Kx are keys for signing. User can login with their credentials and request the token from trustee for the file upload using the user id. After the user id is received by the trustee, trustee will create token using user id, key and user signature (SHA).Then the trustee will issue a token to the particular user and then trustee will be able to view the logs.

5.3 Data Storage in Clouds

User on receiving the token from the trustee presents the token to the KDC. Then the token is verified by the KDC if the user credentials are valid, KDC will provide the public and Private Key to the user. After users receive their keys, they can encrypt the files with the public keys and set their access policies (privileges). The message is encrypted under the access policies. The access policies decide who can access the data stored in the cloud. The cipher text C with signature is c and is sent to the cloud. The cloud verifies the signature and stores the cipher text C. A user first registers itself with one or more trustees.

5.4 File Accessing

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. When a user requests data from the cloud, the cloud sends the cipher text C. Write proceeds in the same way as file creation. 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. The current time stamp T is attached to the cipher text to prevent replay attacks.

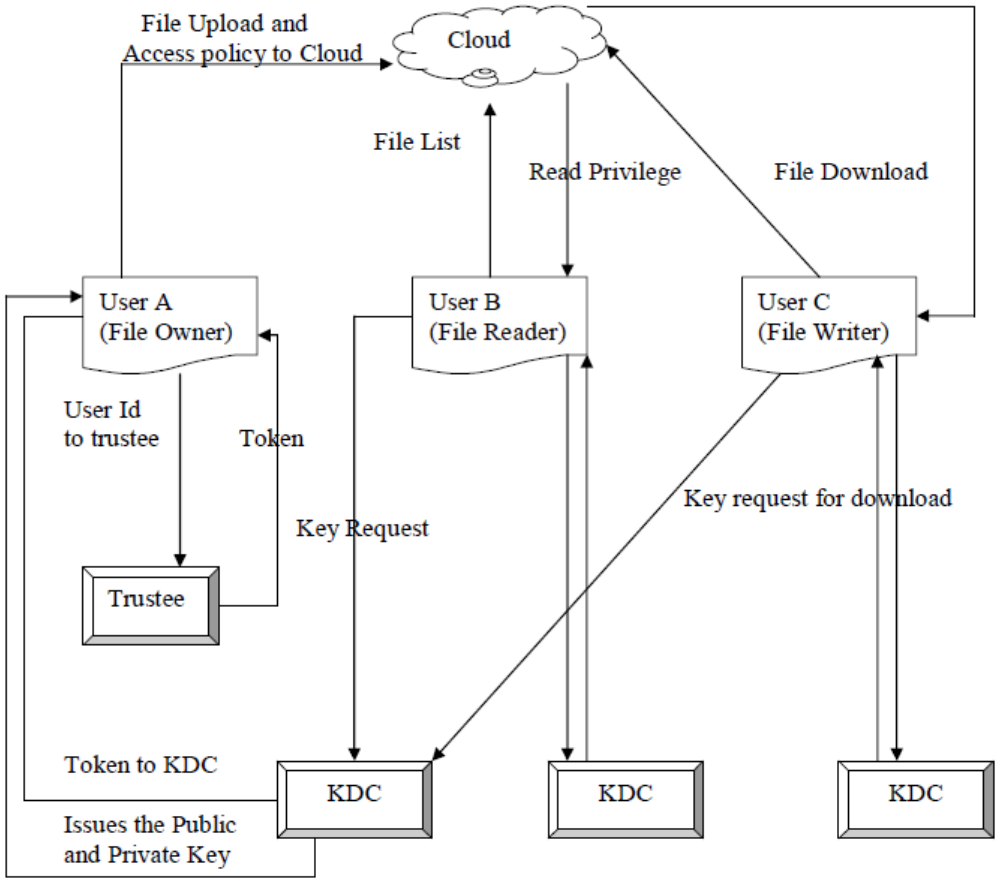

Fig 2: Secure cloud storage model

VI. COMPARISON WITH OTHER ACCESS CONTROL SCHEMES

On comparing the proposed scheme with other access control schemes and it seems that proposed scheme supports many features that the other schemes did not 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. Most schemes do not support many writes which is supported by this scheme. This scheme is robust and decentralized; most of the others are centralized. This also supports privacy preserving authentication, which is not supported by others. Most of the schemes do not support user revocation, which our proposed scheme does.

Table 1 Comparison of proposed scheme with existing access control scheme

Schemes	**Fine-grained access control**	**Centralized/Decentralized**	**Write/ read access**	**Type of access control**	**Privacy preserving authentication**	**User revocation**
Secure and efficient access	Yes	Centralized	1-W-M-R	Symmetric key cryptography	No authentication	No
Fine grained access control	Yes	Centralized	1-W-M-R	ABE	No authentication	No
Attribute based data sharing	Yes	Centralized	1-W-M-R	ABE	No authentication	No
proposed scheme	Yes	Decentralized	M-W-M-R	ABE	Authentication	Yes

VII. CONCLUSION

Thus the proposed decentralized access control technique supports 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. Moreover, this authentication and access control scheme is decentralized and robust, unlike other access control schemes designed for clouds which are centralized. One limitation is that the cloud knows the access policy for each record stored in the cloud. In future, SQL queries are used for hiding the attributes and access policy of a user. Files stored in cloud can be corrupted. So for this issue using the file recovery technique is used to recover the corrupted file and to hide the access policy and the user attributes.

REFERENCES

[1] S. Ruj, M. Stojmenovic and A. Nayak, “Privacy Preserving Access Control with Authentication for Securing Data in Clouds”, IEEE/ACM International Symposium on 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 T. Services Computing, vol. 5, no. 2, pp. 220–232, 2012.
[3] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, “Fuzzy keyword search over encrypted data in cloud computing,” in IEEE INFOCOM. , pp. 441–445, 2010.
[4] S.Kamara and K.Lauter, “Cryptographic cloud storage,” in Financial Cryptography Workshops, ser.Lecture Notes in Computer Science, vol. 6054. Springer, pp. 136–149, 2010.
[5] H. Li, Y. Dai, L. Tian, and H. Yang, “Identity-based authentication for cloud computing,” in Cloud Com, ser. Lecture Notes in Computer Science, vol. 5931. Springer, pp. 157–166, 2009.
[6] C. Gentry, “A fully homomorphic encryption scheme,” Ph.D. dissertation, Stanford University, 2009, http://www.crypto.stanford.edu/craig.
[7] A.-R. Sadeghi, T. Schneider, and M. Winandy, “Token-based cloud computing,” in TRUST, ser. Lecture Notes in Computer Science, vol. 6101. Springer, pp. 417–429, 2010.
[8] R. K. L. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, and B. S. Lee, “Trust cloud: A framework for accountability and trust in cloud computing,” HP Technical Report HPL-2011-38. Available at http://www.hpl.hp.com/techreports/2011/HPL-2011-38.html.
[9] R. Lu, X. Lin, X. Liang, and X. Shen, “Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing,” in ACM ASIACCS, pp. 282–292, 2010.
[10] D. F. Ferraiolo and D. R. Kuhn, “Role-based access controls,” in 15th National Computer Security Conference, 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, 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 multiowner settings,” in SecureComm, pp. 89–106, 2010.
[13] S. Yu, C. Wang, K. Ren, and W. Lou, “Attribute based data sharing with attribute revocation,” in ACM 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,” in ACM 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,” in ISPEC, ser. Lecture Notes in Computer Science, vol. 6672. Springer, pp. 83–97, 2011.