International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

A Survey Paper on Key Aggregate Cryptosystem: A Key Assignment Scheme for Scalable Data Sharing Over Cloud Storage

Rachana Gangwani[1], H. A. Hingoliwala[2]

[1]M.E (Computer) Department of Computer Engineering, Jayawantrao Sawant College of Engineering, Pune, India. Savitribai Phule Pune University, Pune, Maharashtra, India-411007

[2]M.E (Computer) Head of Department and Asst Prof (Computer) Department of Computer Engineering, Jayawantrao Sawant College of Engineering, Pune, India. Savitribai Phule Pune University, Pune, Maharashtra, India-411007

Abstract: ***Cloud storage is a model of data storage online in cloud which is accessible from multiple and connected resources, whose demand is greatly increasing. Cloud storage has a requisite functionality i.e. sharing of data securely, efficiently, flexibly over the safe network. Various schemes and methodologies are implemented to make data sharing more effective. Data security is crucial aspect in cloud storage. Providing security to a single file or a set of files is an important factor. With the introduction of encryption and decryption schemes, the storing, sharing and securing of data became rampant. The storing of these ciphertexts and the decryption keys is one of the major issues. There is a need for a mechanism which can minimize the cost of storing these ciphertexts and keys in a secured way. In this era of information, where there is presence of rich data, the true value lies in sharing, securing and storing them. Protecting user's data privacy is one of the critical aspects of cloud storage. The survey depicts some encryption schemes introduced in this data privacy for securely and efficient sharing of confidential data over a secure channel. The present research efforts concentrates more on aggregation of these keys into a single aggregate key which will in turn reduce the burden on the network overhead.***

Keywords: Cloud Storage, Attribute Based Encryption, Pre Defined Hierarchy, Compact Key Encryption, Identity Based Encryption, Key Aggregate Cryptosystem.

1. Introduction

Cloud storage is a cloud computing model in which data is stored on remote servers accessed from the internet, or cloud. It is maintained, operated and managed by a cloud storage service provider on a storage server that is built on the virtualization techniques. Cloud storage is also known as utility storage. Cloud storage works through a data centre virtualization, providing data owner and applications with a virtual storage architecture that is scalable according to the application requirements.

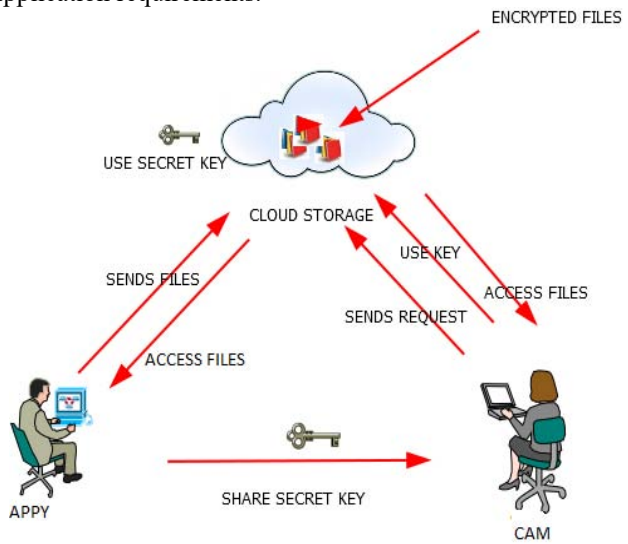

Figure 1: Sharing of data in cloud storage

In modern cryptography, encryption keys obtained are of two categories, symmetric and asymmetric (public) key. The public key encryption tends to be much more secured as it involves combination of two different keys, public and private key respectively. Cryptography is the way of storing and sharing the data in the form of that only those authenticated for it can access. It is the knowledge of securing the message by encoding it into an unreadable format. The basic goal of cryptography is the ability to send the data to the receiver in a way that prevents attackers from accessing it. This data is stored on cloud through the internet. The cloud storage is a cloud computing model in which the data is stored and remote servers are accessed from the internet. The cloud storage provider is maintaining, operating and managing the cloud storage on a server. Cryptographic mechanism is used to hide the data from unauthorized users. The most encryption algorithms can be broken and the data is stolen by the attacker. So a more realistic goal of cryptography is to make gaining the data too severe to be value it to the attacker.

Figure 2: The encryption process converts plaintext into ciphertext

Figure 3: The decryption process converts ciphertext into plaintext

Encryption is a technique of converting original message called clear text or plaintext, into unreadable format that can't understood by the attacker, called ciphertext. Once it can't be converted into plaintext, the user can't access it until it is decrypted. This enables the broadcast of top secret data over insecure channels without illegal disclosures. When data is stored on a computer, logical and physical access controls are confined it. When this same susceptible data is sent over internet, it can't take longer these controls for allowed and the data is in much more susceptible state as showed in figure 2.

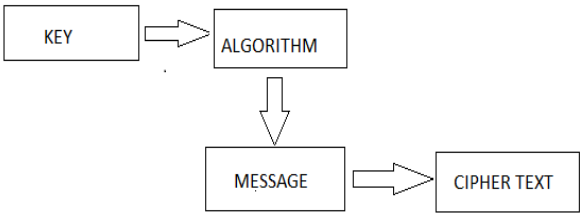

Figure 4: The algorithm is associated with the key and the result is applied to the message which produces the ciphertext.

Encryption and decryption processes are provided by a computer system is referred to as cryptosystem and hardware components and program codes are used to create this system. The cryptosystem uses an encryption algorithm for creating a ciphertext. Most algorithms are difficult mathematical formulas that are applied to the plaintext. Most encryption techniques use a secret value called a key, which is used to encrypt the plaintext and decrypt the ciphertext.

2. Literature Survey and Related Work

In this section we compare the various key assignment schemes with the basic KAC algorithm for sharing in secure cloud storage. We summarize our comparisons in Table 1. Various key assignment schemes are as mentioned below:

2.1 Predefined Hierarchical Scheme

In [1], Predefined Hierarchical Schemes the author aims to minimize the cost in storing and managing secret keys for general cryptographic use. Employing a tree structure a secret key for a given root can be used to procure the secret keys of its leaf nodes. Sandhu [2] proposed a technique to initiate a tree hierarchy of symmetric keys by using reiterated evaluations of pseudo random functions block cipher on stable secret. This notion can be verbalized from a tree to a graph. Most of these schemes construct keys for symmetric key cryptosystems, even though the key derivation may require modular arithmetic as used in public key cryptosystems, which are more expensive than" symmetric key operation" such as pseudorandom function[3]. Taking tree structure as an example, Cam can first classify the ciphertext classes according to their subjects like Fig 5.

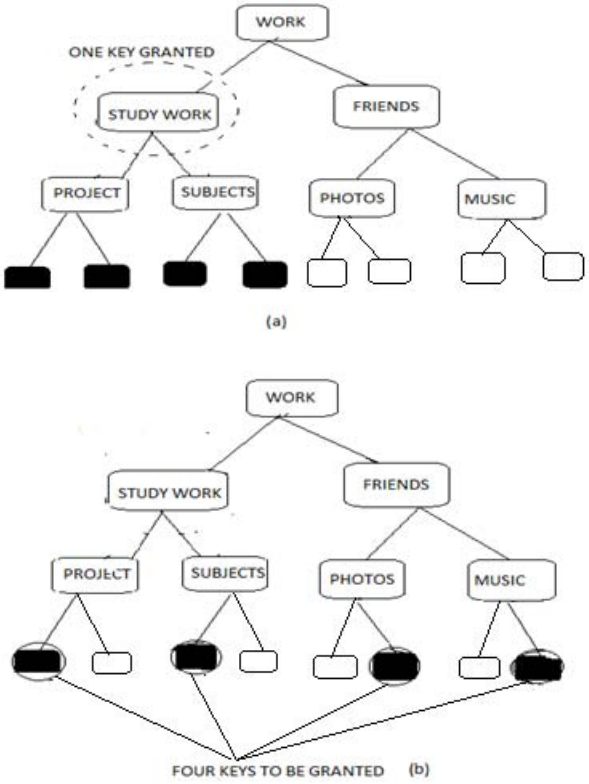

Figure 5: a) One key for hierarchy, b) Four keys for hierarchy

Each node in the tree represents a secret key, while the leaf nodes represent the keys for individual ciphertext classes. Filled rectangles represent the keys for the classes to be delegated and circles circumvented by dotted lines represent the keys to be granted. Note that every key of the non-leaf node can derive the keys of its child nodes. In Fig 5(a), if Cam wants to share all the files in the "study work" category, she only needs to grant the key for the node "study work", which automatically grants the delegate the keys of all the child nodes ("project","subjects"). This is the ideal case, where most classes to be shared belong to the same branch and thus a parent key of them is sufficient. However, it is still difficult for general cases. As shown in Fig 5(b), if user wants to share the files from different branches, she has to grant as many number of keys as the number of different branches containing these classes else the files from child nodes can also be accessed. One can see that this approach is not flexible when the classification share more complex and she wants to share different sets of files to different people [4].

2.2 Attribute-based Encryption

In attribute based encryption, the data owner with master secret key can obtain a secret key for the policy of attributes so that a ciphertext can be decrypted by this key if its associated attributes conforms to policy. Each attribute is associated with data this leads to increase in size of keys. For example with the secret key for the policy (2V3V6V8), one can decrypt ciphertext tagged with class 2, 3, 6 or 8 [5].The measure perturbed in attribute based encryption is collusion-resistance but not the compactness of secret keys. Actually,

the size of the key often increases linearly with the number of attributes it encompasses, or the ciphertext-size is not immutable. To delegate the decryption power of some ciphertexts without sending the secret key to the delegate, a useful primitive is proxy re-encryption (PRE) [6].

A PRE permits Appy to delegate to the server (proxy) the ability to transform the ciphertexts encrypted under his public-key into ones for. Nevertheless, Cam has to trustworthy the proxy that it only turns ciphertexts according to her instruction, which is what user wants to avoid at the first place. Even worse, if the proxy colludes with, some form of Cam's secret key can be recovered which can decrypt Cam's (convertible) ciphertexts without Appy's further help. That also means that the transformation key of proxy should be well protected. Using PRE just moves the secure key storage requirement from the delegate to the proxy. It is thus undesirable to let the proxy reside in the storage server. That will also be inconvenient since every decryption requires separate interaction with the proxy.

2.3 Symmetric-Key Encryption using Compact Key

The author Benaloh et al. [8], presented an encryption scheme which is originally proposed for concisely transmitting large number of keys in broadcast scenario [9]. The derivation of the key for a set of 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.

2.4 IBE using Compact Key

Identity-based encryption (IBE), [10] is a public-key encryption in which the public-key of a data 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 ciphertext by his secret key. Guo et al. [11] tried to build IBE with key aggregation. While there are an exponential number of identities and thus secret keys, only a polynomial number of them can be aggregated [7]. This significantly increases the costs of storing and transmitting ciphertexts, which is impractical in many situations such as shared cloud storage. 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. In fuzzy IBE [12], one single compact secret key can decrypt ciphertexts 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.

Table 1: Comparison between different key assignment schemes

Key Assignment Schemes	*Encryption Key*	*Decryption Key*
Predefined Hierarchical Scheme	Constant	Non constant
Attribute Based Encryption	Constant	Non constant
Identity Based Encryption	Non constant	Constant
Symmetric Key Encryption	Constant	Constant
KAC	Constant	Constant

3. Conclusion and Future Scope

Scalable sharing of data is the main issue in cloud computing. Data owner prefer cloud to upload their data with different user's. Uploading of data to server may lead to leakage of private data of data owner to everyone. Encryption is the best solution, which is provided to share selected data with desired user's. Sharing of decryption keys in secure way plays important role. Public-key cryptosystems provide delegation of secret keys for different cipher text classes in cloud storage. The delegate gets securely a constant size of an aggregate key in order to maintain limited number of cipher text classes.

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] 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.
[4] R. S. Sandhu, "Cryptographic Implementation of a Tree Hierarchy for Access Control," Information Processing Letters, vol. 27, no. 2, pp. 95–98, 1988
[5] C.-K. Chu, J. Weng, S. S. M. Chow, J. Zhou, and R. H. Deng, "Conditional Proxy Broadcast Re-Encryption," in Australasian Conference on Information Security and Privacy (ACISP '09), ser.LNCS, vol. 5594. Springer, 2009, pp. 327–342.
[6] S. S. M. Chow, J. Weng, Y. Yang, and R. H. Deng, "Efficient Unidirectional Proxy Re-Encryption," in Progress in Cryptology -AFRICACRYPT 2010, ser. LNCS, vol. 6055.Springer, 2010, pp.316–332.
[7] 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.
[8] 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.
[9] J. Benaloh, ―Key Compression and Its Application to Digital Fingerprinting,‖ Microsoft Research, Tech. Rep., 2009.
[10] 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.
[11] 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.
[12] 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.

Author Profile

Rachana Gangwani is currently pursing M.E. (Computer) from Department of computer Engineering, Jayawantrao Sawant College of Engineering, Pune, India. Savitribai Phule Pune University, Pune, Maharashtra, Pune-411007.She received her B.E. (Computer) Degree from Jayawantrao Sawant College of Engineering, Pune, India. Savitribai Phule Pune University, Pune, Maharashtra, Pune-411007.

H.A Hingoliwala, M.E (Computer) Head of Department and Asst Prof (Computer) Department of Computer Engineering, Jayawantrao Sawant College of Engineering, Pune, India. Savitribai Phule Pune University, Pune, Maharashtra, India-411007. He is awarded with the degree of B.E. (Computer) and M.E. (Computer). He has 17 years of teaching experience. His area of interest is image processing.