ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Attribute Based Encryption with Privacy Protection in Clouds

Geetanjali. M[1], Saravanan. N[2]

PG Student, Department of Information Technology, K.S.R College of Engineering, Tiruchengode, Tamilnadu, India [1]

Guide, Department of Information Technology, K.S.R College of Engineering, Tiruchengode, Tamilnadu, India[2]

ABSTRACT- I propose a new decentralized access control scheme for secure data storage in clouds, which supports anonymous authentication. In the proposed scheme, the cloud verifies the authenticity of the user without knowing the user's identity before storing data. Decentralized scheme also has the added feature of access control in which only valid users are able to decrypt the stored information. This scheme prevents replay attacks and supports creation, modification, and reading data stored in the cloud and also address user revocation. Moreover, this 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.

I.INTRODUCTION

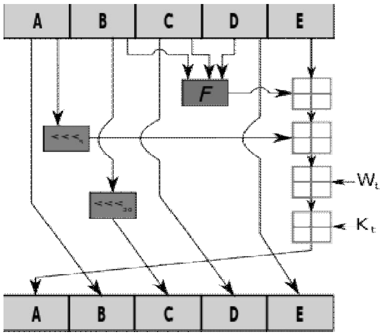
The mainstay of this is to propose a new decentralized access control scheme for secure data storage in clouds that supports anonymous authentication. The proposed scheme is resilient to replay attacks. A writer whose attributes and keys have been revoked cannot write back stale information. Distributed 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.MPLEMENTATION TECHNIQUES

A.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 32-bit 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. ⊞ denotes addition modulo 2^{32}.



B.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 numbers 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., $p,q \in 1||\{0,1\}^{s-1}$ for security parameter s.

- Compute $n = pq$ and $\lambda = \operatorname{lcm}(p-1,q-1)$.
- Select random integer g where $g \in \mathbb{Z}^*_{n^2}$
- Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse: $\mu = (L(g^\lambda \mod n^2))^{-1} \mod n$,

where function L is defined as .

$$L(u) = \frac{u-1}{n}$$

Note that the notation $\frac{a}{b}$ does not denote the modular multiplication of a times the modular multiplicative inverse of b but rather the quotient of a divided by b

- The public (encryption) key is (n,g).

- The private (decryption) key is (λ, μ).

If using p,q of equivalent length, a simpler variant of the above key generation steps would be to set $g = n+1, \lambda = \varphi(n)$, and $\mu = \varphi(n)^{-1} \mod n$, where $\varphi(n) = (p-1)(q-1)$.[1]

Encryption
Let m be a message to be encrypted where $m \in \mathbb{Z}_n$
Select random r where $r \in \mathbb{Z}_n^*$
Compute ciphertext as: $c = g^m \cdot r^n \mod n^2$

Decryption
Ciphertext $c \in \mathbb{Z}_{n^2}^*$
Compute message: $m = L(c^\lambda \mod n^2) \cdot \mu \mod n$

As the original paper points out, decryption is "essentially one exponentiation modulo n^2."

- Creation of KDC
- KDC Authentication
- Trustee and User Accessibility
- Creation of Access Policy
- File Accessing
- File Restoration

III. 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 input as username and user id.

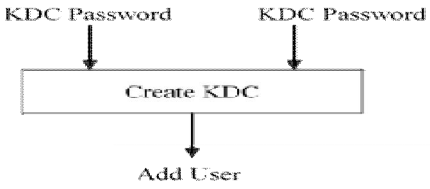


Fig Creation of KDC

IV. KDC AUTHENTICATION

Users have an initial level Registration Process at the web end. The users provide their own personal information for this process. The server in turn stores the information in its database.

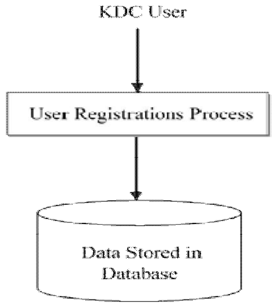


Fig User Validation

V. TRUSTEE AND KEY DISTRIBUTION CENTER

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 (like health/social insurance number), the trustee gives her a token. There are multiple KDCs (here 1), 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.

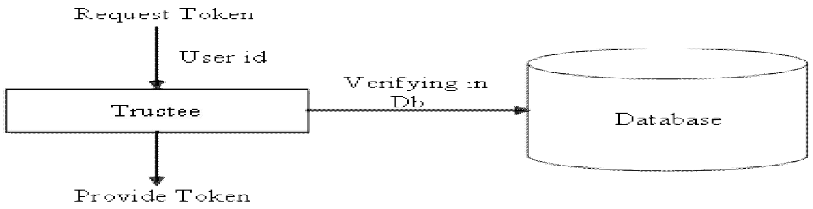


Fig: User Accessibility

VI. 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.

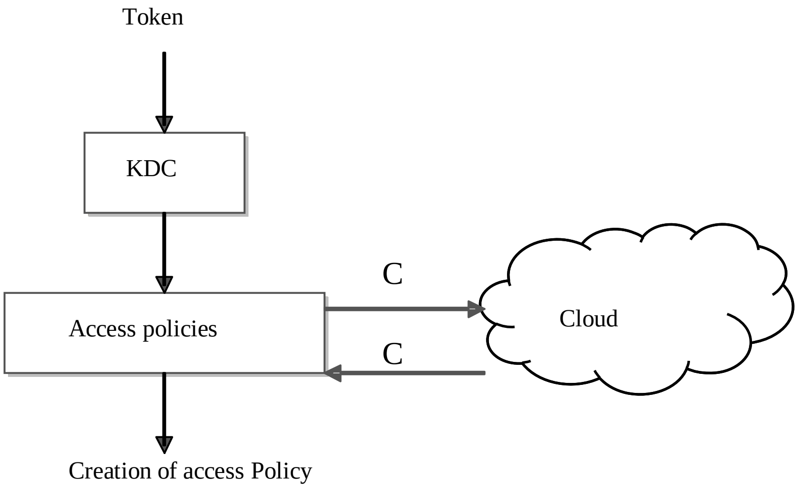


Fig: Creation of access policy

VII. 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).

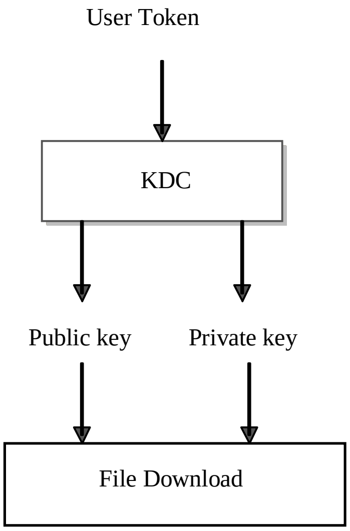


Fig: File Accessing

VIII. FILE RESTORATION

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

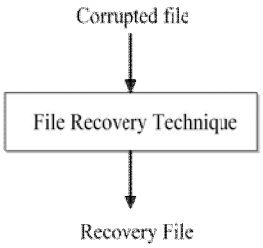


Fig: File Restoration

IX. CONCLUSION

I 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, I would like to hide the attributes and access policy of a user.

REFERENCES

[1] Ruj, S, Stojmenovic, M and Nayak, A 2013, 'Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds', IEEE transactions on parallel and distributed systems.

[2] Beimel, A 1996, 'Secure Schemes for Secret Sharing and Key Distribution', PhD Thesis. Technion, Haifa.

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

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

[5] Goyal, V, Pandey, O, Sahai, A and Waters, B 2006, 'Attribute-based encryption for fine-grained access control of encrypted data', in ACM Conference on Computer and Communications Security, pp. 89–98.

[6] Li, J, Wang, Q, Wang, C, Cao, N, Ren, K and Lou, W 2010, 'Fuzzy keyword search over encrypted data in cloud computing', in IEEE INFOCOM. , pp. 441–445.

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

[8] Lin, H, Cao, Z, Liang, X and Shao, J 2008, '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.

[9] Maji, H,K, Prabhakaran, M and Rosulek, M 2008, 'Attribute-based signatures: Achieving attribute-privacy and collusion-resistance', IACR Cryptology Print Archive.

[10] Ruj, S, Stojmenovic, M and Nayak, A 2012, 'Privacy Preserving Access Control with Authentication for Securing Data in Clouds', IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 556–563.

[11] Sahai, A and Waters, B 2005, 'Fuzzy identity-based encryption', in EUROCRYPT, ser. Lecture Notes in Computer Science, vol. 3494. Springer, pp. 457–473.

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