© 2015 IJSRSET | Volume 1 | Issue 2 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Cryptographic Cloud Storage with Anonymous Authentication

S.Meera, G. NasreenHameedaBanu, V. Vaishnavi, S. Geetha Rani
Dhanalakshmi College of Engineering, Chennai, Tamilnadu, India

ABSTRACT

We propose a new decentralized access control scheme for protected data storage in clouds that affirms anonymous authentication. In this paper, the cloud verifies the authenticity without knowing the user's identity before storing data. Our paper has the added feature of access control in which only validated users are able to decrypt the stored information. Here we also forbid replay attacks and supports creation, modification, and reading of information stored in the cloud. We also cover user revocation and here the authentication and access control scheme is decentralized and robust, unlike other access control schemes designed for clouds which are centralized.

Keywords: Attribute-based encryption, Secure cloud storage, Privacy Preservation, Anonymous Authentication, Decentralized Access Control, Paillier's Cryptosystem.

I. INTRODUCTION

In cloud computing, users will outsource their computation and storage to servers (also called clouds) using Internet. Much of the data stored in clouds is highly sensitive, for example, medical records, financial records and social networks. Security and privacy are very important issues in cloud computing. In one hand, the users should authenticate themselves before starting any transaction and on the other hand, it must be ensured that the cloud does not meddle 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 responsible for the services it provides. The validity of the user who stores the data is also verified.

To provide secure data storage, the data needs to be encrypted. However, the data is often modified and this impulsive 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. 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.

II. METHODS AND MATERIAL

A. Proposed System

In Existing work on access control in cloud are centralized in nature. Some scheme uses a symmetric key approach and does not support authentication. Earlier work provides privacy preserving authenticated access control in cloud. However, it takes 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 propose a new decentralized access control scheme for secure data storage in clouds that supports anonymous authentication. In the proposed scheme, the cloud verifies the authenticity of the user without knowing the user's identity before storing data. Our scheme also has the added feature of access control in which only valid users are able to decrypt the stored information. A third party (trusted legal authority named trustee) is present to provide privacy preservation and

data integrity. This proposed system is used to maintain sensitive information like medical and financial records. 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 the information/data.
- ✓ The identity of the user is shielded from the cloud during authentication.
- ✓ The architecture is decentralized that is there can be several KDCs for key management.
- ✓ Revoked users cannot access data after they have been revoked.
- ✓ The proposed scheme is resilient to replay attacks. A user whose attributes and key have been revoked cannot perform any operations.
- ✓ It supports multiple read and write on the data stored in the cloud
- ✓ . By using cryptographic storage, we process and verify the data and also generate token.

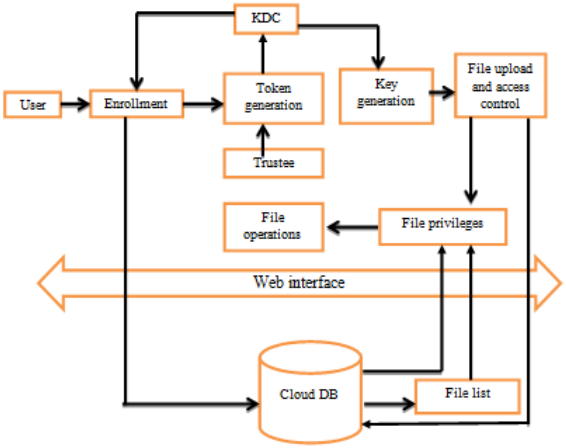

Figure 1: Architecture Diagram

B. Algorithms Used

i. Paillier's Cryptosystem:

a. Key generation

1. Choose two large prime numbers p and q randomly and independently of each other such that. $\gcd(pq,(p-1)(q-1))=1$ (eq 1.1) This property is assured if both primes are of equal length.[1]
2. Compute $n=pq$ and $\lambda=\operatorname{lcm}(p-1,q-1)$.
3. Select random integer g where $g\in\mathbb{Z}_{n^2}^*$
4. Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse:

$$\mu=(L(g^{\lambda}\bmod n^2))^{-1}\bmod n \text{ (Eq 1.2)}$$

Where function L is defined as $L(u)=\dfrac{u-1}{n}$

Note that the notation $\dfrac{a}{b}$ does not denote the modular multiplication of a times the modular multiplicative inverse of b but rather the quotient of a divided by b, i.e., the largest integer value $v\geq 0$ to satisfy the relation $a\geq vb$.

- The public (encryption) key is (n,g).
- The private (decryption) key is (λ,μ).

If using p, q of equivalent length, and a simpler variant of the above key generation steps would be to set

$$g=n+1,\lambda=\varphi(n), \text{ (Eq 1.3)}$$

And

$$\mu=\varphi(n)^{-1}\bmod n$$

(Eq 1.4)

Where, $\varphi(n)=(p-1)(q-1)$. [1]

b. Encryption

1. Let m be a message to be encrypted where $m\in\mathbb{Z}_n$
2. Select random r where $r\in\mathbb{Z}_n^*$
3. Compute ciphertext as:

$$c=g^m\cdot r^n\bmod n^2 \text{ (Eq 1.5)}$$

c. Decryption

1. Let c be the ciphertext to decrypt, where $c\in\mathbb{Z}_{n^2}^*$

Compute the plaintext message as:

$$m = L(c^{\lambda} \bmod n^2) \cdot \mu \bmod n \quad \text{(Eq 1.6)}$$

ii. SHA Algorithm:

Steps:

1. Append Padding Bits….

Message is "padded" with a 1 and as many 0's as necessary to bring the message length to 64 bits less than an even multiple of 512.

2. Append Length....

64 bits are appended to the end of the padded message. These bits hold the binary format of 64 bits indicating the length of the original message.

3. Prepare Processing Functions….
   a. SHA1 requires 80 processing functions defined as:
      i. f(t;B,C,D) = (B AND C) OR ((NOT B) AND D) (0 <= t <= 19)
      ii. f(t;B,C,D) = B XOR C XOR D
   b. (20 <= t <= 39)
      i. f(t;B,C,D) = (B AND C) OR (B AND D) OR (C AND D) (40 <= t <=59)

      ii. f(t;B,C,D) = B XOR C XOR D (60 <= t <= 79)
4. Prepare Processing Constants....
   a. SHA1 requires 80 processing constant words defined as:
      i. K(t) = 0x5A827999 (0 <= t <= 19)
      ii. K(t) = 0x6ED9EBA1 (20 <= t <= 39)
      iii. K(t) = 0x8F1BBCDC (40 <= t <= 59)
      iv. K(t) = 0xCA62C1D6 (60 <= t <= 79)
5. Initialize Buffers….
   a. SHA1 requires 160 bits or 5 buffers of words (32 bits):
   b. H0 = 0x67452301
      1. H1 = 0xEFCDAB89
      2. H2 = 0x98BADCFE
      3. H3 = 0x10325476
      4. H4 = 0xC3D2E1F0
6. Processing Message in 512-bit blocks (L blocks in total message)….

This is the main task of SHA1 algorithm which loops through the padded and appended message in 512-bit blocks. Input and predefined functions:

M[1, 2, ..., L]: Blocks of the padded and appended message f(0;B,C,D), f(1,B,C,D), ..., f(79,B,C,D): 80 Processing Functions K(0), K(1), ..., K(79): 80 Processing Constant Words H0, H1, H2, H3, H4, H5: 5 Word buffers with initial values

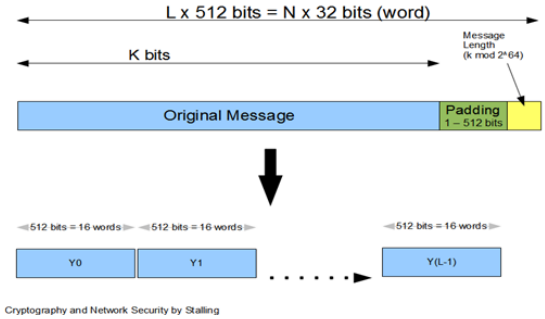

Figure 2: Algorithm

III. RESULTS AND DISCUSSION

We implement this project for the secure data exchange among the universities and in the implementation part both the KDCs(universities themselves) and the eligible users(professors, lecturers and students) will 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. If the user is going to upload a file, then he/she should get a token from trustee for key generation. 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. 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.

IV. CONCLUSION

We have portrayed a decentralized access control technique with anonymous authentication which provides user revocation and prevents replay attacks. Here we use cryptographic storage for the secure data storage in clouds. The cloud does not know the identity of the user who stores information, but only verifies the user's bona fides. 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.

V. REFERENCES

[1] 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.

[2] http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cs-01-en.pdf, 2013.

[3] http://securesoftwaredev.com/2012/08/20/xacml-in-the-cloud, 2013.

[4] 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.

[5] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward Secure and Dependable Storage Services in Cloud Computing,"

[6] IEEE Trans. Services Computing, vol. 5, no. 2, pp. 220-232, Apr.-June 2012.

[7] 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 Security Practice and Experience (ISPEC), pp. 83-97, 2011.

[8] 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 Security Practice and Experience (ISPEC), pp. 83-97, 2011.

[9] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed Access Control in Clouds," Proc. IEEE 10th Int'l Conf. Trust, Security and Privacy in Computing and Communications (TrustCom), 2011.

[10] S. Jahid, P. Mittal, and N. Borisov, "EASiER: Encryption-Based Access Control in Social Networks with Efficient Revocation,"

[11] Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS), 2011.

[12] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the Decryption of ABE Ciphertexts," Proc. USENIX Security Symp., 2011.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy Keyword Search Over Encrypted Data in Cloud Computing,"

[20] Proc. IEEE INFOCOM, pp. 441-445, 2010.

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