ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Group Data Sharing via Cloud Storage using Key Aggregate Encryption Technique

Dileep Dias
M.Tech Student,
Department of CSE,
Malla Reddy College of Engineering and Technology.

P.Bikshapathy
Assistant Professor,
Department of CSE,
Malla Reddy College of Engineering and Technology.

ABSTRACT

The capability of selectively sharing encrypted data with different users via public cloud storage may greatly ease security concerns over inadvertent data leaks in the cloud. A key challenge to designing such encryption schemes lies in the efficient management of encryption keys. The desired flexibility of sharing any group of selected documents with any group of users demands different encryption keys to be used for different documents. However, this also implies the necessity of securely distributing to 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 equally large number of keyword trapdoors to the cloud in order to perform search over the shared data. The implied need for secure communication, storage, and complexity clearly renders the approach impractical. In this paper, we address this practical problem, which is largely neglected in the literature, by proposing 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 our proposed schemes are provably secure and practically efficient.

INTRODUCTION

Cloud storage has emerged as a promising solution for providing ubiquitous, convenient, and on-demand accesses to large amounts of data shared over the Internet. Today, millions of users are sharing personal data, such as photos and videos, with their friends through social network applications based on cloud storage on a daily basis. Business users are also being attracted by cloud storage due to its numerous benefits, including lower cost, greater agility, and better resource utilization.

However, while enjoying the convenience of sharing data via cloud storage, users are also increasingly concerned about inadvertent data leaks in the cloud. Such data leaks, caused by a malicious adversary or a misbehaving cloud operator, can usually lead to serious breaches of personal privacy or business secrets (e.g., the recent high profile incident of celebrity photos being leaked in iCloud). To address users' concerns over potential data leaks in cloud storage, a common approach is for the data owner to encrypt all the data before uploading them to the cloud, such that later the encrypted data may be retrieved and decrypted by those who have the decryption keys. Such cloud storage is often called the cryptographic cloud storage. However, the encryption of data makes it challenging for users to search and then selectively retrieve only the data containing 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 corresponding keyword trapdoor to the cloud for performing search over the encrypted data.

Although combining a searchable

Encryption scheme with cryptographic cloud storage can achieve the basic security requirements of cloud storage, implementing such a system for large scale applications involving millions of users and billions of files may still be hindered by practical issues involving the efficient management of encryption keys, which, to the best of our knowledge, are largely ignored in the literature. First of all, the need for selectively sharing encrypted data with different users (e.g., sharing a photo with certain friends in a social network application, or sharing a business document with certain colleagues on a cloud drive) usually demands different encryption keys to be used for different files. However, this implies the number of keys that need to be distributed to users, both for them to search over the encrypted files and to decrypt the files, will be proportional to the number of such files. Such a large number of keys must not only be distributed to users via secure channels, but also be securely stored and managed by the users in their devices. In addition, a large number of trapdoors must be generated by users and submitted 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.

EXISTING SYSTEM:

- There is a rich literature on searchable encryption, including SSE schemes and PEKS schemes. In contrast to those existing work, in the context of cloud storage, keyword search under the multi-tenancy setting is a more common scenario. In such a scenario, the data owner would like to share a document with a group of authorized users, and each user who has the access right can provide a trapdoor to perform the keyword search over the shared document, namely, the “multi-user searchable encryption” (MUSE) scenario.
- Some recent work focus to such a MUSE scenario, although they all adopt single-key combined with access control to achieve the goal.
- In MUSE schemes are constructed by sharing the document’s searchable encryption key with all users who can access it, and broadcast encryption is used to achieve coarse-grained access control.
- In attribute based encryption (ABE) is applied to achieve fine-grained access control aware keyword search. As a result, in MUSE, the main problem is how to control which users can access which documents, whereas how to reduce the number of shared keys and trapdoors is not considered.

DISADVANTAGES OF EXISTING SYSTEM:

- Unexpected privilege escalation will expose all
- It is not efficient.
- Shared data will not be secure.

PROPOSED SYSTEM:

- In this paper, we address this challenge by proposing the novel concept of key-aggregate searchable encryption (KASE), and instantiating the concept through a concrete KASE scheme.
- The proposed KASE scheme applies to any cloud storage that supports the searchable group data sharing functionality, which means any user may selectively share a group of selected files with a group of selected users, while allowing the latter to perform keyword search over the former.
- To support searchable group data sharing the main requirements for efficient key management are twofold. First, a data owner only needs to distribute a single aggregate key (instead of a group of keys) to a user for sharing any number of files. Second, the user only needs to submit a single aggregate trapdoor (instead of a group of trapdoors) to the cloud for performing keyword search over any number of shared files.
- We first define a general framework of key aggregate searchable encryption (KASE) composed of seven polynomial algorithms for security parameter setup, key generation, encryption, key extraction, trapdoor generation, trapdoor adjustment, and trapdoor testing. We then describe both functional and security requirements for designing a valid KASE scheme.
- We then instantiate the KASE framework by designing a concrete KASE scheme. After providing detailed constructions for the seven algorithms, we

analyze the efficiency of the scheme, and establish its security through detailed analysis.

- ❖ We discuss various practical issues in building an actual group data sharing system based on the proposed KASE scheme, and evaluate its performance. The evaluation confirms our system can meet the performance requirements of practical applications.

ADVANTAGES OF PROPOSED SYSTEM:

- ❖ It is more secure.
- ❖ Decryption key should be sent via a secure channel and kept secret.
- ❖ It is an efficient public-key encryption scheme which supports flexible delegation.
- ❖ To the best of our knowledge, the KASE scheme proposed in this paper is the first known scheme that can satisfy requirements.

IMPLEMENTATION MODULES:

1. Data Owner
2. Network Storage
3. Encrypted Aggregate Key and Searchable Encryption key Transfer
4. Trapdoor Generation
5. File User

MODULES DESCRIPTION:

Data Owner:

In this module we executed by the data owner to setup an account on an un trusted server. On input a security level parameter 1^λ and the number of ciphertext classes n (i.e., class index should be an integer bounded by 1 andn), it outputs the public system parameter param, which is omitted from the input of the other algorithms for brevity.

Network Storage (Drop box):

With our solution, Alice can simply send Bob a single aggregate key via a secure e-mail. Bob can download the encrypted photos from Alice's Dropbox space and then use this aggregate key to decrypt these encrypted photos. In this Network Storage is entrusted third party server or dropbox.

Encrypted Aggregate Key and Searchable Encrypted key Transfer:

The data owner establishes the public system parameter via Setup and generates a public/master-secretkey pair via KeyGen. Messages can be encrypted via Encrypt by anyone who also decides what ciphertext class is associated with the plaintext message to be encrypted. The data owner can use the master-secret to generate an aggregate decryption key for a set of cipher text classes via Extract. The generated keys can be passed to delegates securely (via secure e-mails or secure devices) 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

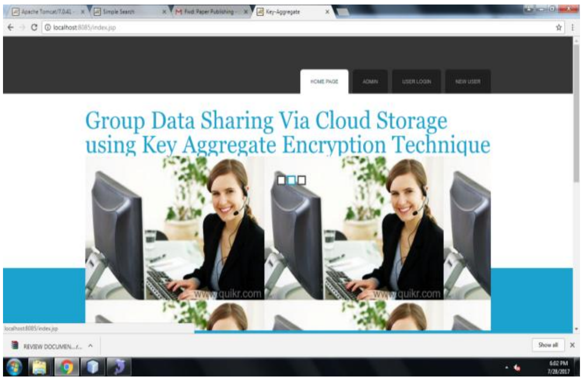
Trapdoor generation

Trapdoor generation algorithm is run by the user who has the aggregate key to perform search. It takes as input the aggregate searchable encryption key kagg and a keyword w, then outputs only one trapdoor Tr.

File User:

The generated keys can be passed to delegates securely (via secure e-mails or secure devices) finally; any user with the Trapdoor keyword generation process can decrypt any ciphertext provided that the ciphertext class is contained in the Encrypted aggregate key and Searchable Encrypted key via Decrypt.

SCREEN SHOTS:



Group Data Sharing Via Cloud Storage using Key Aggregate Encryption Technique

Fig: Home Page

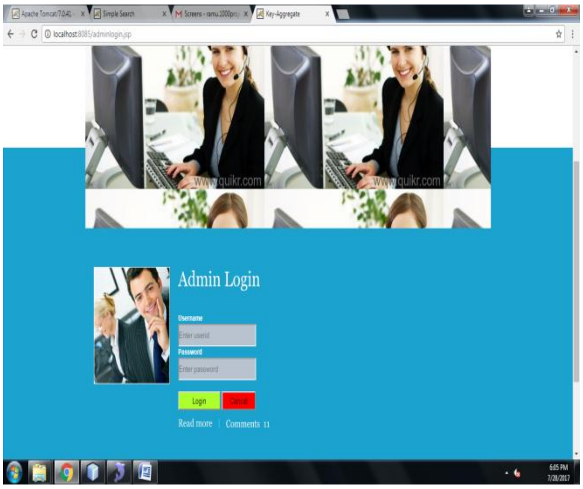


Fig: Admin Login Page

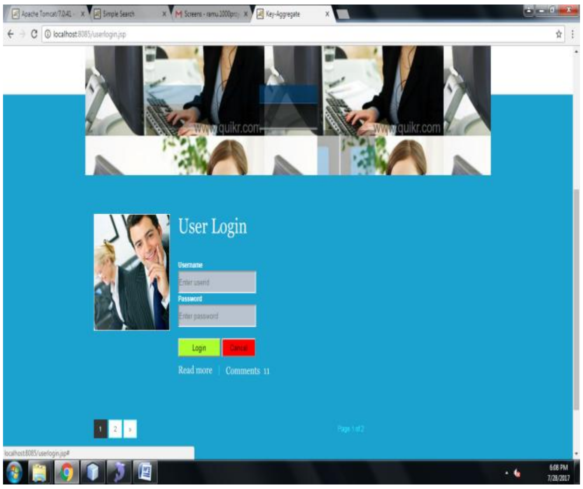


Fig: User Login Page

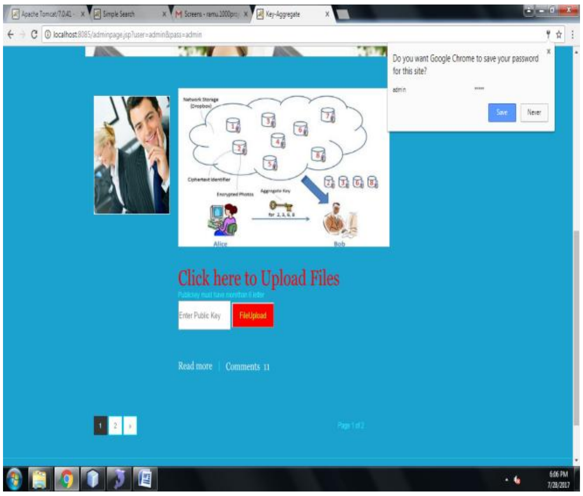


Fig: File Upload Page

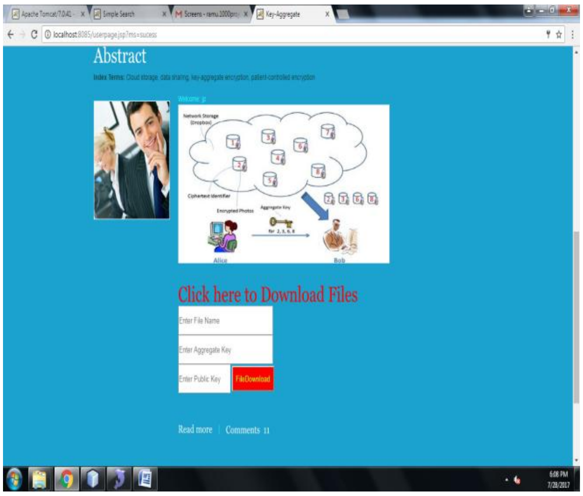
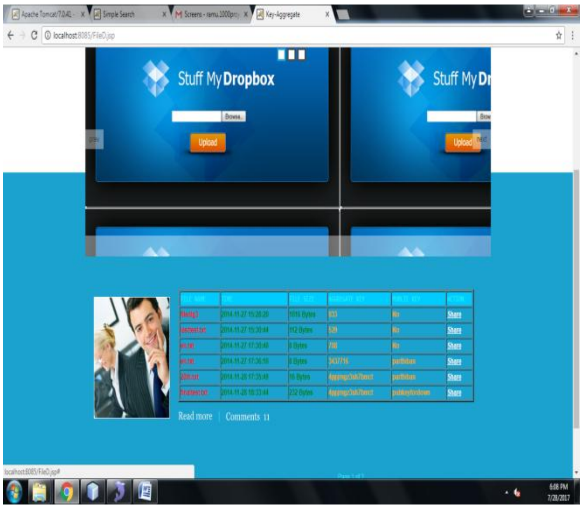

localhost:8085/userpage.jsp?ms=sucess
Abstract
Index Terms: Cloud storage, data sharing, key-aggregate encryption, patient-controlled encryption
Network Storage (Dropbox)
Ciphertext Identifier
Encrypted Photos
Aggregate Key
for 2, 3, 6, 8
Alice
Bob
Click here to Download Files
Enter File Name
Enter Aggregate Key
Enter Public Key
FileDownload
Read more | Comments 11

Fig: File Download Page



[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	2014.11.27 15:28:20	1016 Bytes	833	No	Share
[illegible]	2014.11.27 15:30:44	112 Bytes	520	No	Share
[illegible]	2014.11.27 17:30:40	0 Bytes	708	No	Share
[illegible]	2014.11.27 17:36:50	0 Bytes	3437716	parthiban	Share
[illegible]	2014.11.28 17:35:40	16 Bytes	[illegible]	parthiban	Share
[illegible]	2014.11.28 18:33:44	232 Bytes	[illegible]	publickeyfordown	Share

Fig: File Details Page

CONCLUSION

How to protect users' data privacy is a central question of cloud storage. With more mathematical tools, cryptographic schemes are getting more versatile and often involve multiple keys for a single application. In this paper, we consider how to "compress" secret keys in public-key cryptosystems which support delegation of secret keys for different ciphertext classes in cloud storage. No matter which one among the power set of classes, the delegatee can always get an aggregate key of constant size. Our approach is more flexible than hierarchical key assignment which can only save spaces if all key-holders share a similar set of privileges. A limitation in our work is the predefined bound of the number of maximum ciphertext classes. In cloud storage, the number of ciphertexts usually grows rapidly. So we have to reserve enough ciphertext classes for the future extension.

Although the parameter can be downloaded with ciphertexts, it would be better if its size is independent of the maximum number of ciphertext classes. On the other hand, when one carries the delegated keys around in a mobile device without using special trusted hardware, the key is prompt to leakage, designing a leakage-resilient cryptosystem yet allows efficient and flexible key delegation is also an interesting direction.

REFERENCES

[1] S.S.M. Chow, Y.J. He, L.C.K. Hui, and S.-M. Yiu, "SPICE – Simple Privacy-Preserving Identity-Management for Cloud Environment," Proc. 10th Int'l Conf. Applied Cryptography and Network Security (ACNS), vol. 7341, pp. 526-543, 2012.

[2] L. Hardesty, Secure Computers Aren't so Secure. MIT press, http://www.physorg.com/news176107396.html, 2009.

[3] 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, Feb. 2013.

[4] B. Wang, S.S.M. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," Proc. IEEE 33rd Int'l Conf. Distributed Computing Systems (ICDCS), 2013.

[5] S.S.M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R.H. Deng, "Dynamic Secure Cloud Storage with Provenance," Cryptography and Security, pp. 442-464, Springer, 2012.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT '03), pp. 416-432, 2003.

[7] M.J. Atallah, M. Blanton, N. Fazio, and K.B. Frikken, "Dynamic and Efficient Key Management for Access Hierarchies," ACM Trans. Information and System Security, vol. 12, no. 3, pp. 18:1-18:43, 2009.

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

[9] F. Guo, Y. Mu, Z. Chen, and L. Xu, "Multi-Identity Single-Key Decryption without Random Oracles," Proc. Information Security and Cryptology (Inscrypt '07), vol. 4990, pp. 384-398, 2007.

[10] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," Proc. 13th ACM Conf. Computer and Comm. Security (CCS '06), pp. 89-98, 2006.

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

[12] G.C. Chick and S.E. Tavares, "Flexible Access Control with Master Keys," Proc. Advances in Cryptology (CRYPTO '89), vol. 435, pp. 316-322, 1989.