ISSN: 2395-7639

International Journal of Multidisciplinary Research in Science, Engineering, Technology & Management (IJMRSETM)

(A Monthly, Peer Reviewed Online Journal)

Visit: www.ijmrsetm.com

Volume 3, Issue 5, May 2016

Secure and Searchable Group Data Sharing Via Public Cloud Storage to Reduce Data User Complexity

Bhuvanesvari G

Cambridge Institute of Technology, Bangalore, India

ABSTRACT: Cloud is increasingly used by people to save their data. The users also share the data uploaded to other users. For the security of data users encrypt the data before uploading the data to cloud and for the users who wanted access get the key from the owner and use it to decrypt the data. If the user uploads many files and he wanted to share to many users, the number of keys to use and share increases and the key management becomes difficult. In [1], authors proposed a key aggregate scheme where the any number of files belonging to single data owner is encrypted with single key and the user who wanted to access gets a aggregated key and using this aggregated key they can access the data based on categorization of user belonging to which group. Though this scheme is able to reduce the key complexity for owner by using a single key to encrypt any number of files, the user has to remember number of keys of different owners and it becomes difficult for him. In this paper we explore the key management problem for user and propose a effective solution for it.

KEYWORDS: Key aggregation, trapdoor, data owner, data user

I. INTRODUCTION

Cloud data storage has become a popular trend among smartphone users. Since smart phone users have limited storage capacity they backup their data on cloud and also they share the cloud data to users at ease with emergence of cloud data sharing options in the mailing and messaging systems. With emergence of cloud for data sharing, security has become a concern, because any one can take the data stored on cloud. Recent trends of personnel information stored on cloud storage getting leaked have affected the user of cloud and users are looking for secure cloud storages at the time with minimal effort for key management. Many solutions are proposed like ABE, CP-ABE etc for encrypting the data with key and managing the key. Most of the schemes see only a peer to peer way of sharing data. For many online users are looking for a group data sharing where a file can be shared to a group of users. In this kind of environment for each file shared by users to N group, the number of keys to be used by minimum , as it is difficult for owner to remember each key he use to encrypt file for different group of users. In [1], authors have proposed a key aggregate based group data sharing solution. In this solution 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. The solution is designed from data owner point of view , but from the data user point of view, the solution is not efficient, In this paper, we deal with this problem and propose a effective solution to reduce the data user complexity. Our solution is extension of [1], to reduce the data user complexity in keeping with multiple keys for different owners.

II. RELATED WORK

In this section we survey the existing solutions for group data sharing. Cryptographic key assignment schemes (e.g., [11], [12], [13], [14]) aim to minimize the expense in storing and managing secret keys for general cryptographic use. Utilizing a tree structure, a key for a given branch can be used to derive the keys of its descendant nodes (but not the other way round). Just granting the parent key implicitly grants all the keys of its descendant nodes. Sandhu [15] proposed a method to generate a tree hierarchy of symmetrickeys by using repeated evaluations of pseudorandom function/block-cipher on a fixed secret. The concept can be generalized from a tree to a graph. More advanced cryptographic key assignment schemes support access policy that can be modeled by an acyclic graph or a cyclic graph [16], [17], [7]. Most of these schemes produce keys for symmetric-key cryptosystems, even though the key derivations may require modular arithmetic as used in public-key cryptosystems, which are generally more expensive than "symmetric-key operations" such as pseudorandom function. We take the tree structure as an example. Alice can first classify the ciphertext classes according to their subjects like Figure 3. Each node in the tree represents a secret key, while the leaf nodes represents the keys for individual ciphertext classes. Filled circles represent the keys for the classes to be delegated and circles circumvented by dotted lines represent the keys to be granted. Note that every key of the non-leaf node can derive the keys of its descendant nodes.

In general, hierarchical approaches can solve the problem partially if one intends to share all files under a certain branch in the hierarchy. On average, the number of keys increases with the number of branches.It is unlikely to come up with a hierarchy that can save the number of total keys to be granted for all individuals (which can access a different set of leaf-nodes) simultaneously

Motivated by the same problem of supporting flexible hierarchy in decryption power delegation (but in symmetric-key setting), Benaloh et al. [8] presented an encryption scheme which is originally proposed for concisely transmitting large number of keys in broadcast scenario [18]. The construction is simple and we briefly review its key derivation process here for a concrete description of what are the desirable properties we want to achieve. The derivation of the key for a set of classes (which is a subset of all possible ciphertext classes) is as follows. A composite modulus N = p · q is chosen where p and q are two large random primes. A mastersecret key Y is chosen at random from Z∗ N. Each class is associated with a distinct prime ei. All these prime numbers can be put in the public system parameter5 A constant-size key for set S0 can be generated (with the knowledge of φ(N)) as kS0 = Y 1/Qj∈S0(ej) mod N. For those who have been delegated the access rights for S where S0 ⊂ S, kS0 can be computed by kQj∈S\S0 (ej) S . As a concrete example, a key for classes represented by e1,e2,e3 can be generated as Y 1/(e1·e2·e3), from which each of Y 1/e1, Y 1/e2, Y 1/e3 can easily be derived (while providing no information about keys for any other class, say, e4). This approach achieves similar properties and performances as our schemes. However, it is designed for the symmetric-key setting instead. The encryptor needs to get the corresponding secret keys to encrypt data, which is not suitable for many applications. Since their method is used to generate a secret value rather than a pair of public/secret keys, it is unclear how to apply this idea for public-key encryption scheme. Finally, we note that there are schemes which try to reduce the key size for achieving authentication in symmetric-key encryption, e.g., [19]. However, sharing of decryption power is not a concern in these schemes.

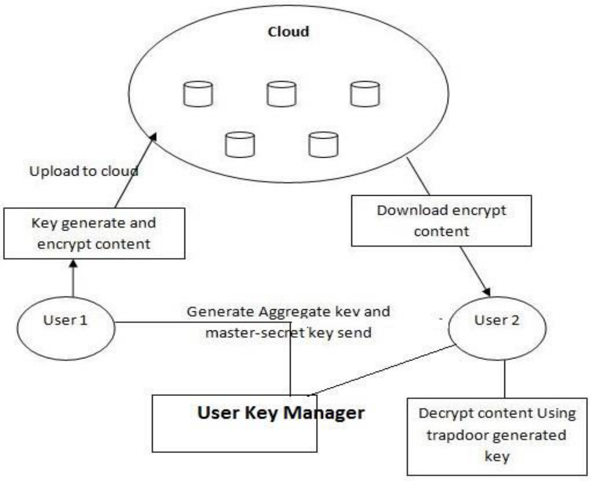
Identity-based encryption (IBE) (e.g., [20], [21], [22]) is a type of public-key encryption in which the public-key of a user can be set as an identity-string of the user (e.g., an email address). There is a trusted party called private key generator (PKG) in IBE which holds a master-secret key and issues a secret key to each user with respect to the user identity. The encryptor can take the public parameter and a user identity to encrypt a message. The recipient can decrypt this ciphertext by his secret key. Guo et al. [23], [9] tried to build IBE with key aggregation. One of their schemes [23] assumes random oracles but another [9] does not. In their schemes, key aggregation is constrained in the sense that all keys to be aggregated must come from different "identity divisions". While there are an exponential number of identities and thus secret keys, only a polynomial number of them can be aggregated. Most importantly, their key-aggregation [23], [9] comes at the expense of O(n) sizes for both ciphertexts and the public parameter, where n is the number of secret keys which can be aggregated into a constant size one. This greatly increases the costs of storing and transmitting ciphertexts, which is impractical in many situations such as shared cloud storage. As we mentioned, our schemes feature constant ciphertext size, and their security holds in the standard model. In fuzzy IBE [21], one single compact secret key can decrypt cipher texts encrypted under many identities which are close in a certain metric space, but not for an arbitrary set of identities and therefore it does not match with our idea of key aggregation

III. PROBLEM DEFINITION

Given a group data sharing and searchable platform where many they can be multiple data owners and data users. Data owners provide single trapdoor access key for users to access the secure data and also search it. When data user receives these multiple data keys, data user must be able to manage it easily.

IV. PROPOSED SOLUTION

We use the user key management as the service on cloud. It can be hosted in the same cloud as data storage or as separate cloud.


Cloud
Upload to cloud
Key generate and encrypt content
Download encrypt content
User 1
Generate Aggregate kev and master-secret key send
User 2
User Key Manager
Decrypt content Using trapdoor generated key

.

The user key manager module runs on cloud as un trusted and the data stored must be made secure.

When the data owner releases the key, all the key are stored in the use key manager module. The key manager is a separate component in cloud and the service is available as key manager as service on the cloud platform. Each data owner key for their files shared when sent to users are redirected and stored in the user key manager module.

The user can request the key to be used for particular data owner from the user key manager module. Since the user key manager module is hosted on cloud, the keys must be secured. To secure the keys, the keys must be encrypted and stored in the cloud.

The data user generates a symmetric key and it is known only to it. Whenever it receives the trap door key from the data owners, it encrypts the key and then stores in user key manager. When the user requires the trap door key, it queries the user key manager to get the key and then decrypt it with its key to get the trapdoor key.

Our solution has following advantages

.

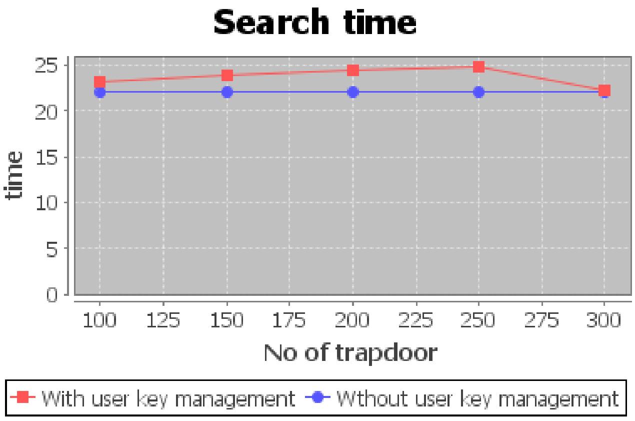
1 The user no need to store remember or store multiple trap door keys. It needs to remember only one key
2. Since the user key manager is available as cloud service, the user can get the trapdoor at any place and the solution is portable.

3. The trapdoor key stored on key manger is fully secure with the encryption done by user with his key.

V. SIMULATION RESULTS

We measured the time taken to search with and without user key management solution. Through this experiment, we measure the extra time added to search for different number of trap door keys. The user key manger module is hosted on Amazon S2 cloud.

The results are shown below



The extra time added for user key management is very low for the added advantage of user management of key. So our solution is acceptable from the performance point of view

VI. Conclusion and Future Work

In this work, we have key management extension for reducing the data user key management complexity and though experiment we have proved that our solution adds only little overhead to existing key aggregate based search method.

.

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," in Applied Cryptography and Network Security - ACNS 2012, ser. LNCS, vol. 7341. Springer, 2012, pp. 526–543.

[2] L. Hardesty, "Secure computers aren't so secure," MIT press, 2009, http://www.physorg.com/news176107396.html.

[3] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, PrivacyPreservingPublicAuditingforSecureCloudStorage,"IEEETrans. Computers, vol. 62, no. 2, pp. 362–375, 2013.

[4] B. Wang, S. S. M. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," in International Conference on Distributed Computing Systems - ICDCS 2013. IEEE, 2013.

[5] S. S. M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R. H. Deng, "Dynamic Secure Cloud Storage with Provenance," in Cryptography and Security: From Theory to Applications - Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday, ser. LNCS, vol. 6805. Springer, 2012, pp. 442–464.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," in ProceedingsofAdvancesinCryptology-EUROCRYPT'03,ser.LNCS, vol. 2656. Springer, 2003, pp. 416–432.

[7] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, "Dynamic and Efficient Key Management for Access Hierarchies," ACM Transactions on Information and System Security (TISSEC), vol. 12, no. 3, 2009.

[8] 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.

[9] 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.

[10] 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.

[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 KnowledgeandDataEngineering(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.

[16] 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.

[17] Q. Zhang and Y. Wang, "A Centralized Key Management Scheme for Hierarchical Access Control," in Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '04). IEEE, 2004, pp. 2067–2071.

[18] J. Benaloh, "Key Compression and Its Application to Digital Fingerprinting," Microsoft Research, Tech. Rep., 2009.

[19] B. Alomair and R. Poovendran, "Information Theoretically Secure Encryption with Almost Free Authentication," J. UCS, vol. 15, no. 15, pp. 2937–2956, 2009.

[20] D.BonehandM.K.Franklin,"Identity-BasedEncryptionfromthe Weil Pairing," in Proceedings of Advances in Cryptology - CRYPTO '01, ser. LNCS, vol. 2139. Springer, 2001, pp. 213–229.

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

[22] 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.

[23] F. Guo, Y. Mu, and Z. Chen, "Identity-Based Encryption: How to Decrypt Multiple Ciphertexts Using a Single Decryption Key," in Proceedings of Pairing-Based Cryptography (Pairing '07), ser. LNCS, vol. 4575. Springer, 2007, pp. 392–406.

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

[25] T. Okamoto and K. Takashima, "Achieving Short Ciphertexts or Short Secret-Keys for Adaptively Secure General Inner-Product Encryption," in Cryptology and Network Security (CANS '11), 2011, pp. 138–159.