The security in Online Data Sharing on the Public server using Secure Key-Aggregate Cryptosystems with Broadcast Aggregate Keys

R.Mahesh Muthulakshmi [1], R.Kanimozhi[2], M.Keerthana[3], V.Sneha[4]

AP/CSE IV CSE STUDENTS

Indira Gandhi College Of Engineering & Technology For Women

ABSTRACT

Online data sharing for increased productivity and efficiency is one of the primary requirements today for any organization. The advent of cloud computing has pushed the limits of sharing across geographical boundaries, and has enabled a multitude of users to contribute and collaborate on shared data. However, protecting online data is critical to the success of the cloud, which leads to the requirement of efficient and secure cryptographic schemes for the same. Data owners would ideally want to store their data/files online in an encrypted manner, and delegate decryption rights for some of these to users, while retaining the power to revoke access at any point of time. An efficient solution in this regard would be one that allows users to decrypt multiple classes of data using a single key of constant size that can be efficiently broadcast to multiple users. Chu et al. proposed a key aggregate cryptosystem (KAC) in 2014 to address this problem, albeit without formal proofs of security. In this paper, we propose CPA and CCA secure KAC constructions that are efficiently implementable using elliptic curves and are suitable for implementation on cloud based data sharing environments.

Keyword—Cloud computing, data sharing, data security, key-aggregate cryptosystem, provable security, scalability, broadcast encryption, semantic security, CCA security

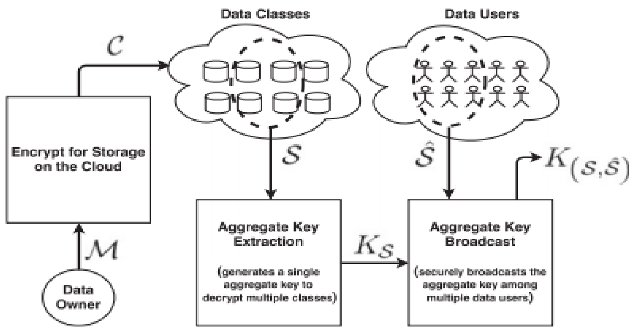
I. INTRODUCTION

The recent advent of cloud computing has pushed the limits of data sharing capabilities for numerous applications that transcend geographical boundaries and involve millions of users. Governments and corporations today treat data sharing as a vital tool for enhanced productivity. Cloud computing has revolutionized education, healthcare and social networking. Perhaps the most exciting use case for cloud computing is its ability to allow multiple users across the globe share and exchange data, while saving the pangs of manual data exchanges, and avoiding the creation of redundant or out-of-date documents. Social networking sites have used the cloud to create a more connected world where people can share a variety of data including text and multimedia. Collaborative tools commonly supported by cloud platforms and are extremely popular since they lead to improved productivity and synchronization of effort. The impact of cloud computing has also pervaded the sphere of healthcare, with Smartphone applications that allow remote monitoring and even diagnosis of patients. In short, cloud computing is changing various aspects of our lives in unprecedented ways.

Data Confidentiality: Unauthorized users (including the cloud service provider), should not be able to access the data at any given time. Data should remain confidential in transit, at rest and on backup media. User revocation: The data owner must be able to revoke any user’s access rights to data the without affecting other authorized users in the group. Scalability and Efficiency: Perhaps the biggest challenge faced by data management on the cloud is maintaining scalability and efficiency in the face of immensely large user bases and dynamically changing data usage patterns. Collusion between entities: Any data sharing service in the cloud must ensure that even when certain malicious entities collude, they should still not be able to access any of the data in an unauthorized fashion.

A traditional way of ensuring data privacy is to depend on the server to enforce access control mechanisms . This methodology is prone to privilege escalation attacks in shared data environments such as the cloud, where data corresponding to multiple users could reside on the same server. Current technology for secure online data sharing comes in two major flavors-trusting a third party auditor, or using the user’s own key to encrypt her data while preserving anonymity . In either case, a user would want a reliable and efficient cryptographic scheme in place, with formal guarantees of security, high scalability and ease of use. The main challenge in designing such a cryptosystem lies in effective sharing of encrypted data. A data sharing scheme on the cloud is only successful if data owners can delegate the access rights to their data efficiently to multiple users, who can then access the data directly from the cloud servers. Fig. 1 describes a realistic online data sharing setup on the cloud. Assume that a data owner Alice is using an online data sharing service such as Microsoft One Drive to store certain classes of data (here class may refer to any data structure such as a file, folder or any collection of these). She wishes to add an additional layer of security for her data by storing them in an encrypted fashion. Now, she intends to share a specific subset S of these documents with a set S^ of data users. For this, she needs to provide each of these users with decryption rights to specific classes of the data that they are authorized to access. The challenge therefore is to design a secure and efficient online partial data sharing scheme that allows Alice to perform this task in an efficient and secure manner.

II. ONLINE DATA SHARING SCHEME


Data Classes
Data Users
$\mathcal{C}$
Encrypt for Storage on the Cloud
$\mathcal{S}$
$\hat{\mathcal{S}}$
$K_{(\mathcal{S},\hat{\mathcal{S}})}$
$\mathcal{M}$
Data Owner
Aggregate Key Extraction
(generates a single aggregate key to decrypt multiple classes)
$K_{\mathcal{S}}$
Aggregate Key Broadcast
(securely broadcasts the aggregate key among multiple data users)

The most efficient proposition pertaining to our problem statement, to the best of our knowledge, is made in . The proposition is to allow Alice to combine the decryption power of multiple data classes into a single key of constant size. Thus, while each class of data is encrypted using a different public key, a single decryption key of constant size is suffi- cient to decrypt any subset of these classes. This system is popularly known as the key-aggregate cryptosystem (KAC), and derives its roots from the seminal work on broadcast encryption by Boneh et al.AC may essentially be considered as a dual notion of broadcast encryption . In broadcast encryption, a single ciphertext is broadcast among multiple users, each of whom may decrypt the same using their own individual private keys. In KAC, a single aggregate key is distributed among multiple users and may be used to decrypt cipher texts encrypted with respect to different classes. For broadcast encryption, the focus is on having shorter cipher texts and low overhead individual decryption keys, while in KAC, the focus is in having short cipher texts and low overhead aggregate keys. However, KAC as proposed in suffers from two major drawbacks, each of which we address in this paper.

1) First, no concrete proofs of cryptographic security for KAC are provided by the authors of . We note here that there exist significant differences in the fundamental constructs for broadcast encryption and key aggregate encryption. Broadcast encryption essentially involves two classes of parties-the broadcaster who broadcasts the secret key, and the data users who decrypt the broadcast message. On the other hand, KAC involves three parties-the data owner who encrypts and puts the data in the online sharing environment, the data users who access the data by decrypting it, and the trusted third party that generates the aggregate key. Thus a security framework for the security of KAC must be defined in order to establish the specific adversarial models against which KAC is secure.

2) Second, the scheme proposed in does not explicitly address the issue of aggregate key distribution among multiple users. In a practical data sharing environment with millions of users, it is neither practical nor efficient to depend on the existence of dedicated one to-one secure channels for key distribution. A public key based solution for broadcasting the aggregate key among an arbitrarily large number of users is hence desirable.

III. COMPACT KEY IDENTITY-BASED ENCRYPTION (IBE)

Identity-Based Encryption is a public key-based encryption scheme in which the public key for any user is an identity string corresponding to that user. Proposed initially in, IBE was concretized by the proposition of two very widely cited and popular IBEs-The Boneh-Franklin scheme and Cocks' encryption scheme. An IBE system comprises of a trusted private key generator that holds a master-secret key and issues a secret key to each user based on the user identity. Each user receives a message that has been encrypted using her id and some public parameters, and can decrypt the same using the secret key allotted to her by the trusted party. Compact key IBEs have been proposed in. The former approach involves the use of random oracles while the latter shuns the use of oracles. Both these schemes allow aggregation of keys; however each key must come from a different identity division. Fuzzy IBE allows for a single compact key to decrypt multiple ciphertexts, but they must have been encrypted under a closed set of identities, and the scheme does not work in practical scenarios for arbitrary identities.

IV.KEY-AGGREGATE CRYPTOSYSTEM: THE BASIC FRAMEWORK

The basic KAC framework presented here is the same as that in and is presented for completeness. KAC is an ensemble of five randomized polynomial-time algorithms. The system administrator is responsible for setting up the public parameters via the SetUp operation. A data owner willing to share her data using this system registers to receive her own public and private key pairs, generated using the KeyGen operation. The data owner is responsible for classifying each of her data files/messages into a specific class i. Each message is accordingly encrypted by an Encrypt operation and stored online in the cloud. When delegating the decryption rights to a specific subset of message classes, the data owner uses the Extract operation to generate a constant-size aggregate decryption key unique to that subset. Finally, an authorized data user can use this aggregate key to decrypt any message belonging to any class i 2 S. We now describe each of the five algorithms involved in KAC:

1) SetUpð1; mÞ: Takes as input the number of data classes n and the security parameter . Outputs the public parameter param.

2) KeyGen(): Outputs the public key PK and the mastersecret key msk for a data owner registering in the system.

3) Encryptðparam; PK; i;MÞ: Takes as input the public key parameter PK, the data class i and the plaintext data M. Outputs the corresponding ciphertext C.

4) Extractðparam; msk; SÞ: Takes as input the master secret key and a subset of data classes Sf1; 2; ... ; ng. Computes the aggregate key KS for all encrypted messages belonging to this subset of classes.

V.SIGNATURE SCHEMES

We briefly recall the standard definition of a signature scheme in the cryptographic literature. A signature scheme consists of the following polynomial-time algorithms:

(1) SigKeyGenð1Þ: Takes as input the security parameter and outputs a key pair ð Þ KSIG; VSIG , where KSIG and VSIG are the private signing key and public verification key respectively.

(2) SignðKSIG; MÞ: Takes as input the signing key KSIG and a message M. Outputs the corresponding signature-message pair ðSig; MÞ.

(3) VerifyðVSIG;ðSig; MÞÞ: Takes as input the verification key VSIG and a signature-message pair ðSig; MÞ. Outputs 1 if Sig is a valid signature for M under the signing key KSIG and 0 otherwise.

VI.CONCLUSION

In this paper, we have proposed an efficiently implementable version of the basic key-aggregate cryptosystem in [6] with low overhead ciphertexts and aggregate keys, using asymmetric bilinear pairings. Our construction serves as an efficient solution for several data sharing applications on the cloud, including collaborative data sharing, product license distribution and medical data sharing. We have proved our construction to be fully collusion resistant and semantically secure against a non-adaptive adversary under appropriate security assumptions. We have then demonstrated how this construction may be modified to achieve CCA-secure construction, which is, to the best of our knowledge, the first CCA secure KAC construction in the cryptographic literature. We have further demonstrated how the basic KAC framework may be efficiently extended and generalized for securely broadcasting the aggregate key among multiple data users in a real-life data sharing environment. This provides a crucial pathway in designing a scalable fully publickey based online data sharing scheme for large-scale deployment on the cloud

REFERENCES

[1] F. Gens, “IT cloud services user survey, pt. 3: What users want from cloud services providers,” Aug. 2008, http://blogs.idc. com/ie/?p=213

[2] S. S. Chow, Y.-J. He, L. C. Hui, and S. M. Yiu, “Spice-simple privacy-preserving identity-management for cloud environment,” in Applied Cryptography and Network Security. Berlin, Germany: Springer, 2012, pp. 526–543.

[3] C. Wang, S. S.-M. Chow, Q. Wang, K. Ren, and W. Lou, “Privacypreserving public auditing for secure cloud storage,” Cryptology ePrint Archive, Report 2009/579, 2009. [Online]. Available: http:// eprint.iacr.org/

[4] S. S. 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. Berlin, Germany: Springer, 2012, pp. 442–464.

[5] E. C. Shallman, “Up in the air: Clarifying cloud storage protections,” Intell. Property Law Bulletin, vol. 19, 2014, Art. no. 49.

[6] C.-K. Chu, S. S. Chow, W.-G. Tzeng, J. Zhou, and R. H. Deng, “Key-aggregate cryptosystem for scalable data sharing in cloud storage,” IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 2, pp. 468– 477, Feb. 2014.

[7] D. Boneh, C. Gentry, and B. Waters, “Collusion resistant broadcast encryption with short ciphertexts and private keys,” in Advances in Cryptology. Berlin, Germany: Springer, 2005, pp. 258–275.

[8] S. G. Akl and P. D. Taylor, “Cryptographic solution to a problem of access control in a hierarchy,” ACM Trans. Comput. Syst., vol. 1, no. 3, pp. 239–248, 1983.