Volume 5, Issue 2, February 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Key Aggregate Cryptosystem Data Sharing in Cloud Storage

Chitraranjan Ngangbamcha*
M.Tech. CSE NMAMIT,
Nitte, Karnataka, India

Savitha Shetty
Asst. Prof. Dept. of CSE, NMAMIT,
Nitte, Karnataka, India

***Abstract*— To securely, efficiently, and flexibly share data with others in cloud storage, we describe new public-key cryptosystems which produce constant-size cipher texts such that efficient delegation of decryption rights for any set of cipher texts are possible. The novelty is that one can aggregate any set of secret keys and make them as compact as a single key. The secret key holder can release a constant-size aggregate key for flexible choices of cipher text set in cloud storage, but the other encrypted files outside the set remain confidential.**

***Keywords*—ABE Schemes, Decrypt, Encrypt, Data Sharing, Key Aggregate Cryptosystem, Cloud storage,**

I. INTRODUCTION

Due to gaining popularity in cloud storages, Data Sharing has become important functionality in cloud storage. Also the question arise how efficiently, secure Data sharing is done in cloud storages [1]. The security provides by cloud server can relied by people using cloud storage, which arise a major concerned for the people using cloud storage and how efficiently Data Sharing is done through to other user in cloud[2].

This major concerned is overcome by encryption their data with their own keys before uploading them to the cloud server. This cryptographic solution of Data Sharing is done using Key aggregate cryptosystem. The Key Aggregation Cryptosystem property is especially useful to share data efficient and flexible. The schemes enable a content provider to share her data in a confidential and selective way, with a fixed and small ciphertext expansion, by distributing to each authorized user a single and small aggregate key [3].

A. Cloud Storage

Cloud storage is a model of data storage where the digital data is stored in logical pools, the physical storage spans multiple servers, and the physical environment is typically owned and managed by a hosting company. These cloud storage providers are responsible for keeping the data available and accessible, and the physical environment protected and running. People and organizations buy or lease storage capacity from the providers to store user, organization, or application data.

Cloud storage is [4] made up of many distributed resources, but still acts as one - often referred to as federated storage clouds. It is highly fault tolerant through redundancy and distribution of data, highly durable through the creation of versioned copies and typically consistent with regard to data replicas.

B. Cloud key characteristic

The following points are few key characteristic of cloud:

1. *Network access:* Cloud services are accessible over the network via standardized interfaces which enables users to access the services not only by complex devices such as personal computers, but also by light weight devices such as smart phones. in addition, the lowered cost of high-bandwidth network communication to the cloud provides access to a larger pool of it resources that sustain a high level of utilization
2. *On-Demand Self-Service:* Cloud customer can make use of cloud resources without any human interaction between them and the cloud service provider (CSP).In addition; they can schedule, manage and deploy any of cloud services such as computation and storage when needed. This leads to reduction in the personnel overhead of the cloud provider, cut in costs of the offered services.
3. *Rapid Elasticity:* It is the ability of the cloud to allocate and release resources quickly and efficiently in order to meet the requirements of the self-service characteristic of cloud computing. This automated process decreases the procurement time for new computing capabilities when the need is there, while preventing an abundance of unused computing power when the need has subsided
4. *Measured Service:* Cloud computing can dynamically and automatically measure the used resources by cloud customers. These measurements can be used to bill the customer and provide them with a payment model based on pay on use.

C. Cloud Services Model

One of the main principles of Cloud Computing is the Service paradigm in which some services are offered by a Cloud Service Provider (CSP) to customers for use. These offered services are often categorized using the Cloud Service

Model. This model represents the different layers/levels of service that can be offered to users by cloud service providers over the different application domains and types of cloud available. Clouds can be used to provide as-a-Service software to use, a platform to develop on, or an infrastructure to utilize summarizes the cloud Service Model.

D. Key Aggregate Cryptosystem

In key-aggregate cryptosystem (KAC), users encrypt a message not only under a public-key, but also under an identifier of ciphertext called class. That means the ciphertext are further categorized into different classes. The key owner holds a master-secret called master-secret key, which can be used to extract secret keys for different classes. More importantly, the extracted key have can be an aggregate key which is as compact as a secret key for a single class, but aggregates the power of many such keys, i.e., the decryption power for any subset of ciphertext classes.

A key-aggregate encryption system basically includes five algorithmic steps as Setup, KeyGen, Extract, Encrypt, Decrypt, in which The data owner establishes the public system parameter by using Setup and generates a public/master-secret key pair by using KeyGen. Messages can be encrypted using Encrypt by anyone who also decide s 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 ciphertext classes by Extract. The generated keys can be passed to Receivers securely via secure e-mails. Finally, any user with an aggregate key can decrypt any ciphertext provided that the ciphertext's class is contained in the aggregate key via Decrypt.

II. EXISTING SYSTEM

There exist several expressive ABE schemes [5] [6] where the decryption algorithm only requires a constant number of pairing computations. Recently, Green *et al.* proposed a remedy to this problem by introducing the notion of ABE with outsourced decryption, which largely eliminates the decryption overhead for users. Based on the existing ABE schemes, Green *et al.* also presented concrete ABE schemes with outsourced decryption.

In these existing schemes, a user provides an untrusted server, say a proxy operated by a cloud service provider, with a transformation key TK that allows the latter to translate any ABE cipher text CT satisfied by that user's attributes or access policy into a simple cipher text CT', and it only incurs a small overhead for the user to recover the plaintext from the transformed cipher text CT'. The security property of the ABE scheme with outsourced decryption guarantees that an adversary (including the malicious cloud server) be not able to learn anything about the encrypted message; however, the scheme provides no guarantee on the correctness of the transformation done by the cloud server. In the cloud computing setting, cloud service providers may have strong financial incentives to return incorrect answers, if such answers require less work and are unlikely to be detected by users.

III. PROPOSED SYSTEM

We considered the verifiability of the cloud's transformation and provided a method to check the correctness of the transformation. However, we did not formally define verifiability. But it is not feasible to construct ABE schemes with verifiable outsourced decryption following the model defined in the existing. Moreover, the method proposed in existing relies on random oracles (RO). Unfortunately, the RO model [7] is heuristic, and a proof of security in the RO model does not directly imply anything about the security of an ABE scheme in the real world. It is well known that there exist cryptographic schemes which are secure in the RO model but are inherently insecure when the RO is instantiated with any real hash function.

In this thesis work, firstly modify the original model of ABE with outsourced decryption in the existing to allow for verifiability of the transformations. After describing the formal definition of verifiability, we propose a new ABE model and based on this new model construct a concrete ABE scheme with verifiable outsourced decryption. Our scheme does not rely on random oracles.

In this paper we only focus on CP-ABE with verifiable outsourced decryption. The same approach applies to KP-ABE with verifiable outsourced decryption. To assess the performance of our ABE scheme with verifiable outsourced decryption, we implement the CP-ABE scheme with verifiable outsourced decryption and conduct experiments on both an ARM-based mobile device and an Intel-core personal computer to model a mobile user and a proxy, respectively.

IV. PROBLEM STATEMENT

One of the main efficiency drawbacks of the most existing ABE schemes is that decryption is expensive for resource-limited devices due to pairing operations, and the number of pairing operations required to decrypt a cipher text grows with the complexity of the access policy. The above observation motivates us to study ABE with verifiable outsourced decryption in this thesis work. Here emphasized that an ABE scheme with secure outsourced decryption does not necessarily guarantee verifiability correctness of the transformation done by the cloud server.

To design an efficient public-key encryption scheme which supports flexible delegation in the sense that any subset of the ciphertext produced by the encryption scheme, is decryptable by a constant-size decryption key generated by the owner of the master-secret key. It make a decryption key more powerful in the sense that it allows decryption of multiple ciphertexts, without increasing its size. The Key Aggregate Cryptosystem

A. Framework

The Framework of the proposed schemes consisted of following phase which are explain below which uses data sharing in cloud storage using Key Aggregate Cryptosystem:

1. *Setup (1λ, n):* Data owner executes Setup to create an account on an untrusted server. With input as security level parameter 1 λ and the number of ciphertext classes n, it outputs the public system parameter param.
2. *KeyGen:* Data owner executes KeyGen to randomly generate a public/master-secret key pair (pk ,msk)
3. *Encrypt(pk, i, m*): Anyone can execute this step who wants to encrypt data with input a public-key pk, an index i denoting the ciphertext class, and a message m, which outputs a ciphertext C.
4. *Extract (msk, S)*: Executed by the data owner to handover the decrypting power for a certain set of ciphertext classes to a Receiver. On input the master-secret key msk and a set S of indices corresponding to different classes, it outputs the aggregate key for set S denoted by Ks
5. *Decrypt (Ks, S, i, C)*: Execute d by a Receiver who received an aggregate key Ks generated by Extract. On input Ks, the set S, an index i denoting the ciphertext class the ciphertext C belongs to, and C, it outputs the decrypted result m if i ϵ S.

The main idea of data sharing in cloud storage using KAC, illustrated in Fig. 1. Suppose user wants to share her data (m1; m2;:: : ;m) on the server. First it will performs Setup to get param and execute KeyGen to get the public/master-secret key pair (pk; msk). The system parameter param and public-key pk can be made public and master-secret key msk should be kept secret by User. Anyone (including User) can then encrypt each mi by Ci = Encrypt (pk; i; mi).

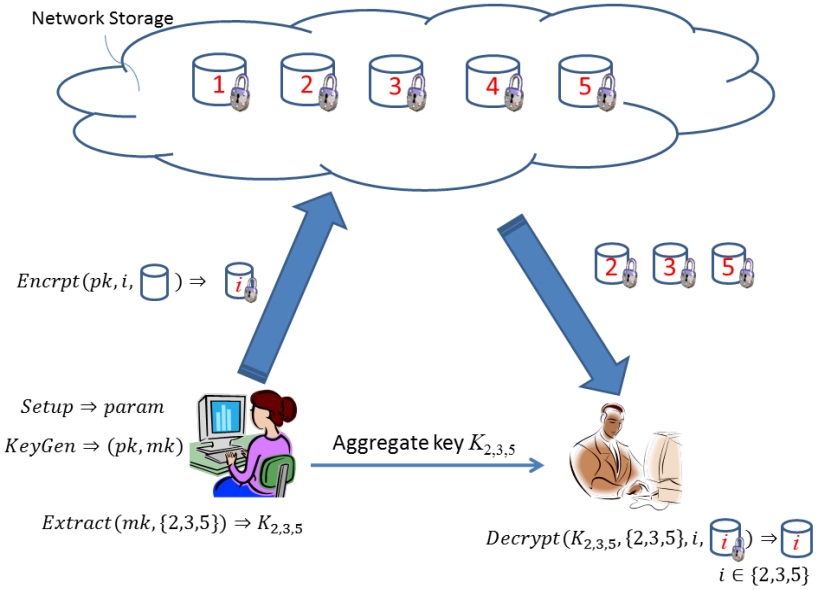

Fig. 1. Using KAC for data sharing in cloud storage

The Encrypted data are uploaded to the Cloud Storages. With param and pk, people who cooperate with User can update User's data on the server. Once User is willing to share a set S of her data with a friend Receiver, It can compute the aggregate key KS for Receiver by performing Extract (msk; S). Since KS is just a constant size key, it is easy to be sent to Receiver via a secure email. After obtaining the aggregate key Receiver, can download the data he is authorized to access with aggregate keys, Reciever can decrypts the share date.

V. CONCLUSION

Our system is more secure and flexible in providing data privacy in cloud storages. Its compress secret keys in public-key cryptosystems which support delegation of secret keys for different cipher text classes in cloud storage which always get an aggregate key of constant size. In cloud storage, the number of cipher texts usually grows rapidly, so we have to reserve enough cipher text classes for the future extension. Otherwise, we need to expand the public-key. Although the parameter can be downloaded with cipher texts, it would be better if its size is independent of the maximum number of cipher text classes.

ACKNOWLEDGMENT

I would like to extend my gratitude to many people who helped me to bring this paper fruition. First I would like to thank Miss Savitha Shetty, Asst. Prof., NMAMIT, Nitte. I am so deeply grateful for her help, professionalism, and valuable guidance throughout this paper. I would also like to thank to my friends and colleague. This accomplishment would not have been possible without them. Thank you.

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] 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, 2013.

[3] D. Boneh, C. Gentry, and B. Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys," in Proceedings of Advances in Cryptology - CRYPTO '05, ser. LNCS, vol. 3621. Springer, 2005, pp. 258–275.

[4] S. Rhea, C. Wells, P. Eaton, D. Geels, B. Zhao, H. Weatherspoon, and J. Kubiatowicz, Maintenance-Free Global Data Storage. IEEE Internet Computing, Vol 5, No 5, September/October 2001, pp. 40–49.

[5] 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.

[6] 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.

[7] C.-K. Chu and W.-G. Tzeng, "Identity-Based Proxy Re-encryption without Random Oracles," in Information Security Conference (ISC' 07), ser. LNCS, vol. 4779. Springer, 2007, pp. 189–202.

