International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

Comparative Analysis of Cryptographic Algorithms for Storing Shared Data on the Cloud

Kavitha K[1], Manjula R[2]

[1]M. Tech, Department of Computer Science, Rajiv Gandhi Institute of Technology & Engineering, Bangalore, India

[2]Assistant professor, Department of Computer Science, Rajiv Gandhi Institute of Technology & Engineering, Bangalore, India

Abstract: ***Cryptography in the field of information technology is an art to protect data privacy using standard mathematical techniques. Cloud computing, one of the emerging techniques to lease computing resources on demand, makes use of remote data storage where data owner does not possess direct control over her data. To protect privacy of users' data and to enable user to verify integrity of the data stored on remote location, modern cryptographic techniques are used. So, if users paying awareness in selection of proper cryptographic scheme then it can achieve confidentiality without losing performance of Cloud. For Achieving high performance and security users can select any of modern encryption techniques. Here considering time and key difference to compare which algorithm is best in security and speed over the data. 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 delegate 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.***

Keywords: outsourced data, share data, authentication, Cryptography, data owner, encrypts data and Key aggregate cryptosystem (KAC).

1. Introduction

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.

It is necessary to make sure that the data integrity without compromising the anonymity of the data user. To ensure the integrity the user can verify metadata on their data, upload and verify metadata [3].

The main concern is how to share the data securely the answer is cryptography. The question is how can the encrypted data is to be shared. The user must provide the access rights to the other user as the data is encrypted and the decryption key should be send securely. For an example Alice keeps her private data i.e. photos on dropbox and she doesn't want to share it with everyone. As the attacker may access the data so it is not possible to rely on predefine privacy preserving mechanism so she all the photos were encrypted by her on encryption key while uploading it.

Naturally, there are two extreme ways for her under the traditional encryption paradigm: . Alice encrypts all files with a single encryption key and gives Bob the corresponding secret key directly. . Alice encrypts files with distinct keys and sends Bob the corresponding secret keys. Obviously, the first method is inadequate since all unchosen data may be also leaked to Bob. For the second method, there are practical concerns on efficiency. The number of such keys is as many as the number of the shared photos, say, a thousand. Transferring these secret keys inherently requires a secure channel, and storing these keys requires rather expensive secure storage. The costs and complexities involved generally increase with the number of the decryption keys to be shared. In short, it is very heavy and costly to do that.

Licensed Under Creative Commons Attribution CC BY

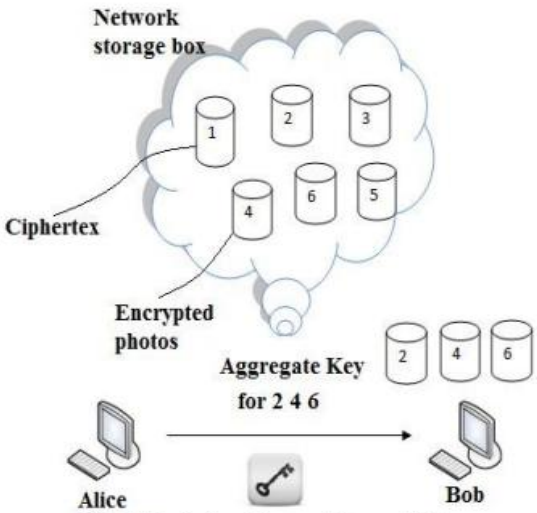

Fig 1 File sharing between Alice and Bob

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 an 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.

2. Literature Survey

Cloud computing is visualized as architecture for succeeding generation. It has many facilities though have risks 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 loud 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 audit ability, 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 uploaded [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 than 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 ciphertext. The user key and the attribute are matched if it matches it can decrypt a particular ciphertext. When there is k attributes are overlay among the ciphertext and a private key the decryption is granted [5].

TABLE I
COMPARISON BETWEEN KAC AND OTHER SCHEMES

	Decryption Key size	Ciphertext size	Encryption Type
Key Assignment Schemes for predefined hierarchy	Non constant	constant	Symmetric key or Public key
Symmetric key Encryption with compact key	constant	constant	Symmetric key
IBE with compact key	constant	Non constant	Public key
Attribute based Encryption	Non constant	constant	Public key
KAC	constant	constant	Public key

3. Key-Aggregate Encryption

Framework:
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 takes implicit 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 ciphertext class. The algorithm encrypts message m and produces a ciphertext C such that only a user that has a set of attributes that satisfies the access structure is able to decrypt the message.

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

- **Setup:** executed by the data owner to setup an account on an untrusted server. On input a security level parameter 1_ and the number of ciphertext classes n (i.e., class index should be an integer bounded by 1 and n), it outputs the public system parameter pram, which is omitted from the input of the other algorithms for brevity.
- **KeyGen:** executed by the data owner to randomly generate a public/master-secret key pair.

Licensed Under Creative Commons Attribution CC BY

- **Encrypt:** executed by anyone who wants to encrypt data. On input a public-key pk, an index i denoting the ciphertext class, and a message m, it outputs a ciphertext C.
- **Extract:** executed by the data owner for delegating the decrypting power for a certain set of ciphertext classes to a delegate. 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.
- . **Decrypt:** executed by a delegate 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 2 S.

3.2 Sharing Encrypted Data:

A canonical application of KAC is data sharing. The key aggregation property is especially useful when we expect the delegation to be 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.

Here, we describe the main idea of data sharing in cloud storage using KAC, illustrated in Fig. 2. Suppose Alice wants to share her data m1;m2; . . .;m_ on the server. She first performs Setupð1_; nÞ to get param and execute KeyGen to get the public/master-secret key pair ðpk; mskÞ. The system parameter param and public-key pk can be made public and master-secret key msk should be kept secret by Alice. Anyone (including Alice herself) can then encrypt each mi by Ci ¼ Encryptðpk; i;miÞ. The encrypted data are uploaded to the server.

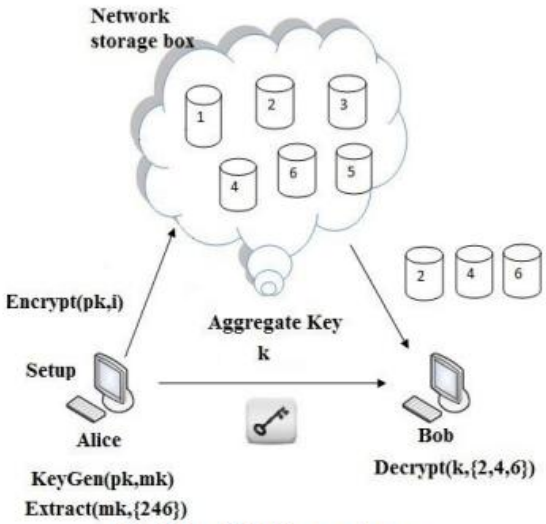

Fig 2 Use of KAC for data sharing

4. Related Work

This section we compare our basic KAC scheme with other possible solutions on sharing in secure cloud storage.

4.1 Cryptographic Keys for a Predefined Hierarchy:

We start by discussing the most relevant study in the literature of cryptography/security. Cryptographic key assignment schemes aim to minimize the expense in storing and managing secret keys for general cryptographic use. Utilizing a tree structure, a key for a given branch can be used to derive the keys of its descendant nodes (but not the other way round).

TABLE 1
Comparisons between Our Basic KAC Scheme and Other Related Schemes

	Decryption key size	Ciphertext size	Encryption type
Key assignment schemes for a predefined hierarchy (e.g., [7])	most likely non-constant (depends on the hierarchy)	constant	symmetric or public-key
Symmetric-key encryption with Compact Key (e.g., [8])	constant	constant	symmetric-key
IBE with Compact Key (e.g., [9])	constant	non-constant	public-key
Attribute-Based Encryption (e.g., [10])	non-constant	constant	public-key
KAC	constant	constant	public-key

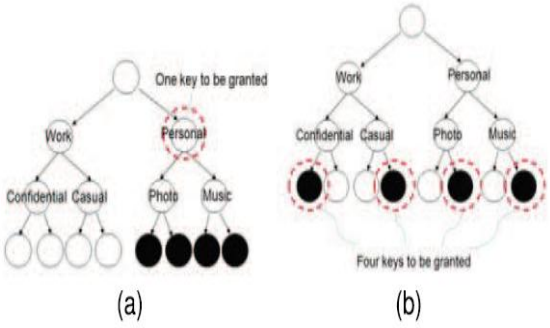

Fig. 3. Compact key is not always possible for a fixed hierarchy.

4.2 Symmetric-Key Encryption With Compact Key

Benaloh 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

Licensed Under Creative Commons Attribution CC BY

those who have been delegated the access rights for S' can be generated. However, it's designed for the symmetric-key setting instead.

4.3 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 ciphertext 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.

4.4 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 ciphertexts 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. [1]

A Basic Construction
The design of our basic scheme is inspired from the collusion-resistant broadcast encryption scheme proposed by Boneh et al. [31]. Although their scheme supports constant-size secret keys, every key only has the power for decrypting ciphertexts associated to a particular index. We, thus, need to devise a new Extract algorithm and the corresponding Decrypt algorithm.

Algorithm:
Algorithms for Secrete key encryption
1. Start
2. Select file which we want to send.
3. Suppose files are f1, f2, f3, generate random key of respective file. Suppose that are k1,k2,k3 Let,k1=123,k2=456,k3=789
4. Combine the keys and add separator (ie.0) between them
5.Take one Quadratic equation, F(x)=n1x+n2x2 +s Let,n1=19,n2=6,x=3(as sending files are 3)
6.F(x)=(19*3)+6*(32)+s F(x)=57+54+s F(x)=111+s
7. F(x) =111+12304560789 =1.230456e10
8.1.230456e10 this is aggregate key send via email.

5. Result Analysis

In figure 1 it's about key difference graph its clearly showing that Key aggregate cryptosystem algorithm is more secure because key length is more. If larger the key there will be a more security. So from that graph the KAC algorithm is best in providing security over the data comparing to AES, DES, RSA and ECC.

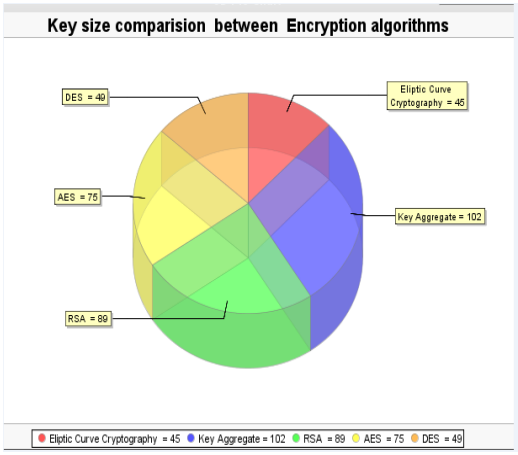

Figure 1: Key Difference

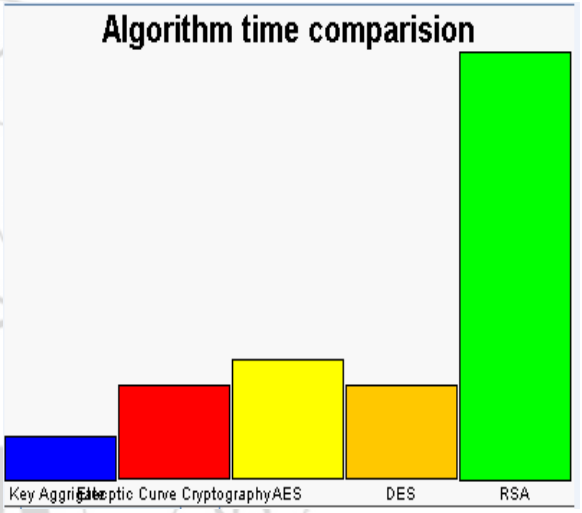

Figure 2: Time Difference

Figure 2: from the figure 2 its clearly showing that the KAC algorithm is taking less encryption time for encrypting the data, so its clear the the KAC algorithm works fast compare to AES, DES, RSA and ECC

6. Conclusion and Future Scope

This application is useful in cloud computing to scalable system using different cryptographic algorithm. How to defend users‟ data privacy is a central problem of cloud storage. With more mathematical concept, cryptographic schemes are getting more multipurpose and often involve multiple keys for a single system. In this System we consider how to ―compact‖ secret keys in public-key cryptosystems which support delegation of secret keys for different cipher text classes in cloud storage. Our approach is more flexible than hierarchical key assignment which can only save storage spaces if all key-holders share a similar set of privileges. A limitation in our work is the predefined bound of the number of maximum cipher text classes. In cloud storage, the number of cipher texts usually grows rapidly. So we have to reserve enough cipher text classes for the future extension.

Licensed Under Creative Commons Attribution CC BY

References

[1] S. G. Akl and P. D. Taylor, "Cryptographic Solution to a Problemb of Access Control in aHierarchy," 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,‖ inProceedings 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] R. S. Sandhu, "Cryptographic Implementation of a Tree Hierarchy for Access Control," Information Processing Letters, vol. 27, no. 2, pp.95–98, 1988.
[6] 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.
[7] Q. Zhang and Y. Wang, "A Centralized Key Management Schem for Hierarchical Access Control,‖ in Proceedings of IEEE Global Telecommunication Conference (GLOBECOM'04). IEEE, 2004, pp. 2067–2071.
[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
[11] Proceedings of Advances in Cryptology – CRYPTO '01, ser. LNCS, vol. 2139. Springer, 2001, pp. 213–229.
[12] 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.

Licensed Under Creative Commons Attribution CC BY