Attribute Based Encryption with Privacy Preserving In Clouds

M. Suriyapriya[1], A. Joicy[2]
PG Scholar[1]
Assistant Professor CSE Department[2]
St.Joseph College of Engineering
Sriperumbudur, Chennai-602105
suriyamathaiyan22@gmail.com, joylouis.90@gmail.com

***Abstract*—** Security and privacy are very important issues in cloud computing. In existing system access control in clouds are centralized in nature. The scheme uses a symmetric key approach and does not support authentication. Symmetric key algorithm uses same key for both encryption and decryption. The authors take a centralized approach where a single key distribution center (KDC) distributes secret keys and attributes to all users. A new decentralized access *control* scheme for secure data storage in clouds that supports anonymous authentication. The validity of the user who stores the data is also verified. The proposed scheme is resilient to replay attacks. In this scheme using Secure Hash algorithm 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 crypto system, is a probabilistic asymmetric algorithm for public key cryptography. Pailier algorithm use for Creation of access policy, file accessing and file restoring process.

Keywords*—Access control, Authentication, Secure hash algorithm, pailler algorithm.*

*****

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 A writer whose attributes and keys have been revoked cannot write back stale information.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 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 Revoked users cannot access data after they have been revoked.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 A writer whose attributes and keys have been revoked cannot write back stale information.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. RELATED WORK

ABE was proposed by Sahai and Waters [26]. In 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 (Goyal *et al.* [27]), 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 Cipher text-policy, CP-ABE ([28], [29]), 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 [30] proposed a multi-authority ABE, in which there are several KDC authorities (coordinated by a trusted authority) which distribute attributes and secret keys to users. Multi-authority ABE protocol was studied in [31], [32], which required no trusted authority which requires every 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. [33] 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 key distribution center makes it less robust than decentralized approaches. Both these approaches had no way to authenticate users, anonymously. Yang *et al.* [34] presented a modification of [33], authenticate users, who want to remain anonymous while accessing the cloud.

To ensure anonymous user authentication Attribute Based Signatures were introduced by Maji *et al.* [23]. This was also a centralized approach. A recent scheme by the same authors [24] takes a decentralized approach and provides authentication without disclosing the identity of the users. However, as mentioned earlier in the previous section it is prone to replay attack.

III. PROPOSED WORK

The main contributions of this paper are the following:

- Distributed access control of data stored in cloud so that only authorized users with valid attributes can access them.
- 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 write 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.

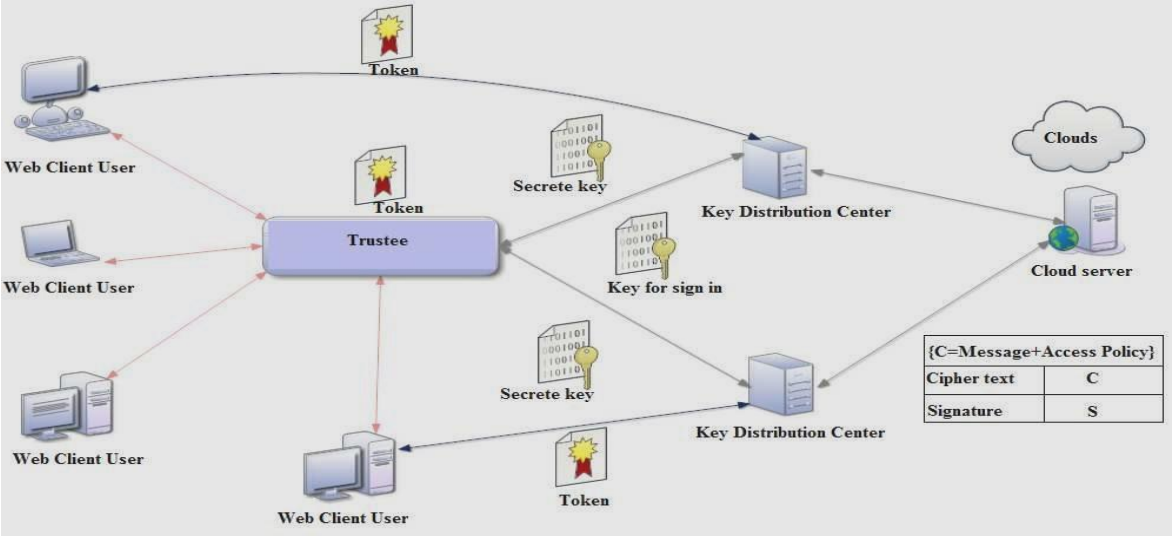

Fig 1: Secure Cloud storage model

- Authentication of users 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 KDC's for key management. There are three users, a creator, a reader and writer. Creator Alice receives a token γ 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 the trustee gives her a token γ.. 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. 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.

A. Creation of KDC

Different number of KDC's are created and to register a user details. KDC name, KDC id and KDC password are given as input to create KDC. Inputs will save in a database and to register a user details given a input as username and user id.

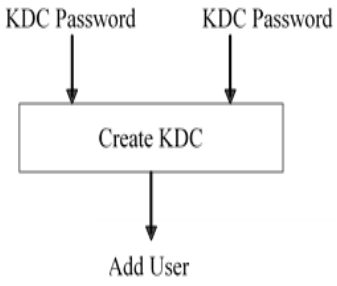

Fig 2: Creation of KDC

B. KDC Authentication

After KDC given a user id to a user, the user will enrolled the personal details to KDC's given a input as user name, user id, password etc. The KDC will be verify the user details and it will insert it in a Database.

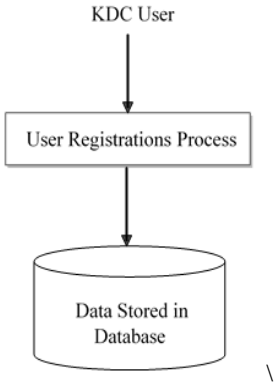

Fig 3: KDC Authentication

C. Trustee and User Accessibility

Users can get the token from trustee for the file upload. After trustee was issuing a token, trustee can view the logs. User can login with their credentials and request the token from trustee for the file upload using the user id. After the user id received by the trustee, trustee will be create token using user id, key and user signature (SHA).

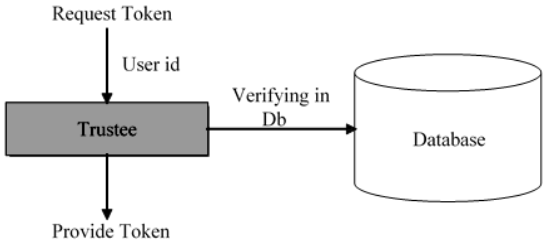

Fig 4: User Accessibility

D. Creation of access policy

After the key was received by the User, the message MSG 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. 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 and user can upload the file after user get key from the KDC.

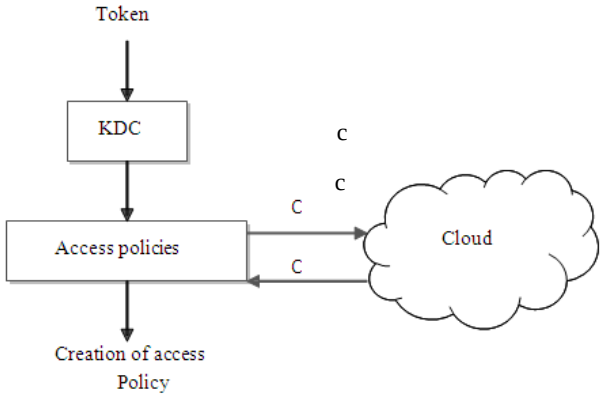

Fig 5: Creation of access policy

E. File accessing

Using their access policies the users can download their files by the help of kdc's to issue the private keys for the particular users. After trustee token issuance for the users, the users produce the token to the KDC then the token verify by the KDC if it is valid then KDC will provide the public and Private key to the user. After users received the

keys the files are encrypt with the public keys and set their Access policies (privileges).

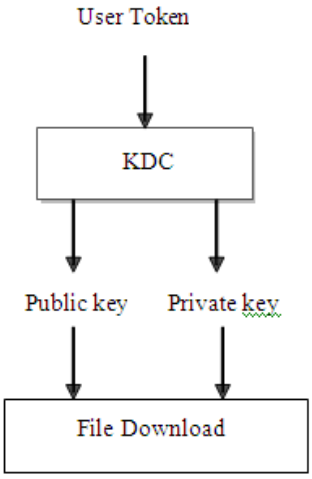

Fig 6: File accessing

F. File Restoration

Files stored in cloud can be corrupted. So for this issue, using the file recovery technique to recover the corrupted file successfully and to hide the access policy and the user attributes.

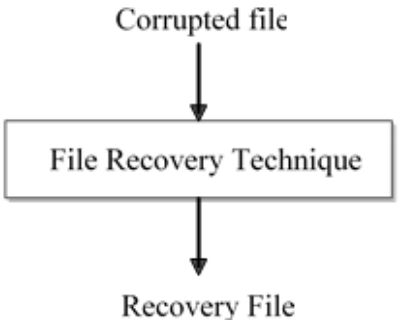

Fig 7: File Restoration

G. Secure Hash Algorithm

Definition: 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 you want to compare. SHA-1 is the second iteration of this cryptographic hash function, replacing the previous SHA-0. An SHA-2 cryptographic hash function is also available and SHA-3 is being developed.

One iteration within the SHA-1 compression function. A, B, C, D and E are 32bit words of the state. *F* is a nonlinear function that varies. $\lll_n$ 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.$\boxplus$ denotes addition modulo 2^{32}.

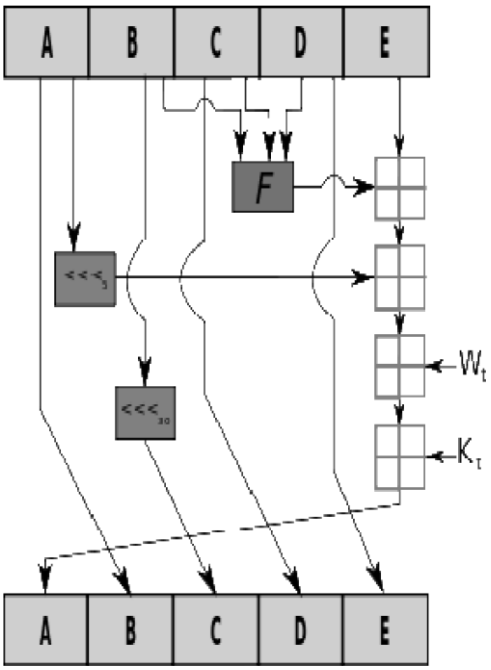

Fig 8: Secure Hash Algorithm

H. Paillier Algorithm

The Paillier cryptosystem, named after and invented by Pascal Paillier is a probabilistic asymmetric algorithm for public key cryptography.

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. This property is assured if both primes are of equivalent length, i.e p,q $\{0,1\}^{s-1}$ for security parameter S . Compute n=pq and λ=lcm(p-1,q-1).

Select random integer g where $gZ^*_{n^2}$. Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse .μ= $(L(g^{\lambda} \bmod n^2))^{-1} \bmod n$, where function L is defind as The public (encryption) key is (n,g).. The private (decryption) key is (λ,μ).

Encryption

Let m be a message to be encrypted where m Z_n. Select random r where r Z^*_n. Compute cipher text as: c= g^m .r^n mod n^2

Decryption

Cipher text: c$Z^*_{n^2}$.Compute message: m =L(c^{λ} mod n^2). M mod n

IV. CONCULSION

A decentralized access control technique with anonymous authentication, which provides user revocation and prevents replay attacks, is achieved. 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 and also hide the attributes and access policy of a user. One limitation is that the cloud knows the access policy for each record stored in the cloud. In future, using SQL queries for hide the attributes and access policy of a user. Files stored in cloud can be corrupted. So for this issue using the file recovery technique to recover the corrupted file successfully and to hide the access policy and the user attributes.

V. 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 *CloudCom*, 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.

[16] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed access control in clouds," in *IEEE TrustCom*, 2011.

[17] http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cs-01-en.pdf.

[18] http://securesoftwaredev.com/2012/08/20/xacml-in-the-cloud.

[19] S. Jahid, P. Mittal, and N. Borisov, "EASiER: Encryption-based access control in social networks with efficient revocation," in *ACM ASIACCS*, 2011.

[20] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *ASIACRYPT*, ser. Lecture Notes in Computer Science, vol. 2248. Springer, pp. 552–565, 2001.

[21] X. Boyen, "Mesh signatures," in *EUROCRYPT*, ser. Lecture Notes in Computer Science, vol. 4515. Springer, pp. 210–227, 2007.

[22] D. Chaum and E. van Heyst, "Group signatures," in *EUROCRYPT*, pp. 257– 265, 1991.

[23] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures: Achieving attribute-privacy and collusion-resistance," *IACR Cryptology ePrint Archive*, 2008.

[24] "Attribute-based signatures," in *CT-RSA*, ser. Lecture Notes in Computer Science, vol. 6558. Springer, pp. 376–392, 2011.

[25] A. Beimel, "Secure Schemes for Secret Sharing and Key Distribution," Ph D Thesis. Technion, Haifa, 1996.

[26] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *EUROCRYPT*, ser. Lecture Notes in Computer Science, vol. 3494. Springer, 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," in *ACM Conference on Computer and Communications Security*, pp. 89–98, 2006.

[28] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *IEEE Symposium on Security and Privacy*. , pp. 321–334, 2007.

[29] X. Liang, Z. Cao, H. Lin and D. Xing, "Provably Secure and Efficient Bounded Ciphertext Policy Attribute Based Encryption," in *ACM ASIACCS*, pp 343–352, 2009.

[30] M. Chase, "Multi-authority attribute based encryption," in *TCC*, ser. Lecture Notes in Computer Science, vol. 4392. Springer, pp. 515–534, 2007.

[31] H. Lin, Z. Cao, X. Liang and J. Shao, "Secure Threshold Multi-authority Attribute Based Encryption without a Central Authority," in *INDOCRYPT*, ser. Lecture Notes in Computer Science, vol. 5365, Springer, pp. 426–436, 2008.

[32] M. Chase and S. S. M. Chow, "Improving privacy and security in multiauthority attribute-based encryption," in *ACM Conference on Computer and Communications Security*, pp. 121–130, 2009.

[33] Matthew Green, Susan Hohenberger and Brent Waters, "Outsourcing the Decryption of ABE Ciphertexts," in USENIX Security Symposium, 2011.

[34] Kan Yang, Xiaohua Jia and Kui Ren, " DAC-MACS: Effective Data Access Control for Multi-Authority Cloud Storage Systems", IACR Cryptology ePrint Archive, 419, 2012.

[35] A. B. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *EUROCRYPT*, ser. Lecture Notes in Computer