Review on Scalable and Secure Data Sharing Technique in Cloud Storage using Key Aggregate Cryptosystem

Ashwini T. Tande[1], Namdev M. Sawant[2], Abhijit A. Rajguru[3]
[1,2,3] *Department of Computer Science & Engineering, Solapur University, Solapur*
SKN Sinhgad College of Engineering, Korti, Solapur, MS, India
[1]ashwinitande44@gmail.com, [2]namdevsawant@gmail.com, [3]abhijitcse08@gmail.com

Abstract - ***Data sharing is an important functionality in cloud storage. In this article we describe new public-key cryptosystems which produce constant-size cipher texts such that efficient delegation of decryption rights for any set of cipher texts are possible. The novelty is that one can aggregate any set of secret keys and make them as compact as a single key, but encompassing the power of all the keys being aggregated. In other words, the secret key holder can release a constant-size aggregate key for flexible choices of cipher text set in cloud storage, but the other encrypted files outside the set remain confidential.***

Keywords - Data sharing, cloud storage, master secret key, aggregate key

I. INTRODUCTION

Cloud storage is nowadays very popular storage system. Cloud storage is storing of data off-site to the physical storage which is maintained by third party. Cloud storage is saving of digital data in logical pool and physical storage spans multiple servers which are manage by third party. Third party is responsible for keeping data available and accessible and physical environment should be protected and running at all time. Instead of storing data to the hard drive or any other local storage, we save data to remote storage which is accessible from anywhere and anytime. It reduces efforts of carrying physical storage to everywhere. By using cloud storage we can access information from any computer through internet which omitted limitation of accessing information from same computer where it is stored. While considering data privacy, we cannot rely on traditional technique of authentication, because unexpected privilege escalation will expose all data. Solution is to encrypt data before uploading to the server with user's own key.

Data sharing is again important functionality of cloud storage, because user can share data from anywhere and anytime to anyone. For example, organization may grant permission to access part of sensitive data to their employees. But challenging task is that how to share encrypted data. Traditional way is user can download the encrypted data from storage, decrypt that data and send it to share with others, but it loses the importance of cloud storage. Cryptography technique can be applied in a two major ways- one is symmetric key encryption and other is asymmetric key encryption. In symmetric key Encryption, same keys are used for encryption and decryption. This scheme is so powerful since it uses aggregate encryption algorithms which are very simple so that large number of data can be stored in cloud without any problem in a single scheme. The scheme detects any change made to the original file and if found clear the errors.

Now assume the scenario that Alice puts all her private photos on Dropbox, and she does not want to expose her photos to everyone. Due to various data leakage possibility Alice can not feel relieved by just relying on the privacy protection mechanisms provided by Dropbox, so she encrypts all the photos using her own keys before uploading. One day, Alice's friend, Bob, asks her to share the photos taken over all these years which Bob appeared in. Alice can then use the share function of Dropbox, but the problem now is how to delegate the decryption rights for these photos to Bob. A possible option Alice can choose is to securely send Bob the secret keys involved.

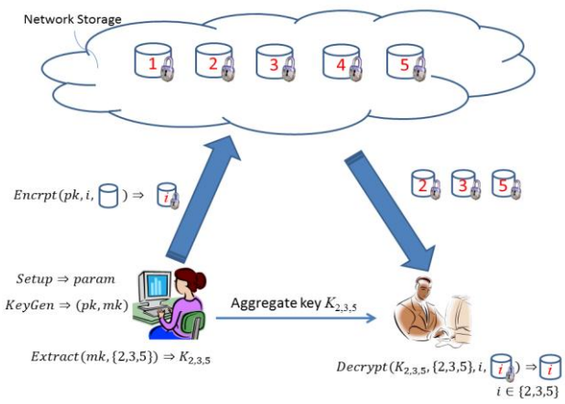

Fig.1. Alice shares files with identifiers 2, 3, 6 and 8 with Bob by sending him a single aggregate key.

Naturally, there are two extreme ways for her under the traditional encryption paradigm:

1. Alice encrypts all files with a single encryption key and gives Bob the corresponding secret key directly.
2. Alice encrypts files with distinct keys and sends Bob the corresponding secret keys.

Obviously, the first method is inadequate since all unchosen data may be also leaked to Bob. For the second method, there are practical concerns on efficiency. The number of such keys is as many as the number of the

shared photos, say, a thousand. Transferring these secret keys inherently requires a secure channel, and storing these keys requires rather expensive secure storage. The costs and complexities involved generally increase with the number of the decryption keys to be shared. In short, it is very heavy and costly to do that. The key idea with first paradigm is shown in fig.1. is public-key cryptosystems which produce a set of constant-size cipher texts such that efficient delegation of decryption rights for any set of cipher texts is possible.

The data owner has the aggregate decryption key which is extracted from different cipher text classes, and by concluding the power of all the keys aggregated but the other encrypted files outside the set remain confidential and very much authenticated. The advantage of this scheme is it provides secure data storage and retrieval and also the schemes detects any changes made to the original file stored in cloud and clear the errors if any changes found. The disadvantage of this scheme is it consumes more time for checking and recovery of every file.

II. KEY AGGREGATE CRYPTOSYSTEM

The data owner establishes the public system parameter through Setup and generates a public/master-secret key pair through KeyGen. Data can be encrypted via Encrypt by anyone who also decides what cipher text class is associated with the plaintext message to be encrypted. The idea of cipher text class is shown in fig.2. Here data owner can encrypt and share multiple files using same constant cipher text class index towards to reduce the no of Class index files for individual files thus it improve the performance and storage space.

The data owner can use the master-secret key pair to generate an aggregate decryption key for a set of cipher text classes through Extract. Finally, any user with an aggregate key can decrypt any cipher text provided that the cipher text's class is contained in the aggregate key via Decrypt.Key aggregate encryption schemes consist of five polynomial time algorithms as follows:

Setup: The data owner executes the setup phase for an account on server which is not trusted. The setup algorithm only takes implicit security parameter.

KeyGen: This phase is executed by data owner to generate the public or the master key pair (pk, msk).

Encrypt: This phase is executed by anyone who wants to send the encrypted data. Encrypt (pk, m, i), the encryption algorithm AES takes input as public parameters pk, a message m, and i denoting cipher text class. The algorithm encrypts message m and produces a cipher text C.

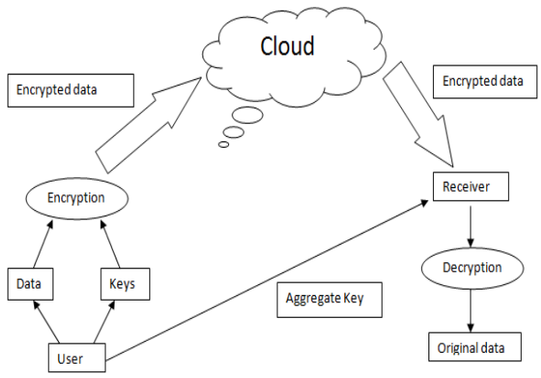

Fig.2. Data sharing architecture

Extract: This is executed by the data owner for delegating the decryption power to the users by providing his Aggregate Decryption key.

Decrypt this is executed by the receiver who has the decryption authorities. Decrypt (Ks, S, i, C), the decryption algorithm takes input as public parameters pk, a cipher text C, i denoting cipher text classes for a set S of attributes. For example Alice wants to upload her data on the server. First she need to Setup an account on the server with security level parameter(1) and cipher text classes(n) and then the public(pk) and master-secret key(msk) is generated by KeyGen algorithm. The data and index are encrypted by Alice as Encrypt (pk, i, m). Alice's master-secret key is used to compute the aggregate key by performing Extract (msk, S). Then Bob can be able to download the data from the server by Decrypt (Ks, S, i, Ci).

III.LITERATURE REVIEW

A. Attribute-Based Encryption:

In this scheme a new cryptosystem for fine-grained sharing of encrypted data called Key-Policy Attribute-Based Encryption (KP-ABE) has been predicted. In this cryptosystem, cipher texts are labeled with sets of attributes and private keys are associated with access structures that control which cipher texts a user is able to decrypt. Further demonstration is the applicability of construction to sharing of audit-log information and broadcast encryption. This construction supports delegation of private keys which assumes Hierarchical Identity-Based Encryption (HIBE) [1].

B. Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records:

In PCE (Patient controlled encryption), the health record is decomposed into a hierarchical representation based on the use of different ontologies, and patients are the parties who generate and store secret keys. When there is a need for healthcare personnel to access part of the record, a patient will release the secret key for the concerned part of the record. For this three solutions have been provided,

which are symmetric-key PCE for fixed hierarchy, public-key PCE for fixed hierarchy and RSA-based symmetric-key PCE for flexible hierarchy [2].

C. A Time-Bound Cryptographic Key Assignment Scheme for Access Control:

This is time bound cryptographic key assignment scheme in which the cryptographic keys of a class are different for each time period, that is the cryptographic key of class C_i at time t is $K_{i,t}$. Key derivation is constrained by not only the class relation but also the time period. Each user holds some secret parameters whose number is independent of the number of the classes in the hierarchy and the total time periods. There are two applications, one is to broadcast data to authorized users in a multilevel security way and the other is to construct a flexible cryptographic key backup system [3].

D. Fuzzy Identity-Based Encryption:

A Fuzzy IBE scheme allows for a private key for an identity, ω, to decrypt a cipher text encrypted with an identity, ω', if and only if the identities ω and ω' are close to each other as measured by the "set overlap" distance metric. A Fuzzy IBE scheme can be applied to enable encryption using biometric inputs as identities; the error-tolerance property of a Fuzzy IBE scheme is precisely what allows for the use of biometric identities, which inherently will have some noise each time they are sampled. Additionally, they show that Fuzzy-IBE can be used for a type of application that has termed as "attribute-based encryption" [4].

E. Chosen-cipher text secure proxy re-encryption:

In a proxy re-encryption (PRE) scheme, a proxy is given special information that allows it to translate a cipher text under one key into a cipher text of the same message under a different key. The proxy cannot, learn anything about the messages encrypted under either key. Author proposed a definition of security against chosen cipher text attacks for PRE schemes, and present a scheme that satisfies the definition. They also formally capture CCA security for PRE schemes via both a game-based definition and simulation-based definitions that guarantee universally compassable security [5].

IV. CONCLUSION

How to protect user's data privacy is a central question of cloud storage. Here we studied different techniques to share data securely and efficiently. The most efficient technique is key aggregate cryptosystem. Using this scheme data owner stores encrypted data on cloud. Then he/she will share data with receiver and will give the decryption rights to receiver using aggregate key. With more mathematical tools, cryptographic schemes are getting more versatile.

REFERENCES

[1] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine- Grained Access Control of Encrypted data," in Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06). ACM, 2006, pp. 89–98.

[2] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," in Proceedings of ACM Workshop on Cloud Computing Security (CCSW '09). ACM, 2009, pp. 103–114.

[3] W.-G. Tzeng, "A Time-Bound Cryptographic Key Assignment Scheme for Access Control in a Hierarchy," IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 14, no. 1, pp. 182–188, 2002.

[4] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," in Proceedings of Advances in Cryptology - EUROCRYPT '05, ser. LNCS, vol. 3494. Springer, 2005, pp. 457–473.

[5] R. Canetti and S. Hohenberger, "Chosen-Ciphertext Secure Proxy Re- Encryption," in Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07). ACM, 2007, pp. 185–194.

[6] C-K. Chu, S. S. M. Chow, W-G. T., J. Zhou, and R. H. Deng, "Key Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage," IEEE Transactions on Parallel and Distributed Systems. Volume: 25, Issue: 2. Year :2014.

[7] F. Guo, Y. Mu, Z. Chen, and L. Xu, "Multi-Identity Single-Key Decryption without Random Oracles," in Proceedings of Information Security and Cryptology (Inscrypt '07), ser. LNCS, vol. 4990. pringer, 2007, pp. 384–398.

[8] M. Vilasini, Dr. K. S. Babu "Secure Compressed Key Sharing for Multiple Cipher Text in Cloud Storage," International Journal of Advanced Research in Computer Science and Software Engineering.

[9] H.Fareesa Firdose , R.Deepthi Crestose Rebekah, "A Key Aggregate Construction with Adaptable Offering of Information in Cloud " International journal of computer engineering in research TRENDS VOLUME 2, ISSUE 5, MAY 2015, PP 355-358 , ISSN (Online): 2349-7084.www.ijcert.org

[10] D. Boneh and M. K. Franklin, "Identity-Based Encryption from the Weil Pairing," in Proceedings of Advances in Cryptology – CRYPTO '01, ser. LNCS, vol. 2139. Springer, 2001,pp. 213–229.