

KEY-CUMULATIVE CRYPTOSYSTEM FOR ACCESSIBLE DATA DISTRIBUTION IN CLOUD STORAGE

Arpitha M G¹, Prasanna B.T²

¹M.Tech 2nd Semester, ²Assistant Professor,

Department of Computer Science & Engineering, Sri Jayachamarajendra College of Engineering(SJCE),JSS S & T University campus Mysore, Karnataka, India

Abstract: Data sharing being important functionality in cloud storage implements how to securely, efficiently, and flexibly share data with others. The public-key cryptosystems produce constant-size cipher texts that efficiently delegates the decryption rights for any set of cipher texts. The importance is that one can aggregate any set of secret keys and make them as compact as a single key, but encompassing the power of all the keys being aggregated. 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. The aggregate key can be conveniently sent to others or be stored in a smart card with very limited secure storage. In this paper we present the work done by different authors in this field.

Keywords: Cloud storage, public key encryption, cryptosystem, key aggregate encryption, and key aggregate cryptosystem.

I. INTRODUCTION

Cloud storage is gaining popularity recently. In enterprise settings, we see the rise in demand for data outsourcing, which assists in the strategic management of corporate data. It is also used as a core technology behind many online services for personal applications. Now a days, it is easy to apply for free accounts for email, photo album, and file sharing and/or remote access, with storage size more than 25GB (or a few dollars for more than 1TB). Together with the current wireless technology ,users can access almost all of their files and emails by a mobile phone in any corner of the world. Its flexible and cost optimizing characteristic motivates the end user as well as enterprises to store the data on cloud. The insider attack is one of security concern which is needs to be focused. Cloud Service provider need to make sure whether audits are held for users who have physical access to the server. As cloud service provider stores the data of different users on same server it is possible that user's private data is leaked to others. The public auditing system of data storage security in cloud computing provides a privacy-preserving auditing protocol [2].

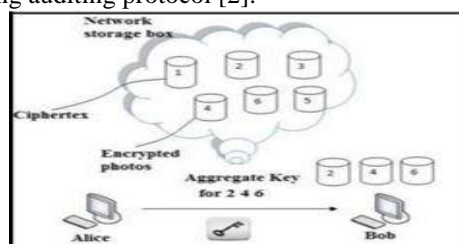


Figure 1

A new way for public-key encryption is used called as key aggregate cryptosystem (KAC)[1]. The encryption is done through an identifier of Cipher text known as class, with public key. The classes are formed by classifying the cipher text. The key owner has the master secret key which is helpful for extracting secret key. So in above scenario now the Alice can send a aggregate key to bob through a email and the encrypted data is downloaded from drop box through the aggregate key. This is shown in Figure1.

II. BACKGROUND

Cloud computing is visualized as architecture for succeeding generation. It has many facilities though have a risk of attacker who can access the data or leak the user's identity. While setting a cloud users and service providers authentication is necessary. The issue arises whether cloud service provider or user is not compromised. The data Will leak if any one of them in compromised. The cloud should be simple, preserving the privacy and also maintaining User 's identity [1].The flexible use of cloud storage for user is a need as it is seams accessing data locally though that is present at remote side. It is important to inspect the data set on the cloud. So it is necessary to allow a public audit for integrity of outsourced data through third party auditor (TPA). TPA is also beneficial for cloud service provider. It checks the correctness of the outsourced data. TPA should be able to do public auditability, storage correctness, privacy preserving, Batch auditing with minimum communication and computation overhead [2].There are many cloud users who wants to upload there data without providing much personal details to other users. The anonymity of the user is to be preserved so that not to reveal the identity of data owner. Provable data possession (PDP)uses similar demonstrating marks to reduce computation on server, and network traffic. PDA ensures the data present on cloud which is un-trusted is original without accessing it. Security mediator (SEM) is approach allows the user to preserve the anonymity. Users are meant to upload all their data to SEM so that the SEM is not able to understand the data although it's going to generate the verification on data. As the users are signed at SEM it should not know the identity of up loader [3]. Another way for sharing encrypted data is Attribute-Based Encryption (ABE). It is likely to encrypt the data with attributes which are equivalent to users attribute rather the only encrypting each part of data. In ABE attributes description is considered as set so that only a particular key which is matched with attribute can decrypt the cipher text. The user key and the attribute are matched if

it matches it can decrypt a particular cipher text. When there are attributes are overlay among the cipher text and a private key the decryption is granted [5]. A multi group key management accomplishes a hierarchical access control by applying an integrated key graph also handling the group keys for different users with multiple access authorities. Centralized key management plan uses tree structure to minimize the data processing, communication and storage overhead. It maintains things related to keying and also updates it. It accomplishes an integrated key graph for every user [6]. Identity-based encryption (IBE) is a vital primary thing of identity bases cryptography. The public key of user contains distinct information of user's identity. The key can be textual value or domain name, etc. IDE is used to deploy the public key infrastructure. The identity of the user is used as identity string for public key encryption. A trusted party called private key generator (PKG) in IBE which has the master secret key and gives secret key to users according to the user identity. The data owner collaborate the public value and the identity of user to encrypt the data. The cipher text is decrypted using secret key [7]. In a multi attribute-authorities numbers of attributes are analyzed regarding the decryption key and the user must get a particular key related to the attribute while decrypting a message. The decryption keys are allocated independently to users those who have attribute identity without interaction between each other. Multi-authority attribute-base decryption allows real time deployment of attribute based privileges as different attributes are issued by different authorities. The attribute authorities ensure the honesty of the user privilege so the confidentiality is maintained by central authority [8].

III. KEY-CUMULATIVE ENCRYPTION

A key aggregate encryption has five polynomial-time algorithms as

3.1 Setup Phase

The data owner executes the setup phase for an account on server which is not trusted. The setup algorithm only take simplicity security parameter.

3.2 KeyGen Phase

This phase is executed by data owner to generate the public or the master key pair (pk, msk).

3.3 Encrypt Phase

This phase is executed by anyone who wants to send the encrypted data. Encrypt (pk, m, i), the encryption algorithm takes input as public parameters pk, a message m, and I denoting cipher text class. The algorithm encrypts message m and produces a cipher text C such that only a user that has a set of attributes that satisfies the access structure is able to decrypt the message.

3.4 Cloud Storage

Cloud storage is nowadays very popular storage system. Cloud storage is storing of data off-site to the physical storage which is maintained by third party. Cloud storage is saving of digital data in logical pool and physical storage

spans multiple servers which are manage by third party. Third party is responsible for keeping data available and accessible and physical environment should be protected and running at all time. Instead of storing data to the hard drive or any other local storage, we save data to remote storage which is accessible from anywhere and anytime. It reduces efforts of carrying physical storage to everywhere. By using cloud storage we can access information from any computer through internet which omitted limitation of accessing information from same computer where it is stored. While considering data privacy, we cannot rely on traditional technique of authentication, because unexpected privilege escalation will expose all data. Solution is to encrypt data before uploading to the server with user's own key. Data sharing is again important functionality of cloud storage, because user can share data from anywhere and anytime to anyone. For example, organization may grant permission to access part of sensitive data to their employees. But challenging task is that how to share encrypted data. Traditional way is user can download the encrypted data from storage, decrypt that data and send it to share with others, but it loses the importance of cloud storage.

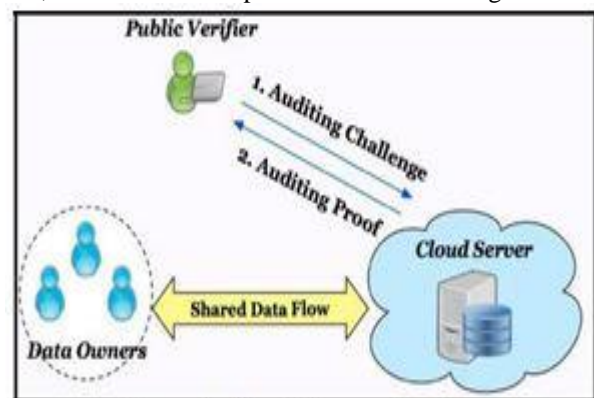


Figure 2: Cloud Storage Structure

Cryptography technique 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. By contrast, in asymmetric key encryption different keys are used, public key for encryption and private key for decryption. Using asymmetric key encryption is more flexible for our approach. This can be illustrated by following example. Suppose Alice put all data on Box.com and she does not want to expose her data to everyone. Due to data leakage possibilities she does not trust on privacy mechanism provided by Box.com, so she encrypt all data before uploading to the server. If Bob ask her to share some data then Alice use share function of Box.com. But problem now is that how to share encrypted data. There are two severe ways: 1. Alice encrypt data with single secret key and share that secret key directly with the Bob. 2. Alice can encrypt data with distinct keys and send Bob corresponding keys to Bob via secure channel. In first approach, unwanted data also get expose to the Bob, which is inadequate. In second approach, no. of keys is as many as no. of shared

Table 1 files, which may be hundred or thousand as well as transferring these keys require secure channel and storage space which can be expensive.

Different Schemes	Cipher text size	Decryption Key Size	Encryption Type
Key assignment	Constant	Non Constant	Symmetric or Public Key
Symmetric Key encryption with compact key	Constant	Constant	Symmetric Key
IBE with compact key	Non Constant	Constant	Public Key
Attribute based Encryption	Constant	Non Constant	Public Key
KAC	Constant	Constant	Public Key

IV. LITERATURE SURVEY

SYMMETRIC-KEY ENCRYPTION WITH COMPACT KEY

Benelux et al. [2] presented an encryption scheme which is originally proposed for concisely transmitting large number of keys in broadcast scenario [3]. 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 cipher text classes) is as follows. A composite modulus is chosen where p and q are two large random primes. A master secret key is chosen at random. Each class is associated with a distinct prime. All these prime numbers can be put in the public system parameter. A constant-size key for set can be generated. For those who have been delegated the access rights for S' can be generated. However, it is designed for the symmetric-key setting instead. The content provider needs to get the corresponding secret keys to encrypt data which is not suitable for many applications. Because 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., [4]. However, sharing of decryption power is not a concern in these schemes.

IBE WITH COMPACT KEY

Identity-based encryption (IBE) (e.g., [5], [6], [7]) 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). There is a 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 content provider can take the public parameter and a user identity to encrypt a message. The recipient can decrypt this cipher text by his secret key. Guo et al. [8], [9] tried to build IBE with key aggregation. 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.[1] This significantly increases the costs of storing and transmitting cipher texts, which is impractical in many situations such as shared cloud. As Another way to do this is to apply hash function to the string denoting the class, and keep hashing repeatedly until a prime is obtained as the output of the hash function.[1] we mentioned, our schemes feature constant cipher text size, and their security holds in the standard model. In fuzzy IBE [10], 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.

ATTRIBUTE-BASED ENCRYPTION

Attribute-based encryption (ABE) [11], [12] allows each cipher text 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 cipher text can be decrypted by this key if its associated attribute conforms to the policy. For example, with the secret key for the policy (1 ∨ 3 ∨ 6 ∨ 8), one can decrypt cipher text tagged with class 1, 3, 6 or 8. However, the major concern in ABE is collusion-resistance but not the compactness of secret keys. Indeed, the size of the key often increases linearly with the number of attributes it encompasses, or the cipher text-size is not constant (e.g., [13]).

KEY-AGGREGATE CRYPTOSYSTEM

In key-aggregate cryptosystem (KAC), users encrypt a message not only under a public-key, but also under an identifier of cipher text called class. That means the cipher texts 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 cipher text classes.[1]With our example, Alice can send Bob a single aggregate key through a secure e-mail. Bob can download the encrypted photos from Alice's Box.com space and then use this aggregate key to decrypt these encrypted data. The sizes of cipher text, public-key, and master-secret key and aggregate key in KAC schemes are all of constant size. The public system parameter has size linear in the number of cipher text classes, but only a small part of it is needed each time and it can be fetched on demand from large (but non-confidential) cloud storage.

V. FRAMEWORK

The data owner establishes the public system parameter through Setup and generates a public/master-secret key pair through Key Gen. Data can be encrypted via Encrypt by anyone who also decides what cipher text 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 cipher text classes through

Extract. The generated keys can be passed to delegates securely through secure e-mails or secure devices. Finally, any user with an aggregate key can decrypt any cipher text provided that the cipher texts class is contained in the aggregate key via Decrypt. Key aggregate encryption schemes consist of five polynomial time algorithms as follows:

1. Setup ($1, \lambda, n$): The data owner establishes a public system parameter via Setup. On input of a security level parameter 1 and number of cipher text classes n , it outputs the public system parameter $param$. 2. Key Gen: It is executed by data owner to randomly generate a public/master-secret key pair (pk, msk) . 3. Encrypt (pk, i, m) : It is executed by data owner and for message m and index i , it computes the cipher text as C . 4. Extract (msk, S) : It is executed by data owner for delegating the decrypting power for a certain set of cipher text classes and it outputs the aggregate key for set S denoted by K_s . 5. Decrypt (K_s, S, I, C) : It is executed by a delegate who received, an aggregate key K_s generated by Extract. On input K_s , set S , an index i denoting the cipher text class cipher text C belongs to and output is decrypted result m .

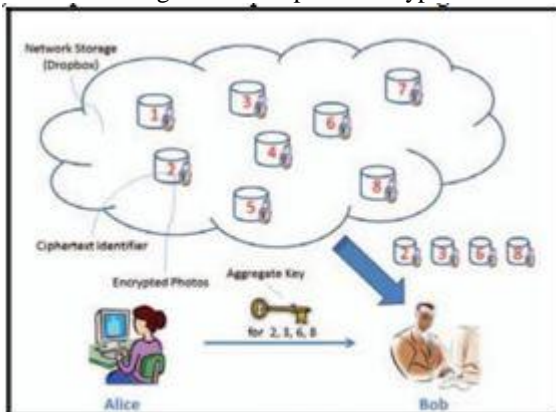


Figure 3: Framework

VI. CONCLUSION

To share data flexibly is a vital thing in cloud computing. Users prefer to upload their data on cloud and among different users. Outsourcing of data to server may lead to leak the private data of user to everyone. Encryption is a solution which provides to share selected data with desired candidate. Sharing of decryption keys in a secure way plays an important role. Public-key cryptosystems provide delegation of secret keys for different cipher text classes in cloud storage. The delegate gets securely an aggregate key of constant size. It is required to keep enough number of cipher text classes as they increase fast and the cipher text classes are bounded that is the limitation.

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," Proc. 10th Int'l Conf. Applied Cryptography and Network Security (ACNS), vol. 7341, pp. 526-543, 2012.
[2] 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.
[3] 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.
[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] Cheng-Kang Chu, Sherman S.M. Chow, 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 System, Vol 25, No. 2 February 2014.
[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] 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.
[8] 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.
[9] M. Chase and S. S. M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," in CM Conference on Computer and Communications Security, 2009, pp. 121–130.
[10] S. Singh, "Different Cloud Computing Standards a Huge Challenge", The Economic times, 4 June 2009.
[11] J. Urquhart, "The Biggest Cloud computing Issue of 2009 is Trust", C-NetNews, 7 Jan 2009. Wang et al., "Scientific Cloud Computing: Early Definition and Experience", Proc. 10th International Conference High-Performance Computing and Communications (HPCC 03)
[12] William Stallings, "Cryptography and Network Security Principles and Practices", Prentice Hall, New Delhi.
[13] National Institute of Standards and Technology. "Request for Candidate Algorithm Nominations for the Advanced Encryption Standard." Federal Register, September 12, 1997.
[14] Neuhvatal, J., et al. Report on the Development of the Advanced Encryption Standard. National Institute of Standards and Technology. October 2, 2000.
[15] Needham, R., and Schroeder, M. "Using Encryption for Authentication in Large Networks of Computers." Communications of the ACM,

- December 1978.
- [17] Daemen, J., and Rijmen, V. "Rijndael: The Advanced Encryption Standard." *Dr. Dobbs' Journal*, March 2001.
- [18] Daemen, J., and Rijmen, V. *The Design of Rijndael: The Wide Trail Strategy Explained*. New York, Springer-Verlag, 2002.