Secure Data Forwarding Using KAC

Neethumol S Nair

M. Tech Student, Department of Computer Science, Mount Zion College of Engineering Pathanamthitta

Abstract: ***Cloud storage is nowadays very popular storage system. Cloud storage is a storage of data online in cloud which is accessible from multiple and connected resources. Cloud storage can provide good accessibility, reliability, strong protection, disaster recovery, and lowest cost. here describes how securely ,efficiently and flexibly sharing data with others in cloud storage. In modern cryptography, a primary problem often studied is that leveraging the secrecy of a small piece of knowledge into the ability to perform the cryptographic functions (e.g., encryption, authentication) multiple times making a decryption key more powerful in the sense that it allows decryption of multiple cipher texts, without increasing its size. This problem is solved by introducing a special type of public-key encryption which is called as key-aggregate cryptosystem (KAC).KAC produces constant-size ciphertexts such that efficient delegation of decryption rights for any set of ciphertexts are possible. Any set of secret keys can be aggregated and make them as single key, which encompasses power of all the keys being aggregated. This aggregate key can be sent to the others for decryption of ciphertext set. The remaining encrypted files outside the set are remains confidential and secure. This compact aggregate key can be conveniently sent to others or to be stored in a smartcard with very limited storage. This scheme give the first public-key patient-controlled encryption for flexible hierarchy, which was yet to be known.***

Keywords: data sharing, cloud storage, public key encryption, patient-controlled encryption, key-aggregate encryption.

1. Introduction

The usage of cloud computing has increased in almost all organizations. It offers delivery of computing services and resources over the internet. Major facilities of cloud computing are data storage and data sharing. Cloud services are popular because they can reduce cost, complexity of owning and operating computers and network accessibility of data and accessibility of data. It is also used as a core technology behind many online services for personal applications. Cloud computing environment provide secure cloud storage as it contain sensitive datas of users.

Users cannot depend of earlier authentication techniques on regarding data privacy, because third party can get all the data through unprivileged access. Data sharing is another important functionality of cloud storage. Using this feature, user can share data from anywhere and anytime to anyone. While data sharing, datas from different clients can be hosted on separate virtual machines(VMs) but belongs to a single physical machine. But data in a target VM could be stolen by instantiating another VM on same physical machine. Here break the trust of data privacy. So cloud user cannot fully depend on cloud server in terms of data security and confidentiality. Solution of this problem is to encrypt the data before uploading to the cloud with users own keys. Thus users are motivated to encrypt their data with own keys thus providing access to only desired Recipients. The challenging task is how effectively share encrypted data.

Modern Cryptography techniques can be applied in a two major ways- one is symmetric key encryption and other is asymmetric key encryption. In symmetric key encryption, same keys are used for encryption and decryption. In asymmetric key encryption different keys are used, public key for encryption and private key for decryption. Asymmetric key encryption is more flexible in this approach. This can be illustrated by following example.

Suppose Alice uploads a set of photos over cloud. But she does not want to share these photos with everyone. So she needs to put some security constraints. She is not satisfied with available security measures. So she encrypts his photos using his own keys before uploading. One day Bob asks Alice to share his photos, then Alice will send him a single constant size decryption key via secure channel or secure device. In symmetric key approach, unwanted data also get expose to the Bob, which is inadequate. In asymmetric key approach, number of keys is as many as number of shared files, which may be hundred or thousand as well as transferring these keys require secure channel and storage space which can be expensive.

So best solution of this problem is provided using Key Aggregate Cryptosystem (KAC). Using this method, Alice encrypts data with distinct public keys, but send single decryption key of constant size to Bob. Since the decryption key should be sent via secure channel and kept secret small size is always needed. The sizes of ciphertext, public-key, master-secret key, and aggregate key in KAC schemes are all of constant size.

2. Key Aggregate Cryptosystem

In this article describes how to effectively share encrypted data and how to make a decryption key more powerful for the decryption of multiple ciphertexts. The aim is to Design a public key encryption scheme in such a way that any subset of the ciphertext is decryptable by a constant size decryption key.

For this, introduce a special type of public-key encryption called Key-Aggregate Cryptosystem. In KAC, users encrypt the message not only under a public-key, but also under an identifier of ciphertext called class. This means the ciphertexts are further classified into different classes. The key owner holds a master-secret called master-secret key, this master-secret key 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. The size of ciphertext, public-key, master-secret key and the aggregate key are all of constant size in the scheme. A canonical application of KAC is data sharing. The key aggregation property is especially useful when we expect delegation to be efficient and flexible. The KAC schemes enables a content provider to share the data in a confidential and selective way, with a fixed and small ciphertext expansion, by distributing to each authorized user a single and small(constant) aggregate key.

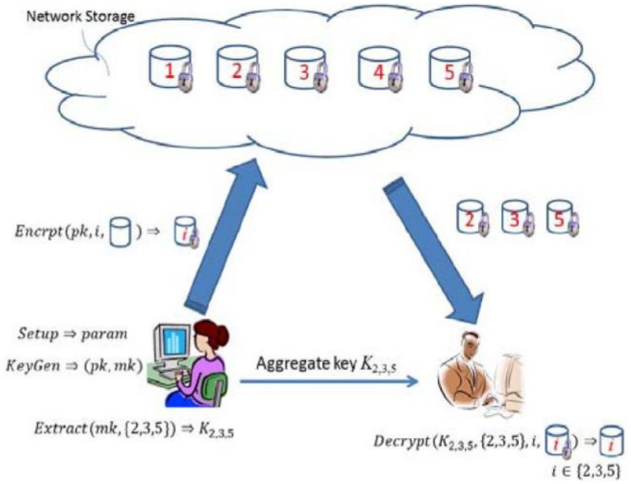

Network Storage
1
2
3
4
5
Encrpt(pk, i,) ⇒ i
2
3
5
Setup ⇒ param
KeyGen ⇒ (pk, mk)
Aggregate key $K_{2,3,5}$
Extract(mk, {2,3,5}) ⇒ $K_{2,3,5}$
Decrypt($K_{2,3,5}$, {2,3,5}, i, i) ⇒ i
i ∈ {2,3,5}

Figure 1: Using KAC for data sharing in cloud storage.

The data owner establishes the public system parameter through **Setup** and generates a public/master-secret key pair through **KeyGen**. Data 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 key pair to generate an aggregate decryption key for a set of ciphertext classes through **Extract.** The generated keys can be passed to recipients securely through secure e-mails or secure devices 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**. This scheme consists of five algorithmic steps as follows.

- Setup (1λ, n): This is executed by the data owner to Setup to create an account on an untrusted server. On input of a security level parameter 1λ and number of ciphertext classes N, it outputs the public system parameter *param.*
- KeyGen: It is executed by data owner to randomly generate a public/ master-secret key pair (Pk, MSK).
- Encrypt (Pk, I, M): It is executed by data owner and for message M and index I, it computes the ciphertext as CT.
- Extract (MSK, S): It is executed by data owner for delegating the decrypting power for a certain set of ciphertext classes and it outputs the aggregate key for set S denoted by As.
- Decrypt (As, S, I, C): It is executed by a delegate who received, an aggregate key Ks generated by Extract. On input As, set S, an index I denoting the ciphertext class ciphertext CT belongs to and output is decrypted result D.

3. Related Works

In this section, compare the KAC scheme with other possible solutions on sharing in secure cloud storage environment.

A) Cryptographic Keys for a Predefined Hierarchy

Cryptographic key assignment schemes works on the basis of minimize the expense in storing and managing secret keys for general cryptographic use by using a tree structure . By using hierarchical tree structure, a key for a given branch can be used to derive the keys of its descendant nodes .This can solve the problem partially if one intends to share all files under a certain branch in the hierarchy which alternatively means that the number of keys increases with the number of branches. So it is difficult to create a hierarchy that can save the number of total keys to be granted for all individuals simultaneously.

B) Compact Key in Symmetric-Key Encryption

This encryption scheme which is originally proposed for concisely transmitting large number of keys in broadcast scenario. The construction is simple. This 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.

C) Compact Key with Identity-Based Encryption (IBE)

Identity-based encryption (IBE) is a 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, mobile number). In this encryption, there is a trusted party called private key generator in IBE which holds a master-secret key and gives a secret key to each user with respect to the user identity. The content provider can take the public parameter and a user identity to encrypt a message. The recipient can decrypt this ciphertext by his secret key. The receiver can decrypt this ciphertext by his secret key. Some tried to build IBE with key aggregation. But their key-aggregation comes at the expense of O(n) sizes for both ciphertext and the public parameter, where n is the number of secret keys. This greatly increases the costs to store and transmit ciphertext.

D) Attribute-based encryption (ABE)

Attribute-based encryption (ABE) allows each ciphertext to be associated with an attribute, and the master-secret key holder can extract a secret key for a policy of these attributes so that a ciphertext can be decrypted by this key if its associated attribute conforms to the policy. All comparison can be summarized in following table

Table 1: Comparison between KAC scheme and other related scheme

Different schemes	Key sizes and encryption types		
	Cipher text size	*Decryption key size*	*Encryption type*
Key assignment schemes	Constant	Non-constant	Symmetric or public-key
Symmetric-key encryption	Constant	Constant	Symmetric key
IBE	Non-constant	Constant	Public key
ABE	Constant	Non-constant	Public key
KAC	Constant	Constant	Public key

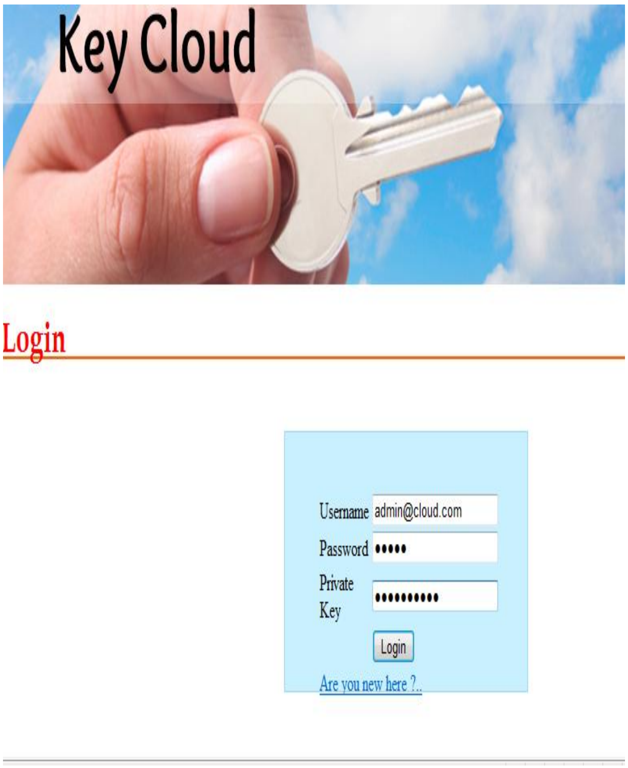
E) Patient-Controlled Encryption (PCE)

Here implemented this KAC in preserving patient's privacy in electronic health record systems .Moving to electronic health records is important to the modernization of healthcare system. But computerized medical records are vulnerable to cyber attacks. Also patient may need to share their data partially with some users. Thus designing Patient Controlled Encryption (PCE). It provides solution to secure and private storage of patients' medical records. In PCE, the health record is decomposed into a hierarchical tree structure based on the use of different ontologies, and patient is the one who generate and store secret keys. So whenever there is a need to access part of the record, a patient will release the secret key for the concerned part of the record. Thus any patient can either define his own hierarchy according to his need, or follow the set of categories suggested by the electronic medical record system, such as disease, x-rays, doctors, allergies, medications, and so on. When the patient wishes to give access rights to her doctor, he can choose any subset of these categories and provide a single key, from which keys for all these categories can be computed. Thus, this cryptosystem helps user to securely and partially share the data over cloud.

4. Result Analysis

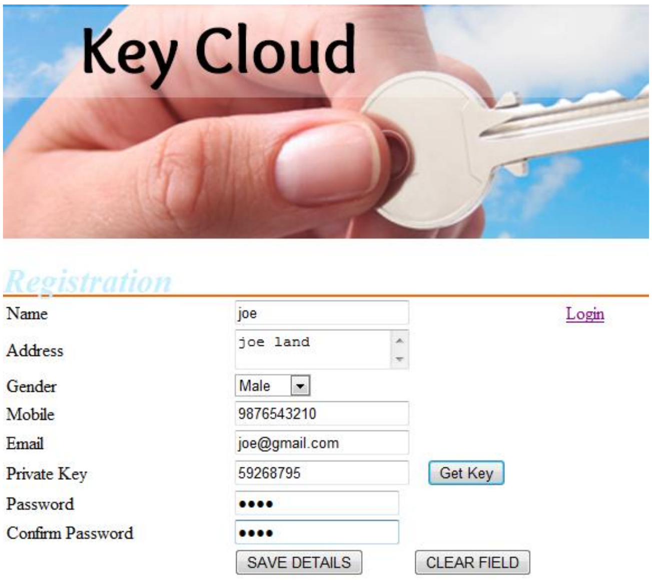
The main advantage of this paper is that it helps user can efficiently and securely transfer datas in cloud storage environment.

Following screenshots represents the output of the work carried out on the project Key Aggregate Cryptosystem.



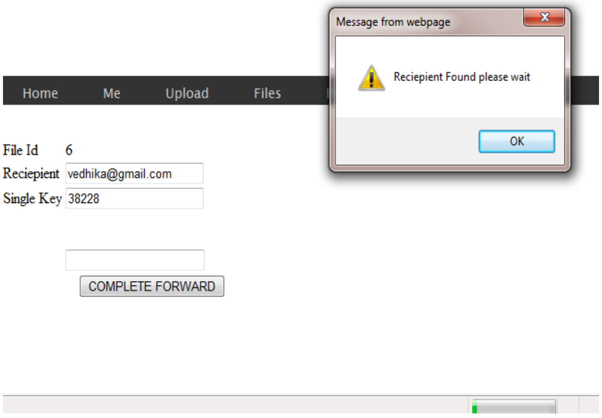
Username admin@cloud.com
Password •••••
Private Key ••••••••••

Figure 2: Login



Name joe
Address joe land
Gender Male
Mobile 9876543210
Email joe@gmail.com
Private Key 59268795
Password ••••
Confirm Password ••••

Figure 3: Registration



Reciepient Found please wait

File Id 6
Reciepient vedhika@gmail.com
Single Key 38228

Figure 4: Aggregate key generation

5. Conclusion

Here implemented this KAC in preserving patient's privacy in electronic health record systems .Moving to electronic health records is important to the modernization of healthcare system. But computerized medical records are vulnerable to cyber attacks. Also patient may need to share their data partially with some users. Thus designing Patient Controlled Encryption (PCE). It provides solution to secure and private storage of patients' medical records. In PCE, the health record is decomposed into a hierarchical tree structure based on the use of different ontologies, and patient is the one who generate and store secret keys. So whenever there is a need to access part of the record, a patient will release the secret key for the concerned part of the record. Thus any patient can either define his own hierarchy according to his need, or follow the set of categories suggested by the electronic medical record system, such as disease, x-rays, doctors, allergies, medications, and so on. When the patient wishes to give access rights to her doctor, he can choose any subset of these categories and provide a single key, from which keys for all these categories can be computed. Thus, this cryptosystem helps user to securely and partially share the data over cloud.

References

[1] 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

[2] Cheng-Kang Chu ,Chow, S.S.M, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng , ―Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage‖, IEEE Transactions on Parallel and Distributed Systems. Volume: 25, Issue: 2. Year :2014

[3] 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.

[4] 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.

[5] 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. 362375, 2013.

[6] 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.

[7] 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.

[8] J. Benaloh, ―Key Compression and Its Application to Digital Fingerprinting,‖ Microsoft Research, Tech. Rep., 2009.

Author Profile

Neethumol S Nair received the Bachelor of Technology degree in Computer Science and Engineering from Musaliar College Of Engineering and Technology Pathanamthitta in 2013. Currently doing Mtech degree in Computer Science and Engineering under Mahatma Gandhi University.