Cryptosystem Based On Aggregate Key for Efficient Data Sharing In Cloud Storage

Rohan Kshirsagar, Mayur Shinde, Vishal Gaikwad, Amol Alhat

Abstract—**Data sharing is an important term in cloud storage. In this paper, we show how to securely, efficiently and flexibly share data with other user in cloud. We describe new public-key cryptosystem which generate constant-size ciphertexts such that well organized delegations of decryption authority for any set of ciphertexts are possible. The creativity is that, one can aggregate any bunch of secret keys and make them close-packed as a single key, but encloses the power of all the keys get aggregated. In another words, the secret key holder can release a constant-size aggregate key for flexible choices of ciphertexts set in cloud storage, but the other encrypted files outside the set remain confidential. This close-packed aggregate key can be conveniently sent to others or be stored in a smart card with very small secure storage.**

We provide security of our schemes in the standard model. We also describe other application of our schemes. Our scheme gives the first public-key encryption for flexible hierarchy, which was yet to be known.

***Index Terms* — Cloud storage, data sharing, key-aggregate encryption, patient-controlled encryption**

I. INTRODUCTION

Cloud storage is increasing its popularity day by day. We see the rise in demand for data outsourcing, which assists in the strategic management of corporate-data. Core technology is used behind many online services for personal application. It is easy to apply for free accounts for email, file sharing or remote access, with minimum storage size. Together with the current wireless technology, user can access all of their files and email by a mobile phone in any corner of the world. Considering data privacy, traditional way to ensure it is to rely on server to enforce the access control after authentication that means any unexpected privilege hike will expose all data. In a shared-tenancy cloud computing environment, things can become even worse. Data from various clients can be hosted on different virtual machines but resides on a single physical machine.

Manuscript received Jan, 2016.

***Rohan Kshirsagar**,Computer Engineering, Pune University/ PGMCOE .Pune ,India, 9762684100.*

***Mayur Shinde**, Computer Engineering, Pune University/ PGMCOE Pune , India,7387387878.*

***Vishal Gaikwad**, Computer Engineering, Pune University/ PGMCOE Pune ,India .9158817373.*

***Amol Alhat**, Computer Engineering, Pune University/ PGMCOE Pune ,India .9762527755.*

Data in targeted Virtual Machine could be stolen by instantiating another VM's co-resident with the target with the data owner's anonymity similarly, cloud users probably will not grip the strong trust that the cloud server is doing a good job assuming confidentiality. A cryptographic solution, with proven security which rely on number of theoretic assumptions is more advisable, whenever user is not happy with believing the security of the virtual machine's. These users are inspired to encrypt their data with their own keys before uploading them to the server.

Data sharing is very important functionality in cloud storage. For example, bloggers can allow their friends to see a subset of their personal pictures; an enterprise may grant its employees to access a portion of a sensitive data. The challenging problem is how we effectively share an encrypted data. Another way, users can download an encrypted data from the known storage, then decrypt them and then send them to others for sharing, because of this value of cloud storage will be lost.

Users should be able to assign the access rights of the sharing data to others so that they can access these data from the server directly. Finding an efficient and secure way to share partial data is not trivial in cloud storage. Assuming that Alice puts all her personal photos on Drop box, and she does not want to show her photos to everyone. Due to many data leakage possibility he/she cannot feel relieved by just depending on the privacy protection mechanisms given by Drop box, so he/she encrypts all the photo's using own keys before uploading.

Regarding the availability of files, there are a series of cryptographic scheme which can go as far as permitting a third-party auditor to examine the availability of files for the data owner without leaking any data.

II. LITERATURE SURVEY

Existing System

A. Predefined Hierarchy using Cryptographic Keys

Cryptographic key assignment main aim is to minimize the expense in storing and managing secret keys for cryptographic use. Utilizing a tree structure, the keys of its descendant nodes can be derived using a key for a given branch.

Advanced cryptographic key assignment scheme(e.g., [1], [2], [3], [4]) support access policy that can be modeled by cyclic graph or acyclic graph.

ISSN: 2278 – 1323

B. Symmetric-Key Encryption using Compact Key

An encryption scheme is originally proposed for concisely transmitting large number of keys in broadcast scenario [5]. Finally, we see that there are many types which try to minimize key size for getting authentication in symmetric-key encryption, e.g., [6]. Hence sharing of decryption power is not a problem in these schemes.

C. Identity-Based Encryption (IBE) using Compact Key

IBE is one of the types of public-key encryption where the public-key of a user can be used as an identity string of the user. (e.g., [7], [8], [9])There is a trusted party known as private key generator in IBE which holds master-secret key and issues a secret key to each and every user with respect to the user identity. The encryption can take the public parameter and an user identity to encrypt a message. Recipient can decrypt this cipher-text by his secret key.

D. Attribute-based encryption (ABE)

ABE permits each cipher text to be associated with an attribute, [10], [11]and master secret key holder extract a secret key for a policy of these attributes so that cipher text can be decrypted by this key its associated attribute observe to the policy.

E. Primitive is proxy re-encryption (PRE)

To delegate decryption power of ciphertexts without sending the secret key to the delegate, a useful primitive is proxy re-encryption (e.g [11], [12]). It allows sender to delegate to the server the ability to convert ciphertext encrypted under the public-key into ones for receiver.

Disadvantages

1) Predefined Hierarchy using a Cryptographic generally is more costly than symmetric-key operation.

2) IBE increases the costs of storing and transmitting cipher-texts, which is impractical in various situations such as shared cloud storage.

3) Under ABE, the size of the key often grows linearly with the number of attributes it encompasses.

4) Primitive is proxy re-encryption just moves the secure key storage requirement from the delegates to the proxy. It is thus undesirable to allow the proxy reside in the storage server oversteps. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity in an equation.

III. PROPOSED SYSTEM

To study a way to build decryptions key more powerful in the sense that it allows decryption of multiple ciphertexts, while not increasing its size. To design associate economical public key encryption scheme which support a flexible delegation in the sense that any subset of the cipher texts is decryptable by an constant size of decryption key. It introduces a special type of public-key encryption which is called, key-aggregate cryptosystem (KAC). In Key-aggregate cryptosystem, user encrypts a message not only under a public-key, but also under an identifier of ciphertext called class. That means the ciphertexts are further differentiated into completely different classes. The key owner holds a master-secret called master-secret key, which may be used to extract secret keys for various categories.

More importantly, the extracted key 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 that is the decryption power for any subset of ciphertext classes. Aggregate key can be sent to receiver or user through a securely registered e-mail ID. Receiver or user can download the encrypted content and decrypt these data using aggregate key.

A. *Key-Aggregation Cryptosystem*

Key-aggregate encryption plan consists of five polynomial time algorithms as follows:

Here data owner establishes the public system parameter with the help of Setup and generates a public/master secret key pair through Key-Gen. With the help of Encrypt, messages can be encrypted by anyone who also decides what cipher-text class is associated with the plain text message that has to be encrypted. The data owner can use the master-secret to generate an aggregate decryption key for a set of ciphertext classes via Extract. These generated keys can be passed to delegates securely by secured e-mails or any secured devices. Finally, any user with an aggregate key can decrypt any ciphertext provided that the ciphertext's class is contained in the aggregate key through Decrypt.

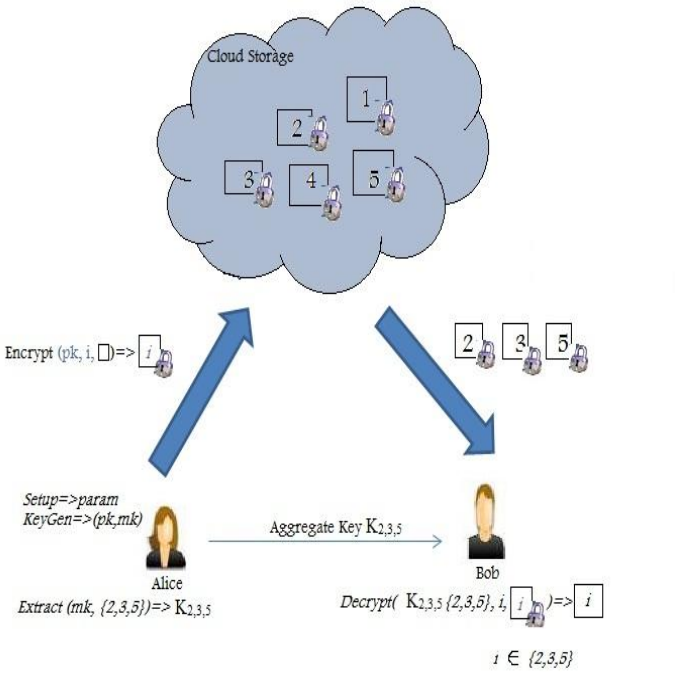

Fig. 1 Using KAC for data sharing in cloud storage

Setup: The algorithm which is to be setup takes no input other than the implicit security parameters. It gives output as a master key *mk* and public parameter *pk*.

KeyGen Phase: To generate the public key or the master key pair (*pk*, *mk*) this phase has to be executed by data owner.

ISSN: 2278 – 1323

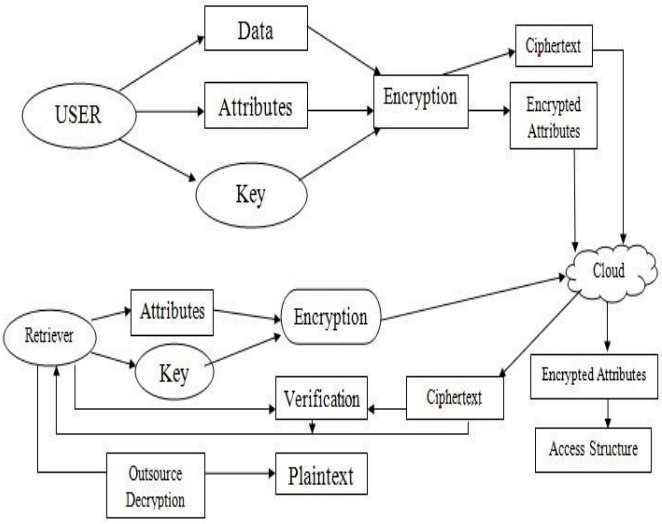

Fig. 2. Architectural Diagram For KAC

Encrypt Phase: This phase can be executed by anyone who wants to send the encrypted data. Encrypt (pk, m, i), the encryption algorithm takes input as a message m, public parameter pk, and i denoting ciphertext class. The algorithm encrypts messages m and produce a ciphertext C such that the only a user that has a set of attributes that satisfies the access structure is able to decrypt the messages.

Input= public key pk, an index i, and message m
Output = ciphertext C.

Extract: For delegation the decrypting power for certain set of ciphertext classes to a delegate they are executed by the data owner. On input a set S of indices and the master-secret key msk corresponding to different classes, it give output as the aggregate key for set S denoted by K_s.

Decrypt Phase: It is executed by the candidate who has decryption authority. The decryption algorithm takes input as a ciphertext C_a public parameters pk, i denoting cipher text classes for a set S of attribute.

Input = K_s and the set S, where index i = ciphertext class
Outputs = m if *i* element of S.

B. *Our Contribution*

In latest cryptography, a fundamental problem we often study is about holding the secrecy of a small piece of knowledge in to the ability to perform cryptography functions (e.g. encryption, authentication) multiple times. In this paper, we will learn how to make a decryption key more powerful in the sense that it allows decryption of multiple cipher texts, without growing its size. Specifically, our problem statement is-"To design an efficient public-key encryption strategy which supports flexible delegation in the sense that any subset of the ciphertexts (produced by encryption scheme) is decryptable by an constant size of decryption key i.e, generated by the owner of the master-secret key."

We can do resolve this downside by launching a special form of public-key encryption which we call key-aggregate cryptosystem. In KAC, users can encrypt a message not only under a public-key, but also under an identifier of ciphertext called class. This means that the ciphertexts are further categorized into different classes. The key owner holds a master-secret called as master-secret key, which can be used to extract secret keys for different classes. More importantly, the extracted key can be an aggregate key which is as compact as a secret key for a single class, but aggregates the ability of the many such keys that is the decryption power for any subset of ciphertext classes.

With our solution, Alice will merely send Bob a single aggregate key via a secure e-mail. Bob can download the encrypted photos from Alice's Drop box space and then can use this aggregated key to decrypt these encrypted photos. The scenario is depicted in Figure 1.
The sizes of aggregate key, public-key, master-secret key and ciphertext in our KAC schemes are all of constant size.

IV. Conclusion

How to secure user's data privacy is a central question of cloud storage. With more mathematical tools, cryptographic schemes are getting more and more versatile and often involve multiple keys for a single application. In this article, we consider how to compress secret keys in public-key cryptosystem which support delegation of secret keys for different ciphertext classes in cloud storage. No matter which one is among the power set of classes, the delegates can always get an aggregate key of constant size. Our way of approach is more flexible than hierarchical key assignment which can only save spaces if all key holders share a similar set of privilege.

A limitation in the work is the predefined bound of the number of maximum cipher-text classes. In cloud storage, the total number of cipher text usually grows rapidly. So we have to reserve enough ciphertext classes for the extension of future. Otherwise, we need to expand the public-key.

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

REFERENCES

[1] 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.
[2] 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.
[3] W.-G. Tzeng, "A Time-Bound Cryptographic Key Assignment Scheme for Access Control in a Hierarchy," IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 14, no. 1, pp. 182–188, 2002.
[4] 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.
[5] 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.
[6] B. Alomair and R. Poovendran, "Information Theoretically Secure Encryption with Almost Free Authentication," J. UCS, vol. 15, no. 15, pp. 2937–2956, 2009

ISSN: 2278 – 1323

[7] D. Boneh and M. K. Franklin, "Identity-Based Encryption from the Weil Pairing," in Proceedings of Advances in Cryptology CRYPTO '01, ser. LNCS, vol. 2139. Springer, 2001, pp. 213–229.
[8] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," in Proceedings of Advances in Cryptology - EUROCRYPT '05, ser. LNCS, vol. 3494. Springer, 2005, pp. 457–473
[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted data," in 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
[10] Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06). ACM, 2006, pp. 89–98.
[11] R. Canetti and S. Hohenberger, "Chosen-Ciphertext Secure Proxy Re-Encryption," in Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07). ACM, 2007, pp. 185–194.
[12] 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.

Rohan Kshirsagar
Computer Engineering,
Pune University/ PGMCOE
Pune, India, Mob 9762684100

Mayur Shinde
Computer Engineering
Pune University/ PGMCOE
Pune , India,7387387878

Vishal Gaikwad
Computer Engineering
Pune University/ PGMCOE
Pune, India .Mob 9158817373

Amol Alhat
Computer Engineering
Pune University/PGMCOE
Pune ,India .Mob 9762527755

ISSN: 2278 – 1323