ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

Secure Data Sharing With Key Aggregate Cryptosystem

Pallavi Mathur[1], Nachiket Mahamuni[2], Nikhil Gondane[3], Kaushal Singh[4]

Computer Department, DYPIET, Pune University (UoP), Pune, India [1,2,3,4]

Abstract: Cloud storage is a storage of information on-line in cloud that is accessible from multiple and connected resources. Cloud storage is the storage that offers smart accessibility and dependability, sturdy protection, disaster recovery, and lowest price. Cloud storage definitely has important practicality i.e. securely, sharing information with others, with great efficiency. New public–key cryptography is introduced that is termed as Key aggregate cryptosystem (KAC). Key-aggregate cryptosystem yield constant size ciphertexts so that trustworthy relinquishment of decryption rights for a prepared bunch of ciphertexts is feasible. Any set of secret keys may be mass composed and form a single key that encompasses power of all the keys being mass composed. This combination key may be sent to the others over a secured channel, and remaining encrypted files are untouched and remain confidential. The system has great potential to leverage secured use of cloud system.

Keywords: Cloud storage, Data sharing, Ciphertext, Key-aggregate, Encryption.

I. INTRODUCTION

Cloud storage is today very fashionable storage system. Cloud storage is storing of knowledge off-site to the physical storage that is maintained by third party. Cloud storage is saving of digital information in logical pool and physical storage carrying loads on multiple servers that are manage by third party. Third party is liable for keeping information on the market and accessible and physical atmosphere ought to be protected and running in the slightest degree time. Rather than storing information to the disk drive or the other native storage, we tend to save information to remote storage which is accessible from anyplace and anytime. It reduces efforts of carrying physical storage to everyplace. By exploitation cloud storage we can access data from any pc through web that omitted limitation of accessing data from same pc where it's kept. While considering information (content) privacy and protection, resolution is to encipher information before uploading to the server with user’s own key. Information sharing is once more necessary functionality of cloud storage, as a result of user will share information from anyplace and anytime to anyone. For instance, organization might grant permission to access a part of sensitive information to their staff. However difficult task is to share encrypted information securely. Traditional manner is user will transfer the encrypted information from storage, decipher that information and send it to share with others; however it loses the importance of cloud storage.

Cryptography technique is applied in a very 2 major ways-one is even key cryptography and other one is uneven key encryption. In even key cryptography, same keys are used for cryptography and coding. Against this, in uneven key encryption totally different keys are used, public key for cryptography and personal key for coding. Exploitation uneven key cryptography is additionally flexible for our approach. This could be illustrated by following example. Suppose Alice place all information on demo.com and she or he doesn't need to show her information to everybody. Thanks to information outpouring prospects she does not trust on privacy mechanism provided by demo.com, thus she encipher all information before uploading to the server. If Bob raise her to share some information then Alice use share operates of demo.com. However drawback now's that the way to share encrypted information.

There are 2 ways:
1. Alice enciphers information with single secret key and shares that secret key directly with the Bob.
2. Alice will encipher information with distinct keys and send Bob corresponding keys to Bob via secure channel. In 1st approach, unwanted information conjointly get expose to the Bob, which is insufficient. In second approach, no. of keys is as several as no. of shared files, which can be hundred or thousand in addition as transferring these keys need secure channel and space for storing which may be dearly-won. Therefore best resolution to on top of drawback is Alice encrypts information with distinct public keys, however send single coding key of constant size to Bob. Since the coding key ought to be sent via secure channel and unbroken secret tiny size is usually desirable. To design associate degree economical public-key cryptography theme that supports versatile delegation within the sense that any set of the ciphertexts (produced by the cryptography scheme) is decryptable by a constant-size coding key (generated by the owner of the master-secret key).

II. RELATED WORK

The data owner establishes general public system parameter by victimization Setup and generates a public/master-secret key combine by victimization Key Gen. Messages are often encrypted victimization cipher by anyone UN agency additionally decides what ciphertext category is related to the simple text message to be encrypted. The info owner will use the master-secret

DOI 10.17148/IJARCCE.2015.4961

to generate associate combination decoding key for a group of ciphertext categories by Extract. The generated keys are often passed to Receivers firmly via secure e-mails. Finally, associate user with a combination key will rewrite any ciphertext given that the ciphertext's category is contained within the combination key via rewrite.

A key-aggregate cryptography system essentially includes5 recursive steps as follows-

- **Setup** (1λ , n): information owner executes Setup to produce associate account on associate untrusted server. With input as security level parameter 1λ and therefore the range of ciphertext categories n, it outputs the general public system parameter param.
- **KeyGen**: information owner executes KeyGen to indiscriminately generate a public/master-secret key combine (pk, msk)
- **Encrypt** (pk, i, m): Anyone will execute this step UN agency needs to cipher information with input a public-key pk, associate index i denoting the ciphertext category, and a message m, that outputs a ciphertext C.
- **Extract** (msk, S): dead by the info owner to relinquishing the cipher power for a precise set of encrypted text categories on a Receiver. A we give input master-secret key msk and a group S of indices related to totally different categories, it outputs the combination key for set S denoted by Sunflower State.
- **Decrypt** (Ks, S, i, C): dead by a Receiver UN agency received associate combination key Extract generated by Sunflower State. On input Sunflower State, the set S, associate index i denoting the ciphertext category the encrypted text C belongs to, and C, it outputs the decipher result m if i ϵ S

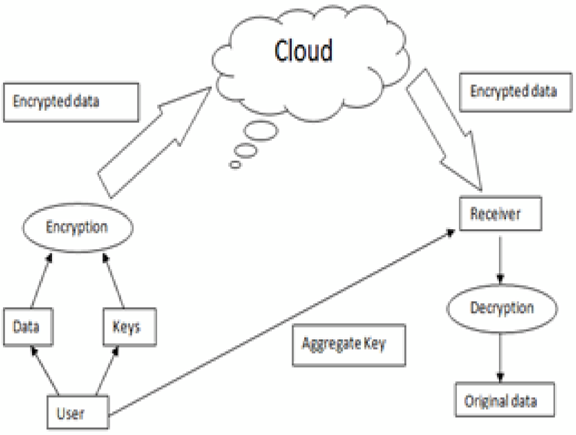

Fig1 Data flow between user and receiver

In this section basic KAC theme is compared with alternative possible solutions on sharing in secure cloud storage.

a) Cryptographic key scheme for hierarchy

Is a assignment schemes works on the premise of minimizing/reducing the expense in storing and managing secret keys for general cryptanalytic use by employing a tree structure (e.g., [6],[7],[8]). By victimization class-conscious tree structure, a key for a given sub key don't need to derive the keys of its descendant nodes. This can solve the matter part if one intends to share all files underneath a precise branch within the hierarchy that instead means the amount of keys will increase with the amount of branches. So it is difficult to form a hierarchy which will save the amount of total keys to be granted for all people at the same time.

b) Compact Key in Symmetric-Key coding

This methodology is employed to come up with a secret rather than a pair of public/ secret keys [2]. It's designed for the symmetric-key setting during which the encryptor gets the corresponding secret keys to cipher knowledge. Thus it is unclear the way to apply this concept for public key coding scheme.

c) Compact Key in Identity-Based coding (IBE)

In this coding, there's a trustworthy party referred to as personal key generator in IBE that holds a master-secret key and gives a secret key to every user with regard to the user identity (e.g., [11],[12],[13]). [2], [4] made an attempt to build 'IBE' using 'key-aggregation'. The encryptor will take the general public parameter and a user identity to cipher a message. The receiver will decrypt this ciphertext by his secret key. Some tried to build IBE with key aggregation. However their key-aggregation comes at the expense of O(n) sizes for each ciphertext and the public parameter, wherever n is that the variety of secret keys. This greatly will increase the prices to store and deploy the ciphertext.

d)Attribute-based coding (ABE)

This theme maintains every ciphertext to be associated with associate degree attribute, and also the master-secret key holder will get a secret key for a role of those attributes in order that a encrypted text may be decrypted by this key. However the scale of the key typically increases with the amount of attributes it includes, ciphertext-size isn't constant.

III.PROPOSED SYSTEM

Following fig. 2 gives idea about how data can be shared using proposed system architecture.

A canonical application of KAC is knowledge sharing. The key aggregation property is particularly helpful after we expect delegation to be economical and versatile. The KAC schemes alter a content supplier to share her knowledge in a very confidential and selective means, with a set and tiny ciphertext enlargement, by distributing to every approved user one and tiny mixture key. Data sharing in cloud storage victimisation KAC, illustrated in Figure one. Suppose Alice desires to share her knowledge m1, m2... mn on the server. She initial performs Setup (1λ, n) to induce param and execute KeyGen to induce the public/master-secret key combine (pk, msk). The system parameter param and public-key pk are often created public and master secure-secret key msk ought to be unbroken secret by Alice. Anyone will then write every mi by Ci = write (pk, i, mi). The encrypted knowledge square measure uploaded to the server.

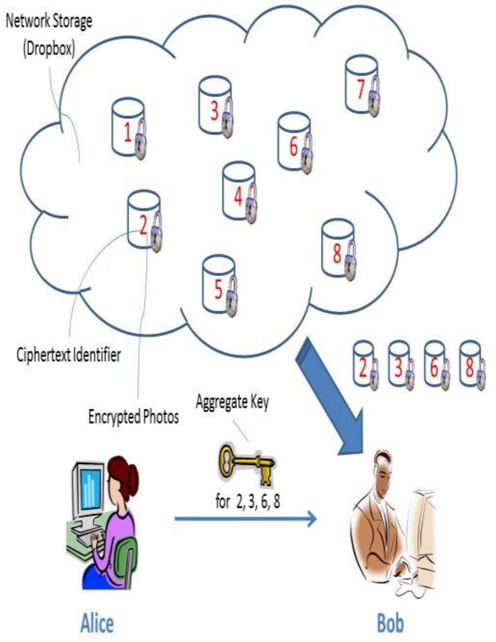

Fig2 System architecture [1].

With param and pk, folks that get together with Alice will update Alice's knowledge on the server. Once Alice is willing to share a collection S of her knowledge with an addict Bob, she will be able to calculate the aggregate key Kansas for Bob by playing Extract (msk, S). Since Kansas is simply a relentless size key, it's straightforward to be sent to Bob through a secure e-mail. once getting the mixture key, Bob will transfer the info he's approved to access. That is, for every i ϵ S, Bob downloads Ci from the server. With the mixture key Kansas, Bob will decode every Ci by decode (KS, S, i, Ci) for every i ϵ S.

IV. CONCLUSION

Users' knowledge (content/data) privacy could be a central question of cloud storage. In this, we emphasize Compression of the secret keys in public-key cryptosystems that support various cipher text categories in cloud storage. It could be any one of the ability set of categories, no issues; the delegate will forever get combination key of constant size. In cloud storage, the quantity of cipher texts sometimes grows speedily with none restrictions. Therefore we've to order enough cipher text categories for the longer term extension. Otherwise, we should expand the public-key.

V. ACKNOWLEDGMENT

The authors herein thank the publishers, researchers for making their resources available, as well as professors for their advice and proper guidance. We also thank the college/institute authority for providing us the desired infrastructure and their honest support.

REFERENCES

[1] Cheng-Kang Chu, Sherman S. M, "Key Aggregate Cryptosystem for Scalable Data Sharing in cloud storage", IEEE Transactions on Parallel and Distributed Systems, vol. 25, issue2, 2014.

[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] 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 .

[4] 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. Springer, 2007, pp. 384–398.

[5] M. Chase and S.S.M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," Proceeding ACM Conference on Computer and Communication Security, pp. 121-130. 2009.

[6] S. G. Akl and P. D. Taylor, "Cryptographic Solution to a Problemof Access Control in a Hierarchy," ACM Transactions on Computer Systems (TOCS), vol. 1, no. 3, pp. 239–248, 1983.

[7] G. C. Chick and S. E. Tavares, "Flexible Access Control with Master Keys," in Proceedings of Advances in Cryptology – CRYPTO '89, ser. LNCS, vol. 435. Springer, 1989, pp. 316–322.

[8] 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.

[9] R. S. Sandhu, "Cryptographic Implementation of a Tree Hierarchy for Access Control," Information Processing Letters, vol. 27, no. 2, pp. 95– 98, 1988.

[10] Y. Sun and K. J. R. Liu, "Scalable Hierarchical Access Control in Secure Group Communications," in Proceedings of the 23th IEEE International Conference on Computer Communications (INFOCOM '04). IEEE, 2004.

[11] 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.

[12] 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.

[13] S. S. M. Chow, Y. Dodis, Y. Rouselakis, and B. Waters, "Practical Leakage-Resilient Identity-Based Encryption from Simple Assumptions," in ACM Conference on Computer and Communications Security, 2010, pp. 152–161

DOI 10.17148/IJARCCE.2015.4961