Global Journal on Application of Data Science and Internet of Things
Vol. 2, Issue 2 - 2018

KEY AGGREGATE SEARCHABLE ENCRYPTION WITH SECURE AND EFFICIENT DATA SHARING IN CLOUD

DR. Y. DASARATHA RAMI REDDY[*], MR. G. SREENIVASA REDDY[**]

ABSTRACT

The proficiency of selectively allocating encrypted data with different users via public cloud storage may greatly affluence security disquiets over unintentional data leaks in the cloud. A crucial defy to scheming such encryption schemes lies in the efficient management of encryption keys. The preferred pliability of partaking any group of selected documents with any group of Users demands altered encryption keys to be used for differentdocuments. However, this also suggests the necessity of securely distributingto users a large number of keys for both encryption and search, and those users will have to securely store the received keys, and submit an equallylarge number of keyword trapdoors to the cloud in order to perform search over the shared data. The practical problem, which is largely abandoned inthe literature, by suggesting the novel concept of key aggregate searchable encryption (KASE) and instantiating the concept through a concrete KASE scheme, in which a data owner only needs to distribute a single key to a user for sharing a large number of documents, and the user only needs to submit a single trapdoor to the cloud for querying the shared documents. The security analysis and performance evaluation both confirm that proposed schemes are provably secure and practically efficient.

KEYWORDS: Authentication, Data Sharing, Cloud Computing, Forward Security, Smart Grid Searchable Encryption, Data Sharing, Cloud Storage, Data Privacy.

INTRODUCTION

Nowadays the storage in the cloud has materialized as a capable answer for suitable and on-demand accesses to huge amounts of information shared over the Internet. Business users are being paying attention by cloud storage due to its several benefits, including lower cost, better agility, and improved resource utilization. Everyday users are also sharing private data, such as photos and videos, with their friends through social network applications based on cloud. On the other hand, while benefiting from the expediency of sharing data through cloud storage, users are also gradually worried about accidental data reveal by the cloud. Such data revealing, will be performed by malicious opponent or a mischievous cloud operator, can habitually direct to severe violation of private data or confidential data regarding bussiness.

[*]Associate Professor, BVSR Engineering College.
[**]Associate Professor, BVSR Engineering College.

To speak about users anxiety over possible data reveal in cloud storage, a general approach is for the data owner to encrypt all the data before uploading them in to the cloud, such that presently the encrypted data may be get back and decrypted by individuals who contains the decryption keys. Such cloud storage is often called the cryptographic cloud storage [6].Though; the encryption of data builds it demanding for users to search and then preferable retrieve only the data including the given keywords. A common solution is to employ a searchable encryption (SE) scheme in which the data owner is required to encrypt potential keywords and upload them to the cloud together with encrypted data, such that, for retrieving data matching a keyword, the user will send the matching keyword to the cloud to react for the search over the encrypted data. Even though merging a searchable encryption Scheme with cryptographic cloud storage can accomplish the essential security needs of a cloud storage, executing such a system for large scale application relating huge number of users and large number of files may still be delayed by realistic issues relating the well-organized management of encryption keys, which, to the finest of our knowledge. Primarily, the want for selectively sharing encrypted data with different users usually demands differentencryption keys to be used for different files. On the other hand, this involves the number of keys that need to be spread 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 spread to users via secure channels, but also be securely stored and handled by the users in their devices. The implicit requirement for secure communication, storage, and computational difficulty may cause system ineffectiveness. In this paper, we propose the novel concept of key- aggregate searchable encryption (KASE), and instantiating the concept through a concrete KASE method. The proposed KASE scheme relates to any cloud storage that supports the searchable group data sharing feature, which means any user may prefer to distribute a group of files which are selective with a group of selected users, while permitting the final to carry out keyword search above the earlier. To maintainsearchable group data sharing the main needs for efficient key management are double. Primarily, a data owner wants to allocate a single aggregate key (instead of a group of keys) to a user for sharing any number of files. Subsequent, the user needs to submit a single aggregate trapdoor to the cloud for performing keyword search over any quantity of shared files. KASE scheme can assure both requests.

RELATED WORK

1. Primarily we describe a common structure of key aggregate searchable encryption (KASE) collected from several polynomial algorithms for security parameter setup, key generation, encryption, key extraction, trapdoor generation, trapdoor adjustment, and trapdoor testing. We then explain both functional and security requirements for scheming a valid KASE scheme.
2. We then instantiate the KASE skeleton by Scheming a concrete KASE scheme. After giving the full structure for the algorithms, we analyze the effectiveness of the scheme, and set up its safety through detailed analysis.

Searchable Encryption

Searchable encryption schemes categorized into two categories, i.e., searchable symmetric encryption (SSE) and public key encryption with keyword search (PEKS). Both SSE and PEKS can be described as the tuple SE= (Setup, Encrypt, Trapdoor (Trpdr), Test):

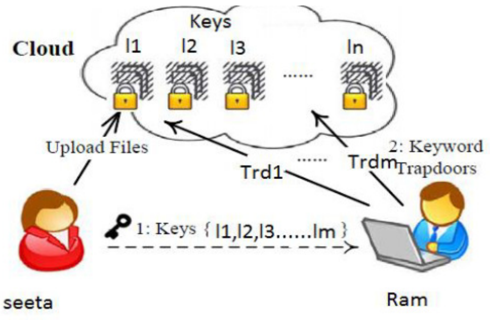
1. **Setup** (1λ): This algorithm is run by the owner to set up the scheme. It takes as input a security parameter 1λ and outputs the necessary keys.

2. **Encrypt** (l;n): This algorithm is run by the owner to encrypt the data and generate its keyword ciphertexts. It takes as input the data n, owner's necessary keys including searchable encryption key l and data encryption key, outputs data ciphertext and keyword ciphertexts Cn.
3. **Trpdr** (l;x): This algorithm is run by a user to generate a trapdoor Trd for a keyword w using key l.
4. **Test** (Trd, Cn): This algorithm is run by the cloud server to perform a keyword search over encrypted data. It takes as input trapdoor Trd and the keyword ciphertextsCn, outputs whether Cn contains the specified keyword. For exactness, it is required that, for a message n containing keyword x and a searchable encryption key l, if (Cn-Encrypt (l;n) and Tr Trpdr(l;x)), then Test(Trd, Cn)=true.

THE KEY-AGGREGATE SEARCHABLE ENCRYPTION

(KASE) CONSTRUCTIONS

In this paper, we propose the novel approach of Key-aggregate searchable encryption (KASE) as a enhanced solution, as depicted in Fig. 1(b). , in KASE, seeta needs to issue a single aggregate key, instead of {ki}mi=1 for sharing m documents with Ram, and ram needs to issue a single aggregate trapdoor, instead of { Tri }mi=1, to thecloud server. The cloud server can utilize this aggregate trapdoor and some public data to carry out keyword search and revisit the result to Ram. As a result, in KASE, the delegation of keyword search right can be achieved by sharing the single aggregate key.



(a) Traditional approach

To design a key-aggregate searchable encryption method under which any subset of the keyword ciphertexts from any set of documents is searchable with a constant- size trapdoor generated by a constant size aggregate key.

THE KASE CONSTRUCTION

The KASE construction is composed of several algorithms. Specially, to set up the method, the cloud server would generate public parameters of the system during the **Setup** algorithm, and these public parameters can be reprocess by dissimilar data owners to distribute their files. For each data owner, they should produce a public/master-secret key pair through the **Keygen** algorithm. Keywords of each document can be encrypted through the **Encrypt** algorithm with the exclusive searchable encryption key. In that case, the data owner can apply the master-secret key to produce an aggregate searchable encryption key for a group of selected documents through the **Extract** algorithm. The aggregate key can be spread securely to approve users who need to access those documents. After that, as shown in Fig.2, an certified user can create a keyword trapdoor via the **Trapdoor** algorithm using thisaggregate key, and submit the trapdoor to the cloud. After getting the trapdoor, to carry out the keyword search over the particular set of documents, the cloud server will run the **Adjust** algorithm to produce the right trapdoor for each document, and then run the **Test** algorithm to test whether the document contains the keyword.

This construction is summarized in the following.

1. **Setup** (1λ, n): This algorithm is run by the cloud service provider to set up the scheme. On input of a security parameter 1λ and the maximum possible number n of documents which belongs to a data owner, it outputs the public system parameter params.
2. **Keygen**: This algorithm is run by the data owner to generate a random key pair (pk,msk).
3. **Encryp t**(pk, i): This algorithm is run by the data owner to encrypt the i-thdocument and generate its keywords' ciphertexts. For each document, this algorithm will create a delta Дi for its searchable encryption key ki. On input of the owner's public key pk and the file index i, this algorithm outputs data ciphertext and keyword ciphertexts Ci.
4. **Extract** (msk, S): This algorithm is run by the data owner to generate an aggregate searchable encryption key for hand over the keyword search right for a certain set of documents to other users. It takes as input the owner's master-secret key msk and aset S which enclose the directory of documents, and then outputs the aggregate key kagg.
5. **Trapdoor** (kagg, x): This algorithm is run by the user who has the aggregate key to perform a search. It takes as input the aggregate searchable encryption key kagg and a keyword w, then outputs only one trapdoor Trd.
6. **Adjust** (params, i, S, Trd): this algorithm is run by cloud server to adjust the aggregate trapdoor to generate the right trapdoor for each different document. It takes as input the system public parameters params, the set S of documents' indices, the index i of target document and the aggregate trapdoor Tr, then outputs each trapdoor Tri for the i-th target document in S.
7. **Test** (Tri, i): this algorithm is run by the cloud server to perform keyword search over an encrypted document. It takes as input the trapdoor Tri and the document indexi, then outputs true or false to denote whether the document doci contains the keywordw.

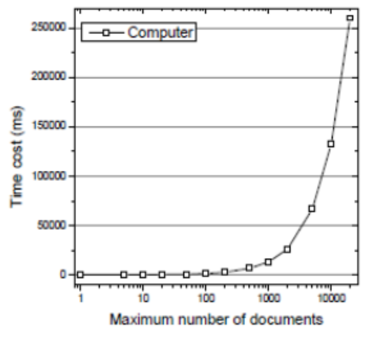
EVALUATION OF KASE ALGORITHMS

Considering that the algorithms including KASE **Setup**, KASE **Adjust** and KASE. **Test** are only run in the cloud server, only the execution times in computer are tested. As shown in Fig.4, we can see that:

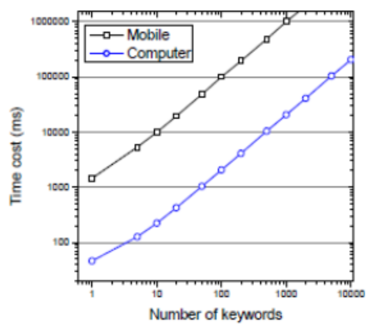
1. The execution time of KASE **Setup** is linear in the maximum number of documents belonging to one owner, and when the maximum number grows up to 20000, it is reasonable that KASE **Setup** algorithm only needs 259 second.
2. The execution time of KASE **Encrypt** is linear in the number of keywords, and when the number grows up to 10000, KASE **Encrypt** algorithm only needs 206 second in computers, but 10018 second in mobile devices. Therefore, we can draw two conclusions; one is that it is not feasible to upload document with lots of keywords using a mobile phone; the other is that the keyword search with pairing computation can be executed quickly in computers now.
3. The execution time of KASE **Extract** is linear in the number of shared documents, and when the number grows up to 10000, KASE **Extract** algorithm only needs 132 second in computer, but 2430 second in mobile devices. Because the KASE **Extract** always runs along with the KASE **Encrypt**, it is not suggested to be executed in the mobile devices.
4. The execution time of KASE **Trapdoor** is a constant, i.e., 0.01 second in computerand 0.25 second in mobile devices. In fact, the mathematical operation in KASE **Trapdoor** is the once multiplication in G, so that the keyword search can be performed efficiently

in both mobile devices and computer. Compared with other schemes, there is a significant improvement in our scheme.

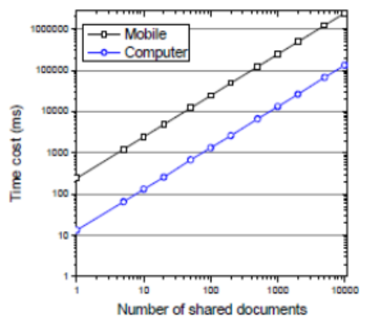
5. The execution time of KASE **Adjust** is linear in the number of documents. In fact, it can be Improved in the practical application, and the details are shown in section6.4.
6. The execution time of KASE **Test** is linear in the number of keyword ciphertexts. In fact, the Mathematical operation in KASE **Test** is twice as much as the pairing computations. When the Number grows up to 20000, it will take 467 second.



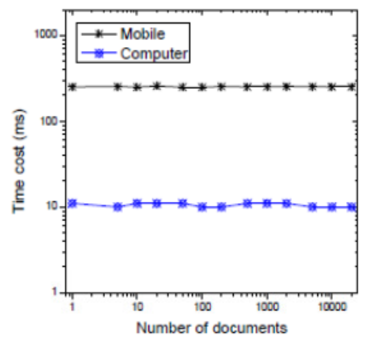
(a) Time cost of Setup



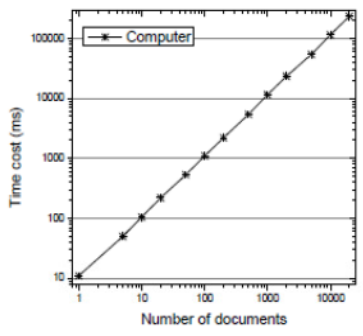
(b) Time cost of Encrypt



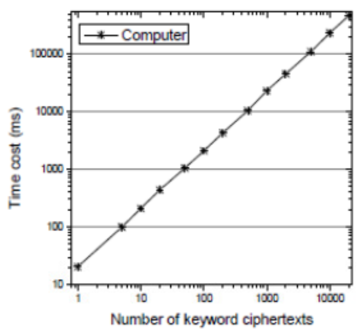
(c) Time cost of Extract



(d) Time cost of Trapdoor



(e) Time cost of Adjust



(f) Time cost of Test

CONCLUSION & FUTURE ENHANCEMENT

The projected concept of key-aggregate searchable encryption (KASE) and erecting a tangible KASE scheme can provide an resourceful solution to edifice practical data sharing system based on public cloud storage. In a KASE scheme, the owner needs to dispense a single key to a user when paying a lot of documents with the user, and the user needs to submit a single trapdoor when they queries over all documents sharedby the same owner. On the other hand, if a user wants to question over documents shared by multiple owners, that user must yield many trapdoors to the cloud. The future enhancement for this proposed work is to find out how to decline the number of trapdoors under multi-owners setting by conquering the security.

REFERENCES

[1] S. Yu, C. Wang, K. Ren, and W. Lou, chieving 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 multiowner 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 Searchand Its Extension to a Multi-user System, In: Pairing-Based Cryptography C Pairing 2007, LNCS, pp. 2-22, 2007.

[11] J. Li, Q. Wang, C. Wang. Fuzzy keyword search over encrypted data in cloud computing, Proc. IEEE INFOCOM, pp. 1-5, 2010.

[12] C. Bosch, R. Brinkma, P. Hartel. Conjunctive wildcard search over encrypted data, Secure Data Management. LNCS, pp. 114-127, 2011.

[13] C. Dong, G. Russello, N. Dulay. Shared and searchable encrypted data for untrusted servers, Journal of Computer Security, pp. 367-397, 2011.

[14] F. Zhao, T. Nishide, K. Sakurai. Multi-User Keyword Search Scheme for Secure Data Sharing with Fine-Grained Access Control. Information Security and Cryptology, LNCS, pp. 406-418, 2012.

[15] J. W. Li, J. Li, X. F. Chen, et al. Efficient Keyword Search over Encrypted Data with Fine-Grained Access Control in Hybrid Cloud, In: Network and System Security 2012, LNCS, pp. 490-502, 2012.

[16] J. Li, K. Kim. Hidden attribute-based signatures without anonymity revocation, Information Sciences, 180(9): 1681-1689, Elsevier, 2010.

[17] X.F. Chen, J. Li, X.Y. Huang, J.W. Li, Y. Xiang. Secure Outsourced Attribute- based Signatures, IEEE Trans. on Parallel and Distributed Systems, DOI.ieeecomputer society.org/10.1109/TPDS.2013.180, 2013.

[18] J.Li, X.F. Chen, M.Q. Li, J.W. Li, P. Lee, Wenjing Lou. Secure Deduplication with Efficient and Reliable Convergent Key Management, IEEE Transactions onParallel and Distributed Systems, 25(6): 1615-1625, 2014.

[19] Z. Liu, Z. Wang, X. Cheng, et al. Multi-user Searchable Encryption with Coarser-Grained Access Control in Hybrid Cloud, Fourth International Conference on Emerging Intelligent Data and Web Technologies (EIDWT), IEEE, pp. 249-255, 2013.

[20] Dr. V. Goutham and M. Tejaswini, A Denial of Service Strategy To Orchestrate Stealthy Attack Patterns In Cloud Computing, International Journal of Computer Engineering and Technology, 7(3), 2016, pp. 179-186.

[21] Kuldeep Mishra, Ravi Rai Chaudhary and Dheresh Soni, A Premeditated CDM Algorithm In Cloud Computing Environment For FPM, International Journal of Computer Engineering and Technology, 4(4), 2013, pp. 213-223.

[22] Supriya Mandhare, Dr. A.K.Sen and Rajkumar Shende, A Proposal On Protecting Data Leakages In Cloud Computing, International Journal of Computer Engineering and Technology, 6(2), 2015, pp. 45-53.

[23] Sujay Pawar and Prof. Mrs. U. M. Patil, A Survey On Secured Data Outsourcing In Cloud Computing, International Journal of Computer Engineering and Technology, 4(3), 2013, pp. 70-76.

[24] N. A. Joshi, Dynamic Load Balancing In Cloud Computing Environments, International Journal of Computer Engineering and Technology, 4(3), 2013, pp. 70-76.

[25] C. Wang, Q. Wang, K. Ren, and W. Lou, Privacy-Preserving.