Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage

[1]Vijay Joseph A, [2]Marrynal S Eastaff

[1]*PG Scholar, PG Department of IT, Hindusthan College of Arts and Science*
[2]*Asst. Professor, PG Department of IT, Hindusthan College of Arts and Science*

---***---

Abstract - *Cloud Storage plays a vital role in data sharing. A secure, efficient and flexible way to share data is discussed in this paper. A new public-key cryptosystems which make constant-size cipher texts such that efficient delegation of decryption rights for any set of cipher text are hopeful. The novelty is that one can collective any set of top covert keys and build them as compact as a single key, but encompassing the run of all the keys person aggregated. In extra conditions the top secret key skill can let go a constant-size collective key for flexible option of cipher book set in cloud storage, but the other encrypted files external the set remain confidential. This compact total key can be conveniently sent to others or be stored in a smart card with very partial locked storage. We give formal security analysis of our scheme in the average model. We also describe other application of our schemes. In exacting, our scheme gives the first public-key patient-controlled encryption for stretchy pecking order, which was yet to be known.*

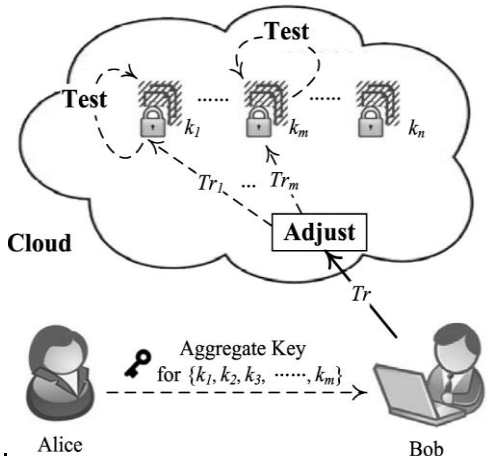
Index Terms—Cloud storage, data sharing, key aggregate encryption, patient-controlled encryption.

1. INTRODUCTION

Cloud storeroom has emerged as a promise solution for providing everywhere, convenient, and on-demand accesses to large amounts of data shared over the Internet. Today, millions of users are sharing special data, such as photos and videos, with their friends through social network applications based on cloud cargo space on a daily basis. However, while enjoying the convenience of sharing data via cloud storage, users are also progressively more concerned about inadvertent numbers leaks in the cloud. Such data leak, caused by a malicious enemy or a misbehaving cloud operator, can typically lead to serious breaches of personal privacy or business secrets (e.g., the recent high profile incident of star photos being leaked in iCloud). To address users' concerns more potential data leaks in cloud storage, a common draw near is for the data owner to encrypt all the data before uploading them to the cloud, such that later the encrypted data may be retrieve and decrypted by those who have the decryption keys. Such a cloud cargo space is often calling the cryptographic cloud storage [6]. However, the encryption of numbers makes it tough for users to explore and then selectively retrieve only the data containing given keywords. A frequent solution is to employ a searchable encryption (SE) scheme in which the data possessor is required to encrypt potential keywords and upload them to the cloud together with encrypted data, such that, for retrieving data similar a keyword, the user will send the corresponding keyword trapdoor to the cloud for drama search over the encrypted data. First of all, the call for selectively sharing encrypted data with unlike users (e.g., sharing a photo with certain friends in a social network application, or division a business document with certain colleagues on a cloud drive) usually demands unlike encryption keys to be used for different files. However, this imply the digit of keys that need to be strewn to users, both for them to search over the encrypted files and to decrypt the files, will be relative to the number of such files. Such a large number of keys must not only be scattered to users via secure channels, but also be securely stored and managed by the users in their policy. In adding up, a large digit of trapdoors must be generated by users and submit to the cloud in order to perform a keyword search over many files. The implied need for secure communication, storage, and computational complexity may render such a system inefficient and impractical. To hold up searchable group data sharing the main requirements for efficient key management are twofold. First, a data owner only needs to distribute a single amassed key (instead of a group of keys) to a user for giving out any number of files. Second, the user only needs to submit a single aggregate trapdoor (instead of a set of trapdoors) to the cloud for the theater keyword search over any number of shared files. To the finest of our information, the KASE plan wished-for in this paper is the first known scheme that can satisfy both requirements (the key-aggregate cryptosystem [4], which has inspired our employ can satisfy the first must but not the second). Contributions More specifically, our main gifts are as follows.

1) We first identify a common scaffold of key aggregate searchable encryption (KASE) poised of seven polynomial algorithms for safety stricture setup, key generation, encryption, key extraction, trapdoor production, trapdoor adjustment, and trapdoor testing. We then describe both functional and security rations for designing a valid KASE method.

(2) We then instantiate the KASE framework by designing a concrete KASE scheme. After providing detailed constructions for the seven algorithms, we analyze the effectiveness of the scheme, and establish its security through full study



Key aggregate searchable encryption.

(3) We converse various useful issues in building an actual group data sharing system based on the proposed KASE scheme, and price its performance. The evaluation confirms our organism can meet the performance requirements of useful applications

2. Key-AGGREGATE SEARCHABLE ENCRYPTION (KASE)

Development of KASE plan ideas is adapted from papers like key-aggregate cryptosystem scheme [7] for scalable data sharing and Multi-key searchable encryption scheme. This was done to generate a single aggregate encryption key in replacement of many numbers of individual independent keys for each id uploaded by the data owner. Defining this scheme each key which is used for thorough is connected with a particular index of uploaded document. Creation of aggregate key is ready by using the data owner's master-secret key with product of his/her public keys used for encryption. Than confuse server can use single adjusted aggregated trapdoor which was created for each set of paper.

3. KASE FRAMEWORK

It was describe in the above section, this KASE scheme consists of seven algorithms.

(1) Setup : This algorithm is run by blur wine waiter to setup all system parameters. Generate a bilinear mapping based group sharing system, set the highest possible number of documents available with the data owner. Two operations are compute which are random generator calculation and selecting a one way hash function. Cloud server televises the generated system parameter and communal key.

(2) Key generation: This algorithm is run by numbers owner to generate his/her key pair which will be worn for document encryption by the Encrypt algorithm. In this point, we have public key and master covert key along with the generate key

(3) Encrypt : This algorithm is dart by data owner to perform data encryption and also create corresponding cipher texts for all the documents which will be uploaded. For the create the keyword cipher texts, it take the document file index, randomly picks a searchable encryption key for each document and generate a delta information. It will produce a cipher book for a keyword, this generate cipher texts are stored underneath shade server.

(4) Extract : This algorithm is run by data holder and generating an aggregate searchable encryption key and this key is send to all authorized user via a secure communication guide. This Algorithm takes giving as master top secret key and generates an aggregate key as production. Data owner than fling this total key to data users, so that they can perform keyword searching over the joint identification.

(5) Trapdoor: This algorithm is scamper by data user and performs keyword searching by generate trapdoor Only one single aggregate trapdoor is generated for a single keyword which is use for searching.Than data user sends this generate solo trapdoor and subset of in time papers.

(6) Adjust : This algorithm is lope by cloud head waiter and creating right set of trapdoor. It accepts input as system publicly to be had parameter, all documents index in the set plus also solo aggregate trapdoor. It performs adjusting process on the only aggregate trapdoor and output a new right only trapdoor. This produced trapdoor will be used for next Test algorithm for performing keyword search over the shared collection of documents.

(7) Test : This algorithm is run by the cloud server. Cloud server does a series of keyword searching by using the input, which is adjusted trapdoor and creates the delta information which is relevant to subset by using searchable encryption key. Harvest produced will be binary, i.e. true or false values after performing various computation.

(8)Key-aggregate searchable encryption (KASE) method of data sharing, it consists of two types of users: Data owner and Data user. facts owner is uploading n numbers of documents to cloud server which are shared with the data user. Generally, here documents is encrypted by a type pair, this obtained key pair is changed into single aggregate key by using data owner public key and master secret key. The sole aggregate key produced is send to the data user via a secure communication channel. Data user can perform searching over the combined documents by generating single aggregate trapdoor. For each searched word, it can generate an summative trapdoor. If a match is obtained, the common documents are unlocked and return to respective authorized data user.. Key-Aggregate Searchable Encryption Framework of Key-aggregate searchable encryption (KASE), it consists of a data owner generates a single aggregate key which was created by using data holder public key and master secret key for encrypting the shared documents. This single aggregate key produced is send to the data user through a safe and reverberation message channel. Then, data user can perform searching over the shared papers by generating single aggregate trapdoor, submitted this trapdoor to the cloud server. Cloud server performs the adjust algorithm/process by using the aggregate trapdoor over the gathering of documents. Then, test algorithm is performed to make certain that the respective requester has the right to access them. If a match occurs, than shade server will return the entire mutual ID to the respective data user.

4. CONCLUSION

With more mathematical tools, cryptographic schemes are getting more versatile and often engage multiple keys for a single application. In this article, we consider how to "squeeze" top secret keys in public-key cryptosystems which support passing on of top secret key for different cipher text course in cloud cargo space. Our go in the way of is extra flexible than hierarchical type job which container only keep spaces if all key-holders share a like set of privileges. In cloud storage, the amount of cipher texts usually grows rapidly. So we have to keep enough nobody text classes for the future extension. or else, we need to expand the public-key. Although the parameter container be downloaded with cipher texts, it would be better if its size is autonomous of the maximum number of cipher text module.

5. REFERENCES

[1] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing", Proc. IEEE INFOCOM, pp. 534-542, 2010.

[2] R. Lu, X. Lin, X. Liang, and X. Shen, "Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing", Proc. ACM Symp. Information, Computer and Comm. Security, pp. 282-292, 2010.

[3] X. Liu, Y. Zhang, B. Wang, and J. Yan. "Mona: secure multi owner data sharing for dynamic groups in the cloud", IEEE Transactions on Parallel and Distributed Systems, 2013, 24(6): 1182-1191.

[4] C. Chu, S. Chow,W. Tzeng, et al. "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage", IEEE Transactions on Parallel and Distributed Systems, 2014, 25(2): 468-477.

[5] X. Song, D.Wagner, A. Perrig. "Practical techniques for searches on encrypted data", IEEE Symposium on Security and Privacy, IEEE Press, pp. 44C55, 2000.

[6] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky. "Searchable symmetric encryption: improved definitions and efficient constructions", In: Proceedings of the 13th ACM conference on Computer and Communications Security, ACM Press, pp. 79-88, 2006.

[7] P. Van,S. Sedghi, JM. Doumen. "Computationally efficient searchable symmetric encryption", Secure Data Management, pp. 87-100, 2010.

[8] S. Kamara, C. Papamanthou, T. Roeder. "Dynamic searchable symmetric encryption", Proceedings of the 2012 ACM conference on Computer and communications security (CCS), ACM, pp. 965- 976, 2012.

[9] D. Boneh, C. G, R. Ostrovsky, G. Persiano. "Public Key Encryption with Keyword Search", EUROCRYPT 2004, pp. 506C522, 2004.

[10] Y. Hwang, P. Lee. "Public Key Encryption with Conjunctive Keyword Search and Its Extension to a Multi-user System", In:Pairing-Based Cryptography C Pairing 2007, LNCS, pp. 2-22, 2007.

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

[12] 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.

[13] 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.

[14] G. Ateniese, A. D. Santis, A. L. Ferrara, and B. Masucci, "Provably-Secure Time-Bound Hierarchical Key Assignment Schemes," J. Cryptology, vol. 25, no. 2, pp. 243–270, 2012.

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