IJCSEC- International Journal of Computer Science and Engineering Communications.
Vol.3, Issue 2, 2015, Page.881-884, ISSN: 2347–8586

Secure Data Sharing With Data Integrity in Public Clouds Using Mediated Certificate-Less Encryption

P.Renukadevi[1] , A.Josephselvakumar[2]

[1]PG Student [2]Asst. Prof Dept. of IT,
[1&2]Dept. of Computer Science and Engineering
Idhaya Engineering College for Women
renusivam92@gmail.com

Abstract: The popularity and widespread use of Cloud have brought great convenience for data sharing and collection. Data sharing with a large number of participants must take into account several issues, including efficiency, data integrity and privacy of data owner. The shared data must be strongly secured from unauthorized accesses. The common approach to ensure confidentiality is to encrypt the data before uploading it to the cloud. Many encryption mechanisms support fine-grained encryption based access control. However, they face the key escrow problem and the revocation problem. The existing mediated Certificateless-Public Key Encryption scheme reduces the key management, but the scheme was found to be insecure against partial decryption attack. Although their scheme relies on pairing operations, that incurs considerably high computational costs. The proposed mediated certificateless Public Key Encryption (mCL-PKE) scheme provides its formal security without pairing operations. The mCL-PKE solves the key escrow problem and revocation problem. The extension of mCL-PKE scheme encrypts data efficiently for multiple users. The data integrity verifier verifies the encrypted data for efficient data sharing among users.

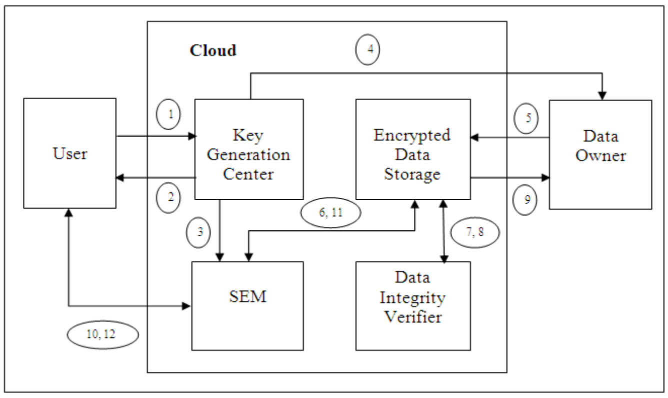
Index Terms: Cloud Computing, Certificateless cryptography, confidentiality, access control.

Introduction

Due to the benefits of public cloud storage, organizations have been adopting public cloud services such as Microsoft Skydrive and Dropbox to manage their data. However, for the widespread adoption of cloud storage services, the public cloud storage model should solve the critical issue of data confidentiality. That is, shared sensitive data must be strongly secured from unauthorized accesses. In order to assure confidentiality of sensitive data stored in public clouds, a commonly adopted approach is to encrypt the data before uploading it to the cloud. Since the cloud does not know the keys used to encrypt the data, the confidentiality of the data from the cloud is assured. However, as many organizations are required to enforce fine-grained access control to the data, the encryption mechanism should also be able to support fine-grained encryption based access control. The key management problem was solved by Identity-Based Public Key Cryptosystem (IB-PKC), but it suffers from the key escrow problem as the key generation server learns the private keys of all users. Recently, Attribute Based Encryption (ABE) has been proposed that allows one to encrypt each data item based on the access control policy applicable to the data. However, in addition to the key escrow problem, ABE has the revocation problem as the private keys given to existing users should be updated whenever a user is revoked. To solve these issues and for secure data sharing, Certificateless Proxy Re-Encryption (CL-PRE) is used. It relies on pairing operations. The recent advances in implementation techniques, the computational costs required for pairing are still considerably high compared to the costs of standard operations. The existing security model is often not sufficient to guarantee security in general protocol setting. Thus a secure mediated scheme without pairings is needed.

The existing mediated Certificateless-Public Key Encryption scheme reduces the key management. In this scheme, user's private key consists of a secret value chosen by the user and a partial private key generated by the Key Generation Center (KGC). If any user compromised the cloud, using its private key attempts to access the data in the cloud. The scheme was found to be insecure against partial decryption attack, since their security model did not consider the capabilities of the adversary in requesting partial decryptions.

Architecture diagram of proposed system



Proposed System

Techniques

User authentication and key generation

Input: New User

Output: Private and public Keys

New user obtains the token from Identity Provider by submitting their details. Whenever a new user accesses CSP (Cloud Service Provider), it asks for token to authenticate the user. Hence user should obtain the token from the Identity Provider before starting access of the data in CSP. Identity provider forwards the details of the user to the CSP. CSP stores the token and user details in database. Users provide the token obtained from Identity provider while login for the access in CSP. CSP validates the token using the details stored in database. If the token is valid, users are allowed for the access in cloud. Otherwise access is denied for the user. Each user first generates its own private and public key pair, called SK and PK, using the SetPrivateKey and SetPublicKey operations respectively.

SetUp:

KGC takes as input a security parameter k to generate two primes p and q such that $q \mid p - 1$.It then performs the following steps:

1) Pick a generator g of Z^*_p with order q.

2) Select $x \in Z^*_q$ uniformly at random and compute $y= g^x$.

3) Choose cryptographic hash functions H1

SetPrivateKey:

The entity A chooses $z_A \in Z^*_q$ uniformly at random as the private key of A.

SetPublicKey:

The entity A computes $U_A = g^{zA}$.

Encryption of data

Input: Private and public keys

Output: Encrypted data

The user sends its public keys and its identity (ID) to the KGC in the cloud. The KGC in turn generates two partial keys and a public key for the user. One partial key, referred to as SEM-key, is stored at the SEM in the cloud. The other partial key, referred to as U-key, is given to the user. The public key, referred to as KGC-key, consists of the user generated public key as well as the KGC generated public key. The KGC-key is used to encrypt data. The SEM-key, U-key, and SK are used together to decrypt encrypted data. The data owner obtains the KGC-keys of users from the KGC in the cloud. The data owner then symmetrically encrypts each data item for which the same access control policy applies using a random session key K and then the data owner encrypts K using the KGC-keys of users.

Data uploading

Input: Encrypted data

Output: Intermediate key with access control list

The encrypted data along with the access control list is uploaded to the cloud. Data owner generates intermediate key with hash value for each data. The encrypted content is stored in the storage service in the cloud and the access control list, signed by the data owner, is stored in the SEM in the cloud. Intermediate key is send to SEM and hash value is send to data integrity verifier.

Data integrity verification

Input: Encrypted data and its hash value

Output: Hash Value for received Encrypted data + Verification

Data integrity verifier generates the hash value for received encrypted data for security. It verifies the encrypted data hash value with data owner encrypted data hash value. Data integrity verifier stores the encrypted data in storage module, if hash value verification is true otherwise; it sends the data to data owner for re-encryption.

Decryption of SEM and user

Input: User request for retrieval

Output: Original data

A user wants to read some data; it sends a request to the SEM to obtain the partially decrypted data. The SEM first checks if the user is in the access control list and if the user's KGC-key encrypted content is available in the cloud storage. If the verification is successful, the SEM retrieves the encrypted content from the cloud and partially decrypts the content using the SEM-key for the user. The partial decryption at the SEM reduces the load on users. The user uses its SK and U-key to fully decrypt the data. In order to improve the efficiency of the system, once the initial partial decryption for each user is performed, the SEM stores back the partially decrypted data in the cloud storage. If a user is revoked, the data owner updates the access control list at the SEM so that future access requests by the user are denied. If a new user is added to the system, the data owner encrypts the data using the public key of the user and uploads the encrypted data along with the updated access control list to the cloud.

Related work

Al-Riyami [1] Certificateless public key cryptography (CL-PKC) schemes that do not require the use of certificates and yet do not have the built-in key escrow feature of ID-PKC. The solution for security is both of these properties; it is a model for the use of public key cryptography that is intermediate between traditional PKI and ID-PKC.Generalized Bilinear Diffie- Hellman Problem (GBDHP) is hard. **Dan Boneh [2]** It provides immediate revocation it is natural to first consider traditional revocation techniques. Many revocation methods have been proposed; they can be roughly classified into two prominent types: 1) explicit revocation structures such as Certificate Revocation Lists (CRLs) and variations on the theme, and 2) real time revocation checking such as the Online Certificate Status Protocol (OCSP) and its variants. Less security while user revocation. **Joonsang Baek [5]** In CLPKC, the user's public key is no longer an arbitrary string. Rather, it is similar to the public key used in the traditional PKC generated by the user.It does not solve revocation problem. **Xiaoxin Wu [7]** It is not sufficient to protect real-world applications and do not establish a strong security model. **Shucheng Yu [8]** It focuses on an important issue of attribute revocation which is cumbersome for CP-ABE schemes. In addition, proposed technique can also be applicable to the Key-Policy Attribute Based Encryption (KP-ABE) counterpart. It does not solve key escrow problem.

Conclusion

It proposed the first mCL-PKE scheme without pairing operations and provided its formal security. The mCL-PKE solves the key escrow problem and revocation problem. Using the mCL-PKE scheme as a key building block, it proposed an improved approach to securely share sensitive data in public clouds. The approach supports immediate revocation and assures the confidentiality of the data stored in an un-trusted public cloud while enforcing the access control policies of the data owner.

Future Work

It provides the data with high security using data integrity verifier that verifies the encrypted data using hash value generation.

REFERENCES

[1] Al-Riyami.S and Paterson.K, (2003) “Certificateless public cryptography”, Springer transaction on Advances in Cryptology - ASIACRYPT, Vol.2894, pp.452–473

[2] Boneh.D et al, (2004) “Fine-grained control of security capabilities”, ACM Transactions on Internet Technology, Vol.4, No.1, pp.60–82

[3] Chow.S.S.M et al, (2006) “Security mediated certificateless cryptography”, In Proceedings of the 9th international conference on Theory and Practice of Public-Key Cryptography, PKC’06, pp.508–524

[4] John Bethencourt et al, (2007) “Ciphertext-Policy Attribute-Based Encryption”, IEEE symposium and privacy, S&P’07, pp. 321-334

[5] Joonsang Baek et al, (2005) “Certificateless Public Key Encryption without Pairing”, Springer- Verlag Berlin Heidelberg, LNCS-3650, pp. 134-148

[6] Seung-Hyun Seo et al, (2013) “An Efficient Certificate-less Encryption for Secure Data Sharing in Public Clouds”, pp.1-14

[7] Wu.X et al, (2012) “POSTER: A certificateless proxy re-encryption scheme for secure data sharing with public cloud”, In ACM Symposium on Information, Computer and Communications Security, CCS’11, pp.17-21

[8] Yu.S et al, (2010) “Attribute based data sharing with attribute revocation”, In Proceedings of the 5thACM Symposium on Information, Computer and Communications Security, ASIACCS’10, pp.261–270