2581-4575

International Journal For Recent Developments in Science & Technology

Crossref

A Peer Reviewed Research Journal

PUBLIC KEY ENCRYPTION BASED DATA SHARING IN CLOUD STORAGE

DR.M.JAWAHAR[1], MD.AZHAR[2]

[1]Associate Professor, Department of Computer Science and Engineering, Narsimha Reddy Engineering College.

[2]Assistant Professor, Department of Computer Science and Engineering, Narsimha Reddy Engineering College.

mjawahar@gmail.com

ABSTRACT

In cloud, the data sharing is the most important functionality that has to be done securely and efficiently. The main problem which we are facing in cloud storage is how to share the data effectively. Although, there are many techniques that are developed to encrypt the data in the cloud for secure storage and sharing, we cannot able to share data flexibly with others. In this paper, a public-key cryptosystem that produces a constant-size cipher text is introduced. Here, the secret keys are accumulated together and make them as a single compact key. Other than the files that are encrypted and present outside the set remains trustworthy. These accumulated keys are sent secretly to other users or to be stored within limited secure storage. Thus, our KAC enables efficient data sharing in cloud storage with confidentiality and integrity.

Key words: Cloud storage, Key aggregate cryptosystem, RSA algorithm, Master key

1. INTRODUCTION

Recently, cloud computing plays a major role in computer technologies. Demands on data outsourcing to third party leads to the foundation of cloud computing. Cloud storage supports the confidentiality, integrity, and availability of data. Cloud storage not only meant for storing data but also shares those data across multiple users. Verification of data integrity and confidentiality is vital in cloud. The major problem in cloud storage is to share data meritoriously. Though, there are many cryptographic techniques by which users can encrypt the plain text into cipher text and again decrypt the cipher text into plain text then sharing those data to others it loses the value of cloud. So, user should provide the access rights to others for accessing data from the server openly. For example, consider that there are two users A and B accessing the cloud storage. When user A wants to share his data with user B then A simply puts his data into his drop box in an encrypted form. Now, user B can access the data of user A, but the problem is to get the decryption rights for those encrypted data. So, user A should send the secret keys for decryption to user B firmly. Encrypting the data with single encryption key and distinct encryption keys leads to various issues. Hence to obtain a solution, we are using a public key encryption technique called key-aggregate cryptosystem. Here the encryption

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

takes place with respect to both the public key and the identifier of the cipher text. The cipher texts are classified into different classes and those classes are given with an identifier. The secret keys for various classes can be extracted by using the master key, which is to be hold with the key holder. The extracted key is compact as the single key but has the control of many keys.

2. CONSTRUCTION OF RSA

Basic construction:

Our RSA consists of four phases such as setup phase, encrypt phase, key gen phase, decrypt phase.

Setup phase:

The setup algorithm takes no input other than the implicit security parameter. It produces the public parameters PK and a master key MK as output.

Encrypt phase:

Encrypt(PK, M, A). The encryption algorithm consumes the public parameters PK, a message M, and an access structure A as input. The algorithm will encrypt M and produce a cipher text CT such that only a user that possesses a set of attributes that satisfies the access structure will be able to decrypt the message.

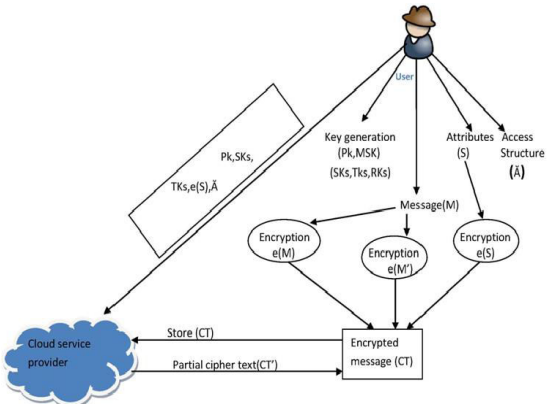

Fig.1. RSA construction

Key gen phase:

Key generation(MK, S). The key generation algorithm takes the master key MK and a set of attributes S that describe the key as input. It delivers a private key SK as the output.

Decrypt phase:

Decrypt (PK, CT, SK). The decryption algorithm takes the public parameters PK, a cipher text CT, and a private key SK, as input. Then, the algorithm will decrypt the cipher text and return a message M as output.

3.PROPOSED SCHEME

To enhance the integrity validation of an untrusted outsourced storage, many techniques were suggested. But, there were no effective results found. Hence, we proposed a RSA based dynamic inspection service. Our system is based on a new audit system architecture that supports run time data operations and appropriate unusual detection by using numerous effective methods which includes index-hash table,

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

random sampling, and fragment structure. In addition, an effective approach for periodic verification in order to improve the performance of inspection services is offered. Thus, to estimate the sustainability and feasibility of our recommended approach a proof-of-concept model is employed. Here, our attention is only on the RSA based Encryption with provable decryption. Our tentative outcomes not only certify the effectiveness of our method, but it also illustrates that our scheme consumes a lesser computational cost, along with smaller additional storage for integrity verification. This method affords accessible, location free platform for handling client's data.

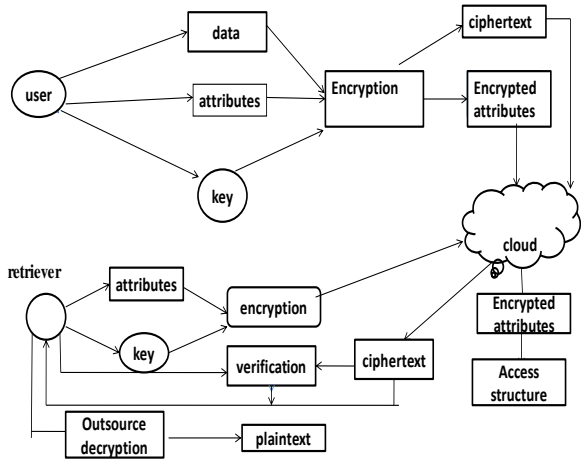

Fig.2 Architecture

4 RELATED WORKS

4.1 ABE

From the related study, we obtain that the ABE method delivers an efficient encryption and decryption processes. The decryption algorithm involves only a persistent number of pairing computations. Lately, Green *et al.* introduced the idea of ABE with outsourced decryption, which mostly excludes the decryption overhead that offers a remedy to this problem. In this method, a user affords an untrusted server, that is functioned by a cloud service provider, containing a transformation key TK in order to convert ABE cipher text CT into a simple cipher text CT' and it earns only a minor overhead for the user to retrieve the plaintext from the cipher text CT'. The main advantage of this scheme is that it guarantees the security that an opponent cannot be able to acquire the content of the encrypted message. Though, this method delivers no assurance on the correctness of the transformation done by the cloud server. It also increases the cost of storing and transmitting the cipher texts. It needs expensive tamper-proof memory for storing the secret keys. With increase in the number of decryption keys, the costs and the complexities involved gets increased.

4.2 Cryptographic keys

Cryptographic keys targets to minimize the cost of storing and maintaining the secret keys. It uses a tree structure where the descendant node keys can be derived from the key of the given branch. R. S. Sandhu. [11] organized a scheme to produce a tree of symmetric keys with the use of repeated evaluation of block cipher. This method is cannot be applied if there exist more complex classifications. Also, this hierarchical technique solves the problem partially if all the files are shared under a common branch of a tree.

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

4.3 Symmetric key Encryption

Thus, in order the overcome the drawback of flexible decryption, Benaloh *et al.* [10] proposed an encryption method for succinctly transmitting large number of keys. The derivation of the key can be done by choosing a master key at random from Z_N. Each class contains an individual prime. Here, the encryption takes place by getting the appropriate secret keys. This technique cannot be applied for most of the applications. As, this method derives a secret value rather than a secret keys, it is uncertain to imply it for the public key encryption.

4.4 Identity based Encryption

The public key encryption technique in which, the public key can be set as an identity of a user. Private key generator in IBE is a trusted party that contains the master secret key and distributes a secret key to every user based on their identity. Here the encryption takes place with respect to the public parameter and the user identity. Guo et al. [12], [10] have tried to construct IBE with key aggregation where, [12] has random oracles and [10] does not have it. The key aggregation takes the cost of O(n) sizes. This increases the expense of storing and distributing cipher texts.

4.5 Proxy re-encyption

This scheme allows the user to access the server that converts the cipher texts into a message. Here, the owner of the data has to fully trust on the server for converting the cipher texts. If the server colludes with other user, then the other data can be decrypted without the permission of the data owner. Hence it is undesirable to let the server exist in the storage server.

5. RESULTS AND DISCUSSION

Features	RSA	Hierarchy	Symmetric	Identity
Cost of storing	Low	Low	Low	High
Key size	Constant	Increases	Constant	Constant
Encryption technique	Asymmetric	Symmetric	Symmetric	Public
Flexibility	Flexible	Not flexible	Flexible	Not flexible

6. CONCLUSION

Protecting and sharing data securely with confidentiality and integrity is the major issue in cloud storage. Most of the techniques uses multiple keys for encrypting and decrypting data. This takes more memory to store those keys secretly. But, our approach is more flexible and consumes only little storage space for holding the secret keys.

REFERENCES

1. Cheng-Kang Chu, Sherman S. M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng,"Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage", Volume: 25, Issue: 2. Year :2014.
2. S. S. M. Chow, Y. J. He, L. C. K. Hui, and S.-M. Yiu, "SPICE -Simple Privacy-Preserving Identity-Management for Cloud Environment", vol. 7341. Springer, 2012, pp. 526–543
3. B. Wang, S. S. M. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud

2581-4575

International Journal For Recent Developments in Science & Technology

Crossref

A Peer Reviewed Research Journal

via Security-Mediator", ICDCS 2013. IEEE, 2013.

4. D. Boneh and M. K. Franklin, "Identity-Based Encryption from the Weil Pairing", CRYPTO'01, vol. 2139. Springer, 2001, pp. 213–229.
5. Q. Zhang and Y. Wang, "A Centralized Key Management Scheme for Hierarchical Access Control", GLOBECOM '04. IEEE, 2004, pp. 2067–2071.
6. C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy- Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, vol. 62, no. 2, pp. 362–375, 2013.
7. M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, "Dynamic and Efficient Key Management for Access Hierarchies," TISSEC, vol. 12,no. 3, 2009.
8. C.K. Chu and W.-G. Tzeng, "Identity-Based Proxy Re-encryption without Random Oracles," ISC'07, ser. LNCS, vol. 4779. Springer, 2007, pp. 189–202.
9. S. G. Akl and P. D. Taylor, "Cryptographic Solution to a Problem of Access Control in a Hierarchy," ACM (TOCS), vol. 1, no. 3, pp. 239–248, 1983.
10. J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," CCSW '09. ACM, 2009, pp. 103–114.
11. R. S. Sandhu, "Cryptographic Implementation of a Tree Hierarchy for Access Control," Information Processing Letters, vol. 27, no. 2, pp. 95–98, 1988.
12. F. Guo, Y. Mu, and Z. Chen, "Identity-Based Encryption: How to Decrypt Multiple Ciphertexts Using a Single Decryption Key," (Pairing '07), ser. LNCS, vol. 4575. Springer, 2007, pp. 392–406.