International Journal of Scientific Engineering and Applied Science (IJSEAS) – Volume-2, Issue-7,July 2016
ISSN: 2395-3470
www.ijseas.com

SURVEY ON KEY AGGREGATION SYSTEM FOR SECURE SHARING OF CLOUD DATA

Mr.Sunil S.khatal[1], Mr. K.S.Kahate[2]
[1] *ME 2nd year ,Computer Engineering,SPCOE ,Dumberwadi,Otur*
[2] *ME Co-Ordinator,Computer Engineering,Dumberwadi,Otur*

Abstract

For sharing data a cryptosystem is important, cryptosystem is a system that provides encryption and decryption schemes. An intrusion detection system observ the security breaches while sharing of data. Cryptosystem integrated Intrusion detection system makes a leakage resilient system. New public key cryptosystems produces constant size cipher texts such that decryption for any set of cipher text can be delegated efficiently. In this article we describe a cryptographic technique in which of secret keys are combined to make them compact as a single key. Here, power of all keys being aggregated. In other words the key owner will share an aggregate key with the second party on demand. The receivers will able to decrypt the files on cipher text choices but other encrypted files outside the cipher text set remains confidential. To make system leakage resilient, provision for intrusion detection and prevention is also made for secure data sharing between participants.

Keywords— **Database storage, data sharing, Encryption, Decryption, Cryptography, AES, Huffman coding, LSB Algorithm , Steganography**

I. INTRODUCTION

Cloud storage is gaining popularity recently. In enterprises,the rise in need for data outsourcing demands the strategic management of corporate data. It is also used as a core technology behind many online services for personal applications. Nowadays, it is easy to apply for free email accounts, social networking sites accounts; file sharing and/or remote access, with storage size more than 25GB. Users can access almost all of their files and emails by a mobile phone in any corner of the world.

Cryptography is the method of storing and transmitting the data in the form of only those for intended for it can read and write the data. Recently cloud gaining more popularity in enterprise setting we see the rise in demand for data outsourcing which assist in strategic management of corporate data. It is also used as a core technology in many online services for personal application. Now a day it is easy to apply for photo album, email, file sharing, remote access. Cryptography is an efficient way of protecting the sensitive information.as it is stored on media or transmitted through the network communication path. A traditional way to ensure data privacy is to relay on the server to provide the access control after authentication which means any unexpected privilege unexceptional escalation will expose all data. Although the ultimate goal of our key-aggregation cryptosystem and the mechanism that it make up, is that to hide information unauthorized user. We have introduced the KAC(Key –Aggregate Cryptosystem) because most of the algorithm can be broken and the information can be revealed, if the attacker has enough time, desire and resources.

A. *INTRODUCTION TO PROPOSED SYSTEM:*

In this paper, By using key aggregation cryptosystem, we make decryption key more powerful the in the sense that it allows decryption of multiple cipher text without increasing its size .Specially, our problem statement is "To design an efficient public-key encryption scheme which supports flexible delegation in the sense by the any subset of the cipher texts (produced by the encryption scheme) is decrypt able by a constant size decryption key (generated by the owner of the master- secret key)."We solve this problem by introducing a special type of public-key encryption which we call key- aggregate cryptosystem (KAC).In 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 into different classes. The key owner holds a master-secret called master-secrete key, which can be used to extract secrete key for different classes. More importantly, the extracted key have can be an aggregate key which can be aggregate key which is as secrete 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. The advantages of our proposed system is the extracted key have can be an aggregate key which is as compact as a

secret key for a single class. The delegation of decryption can be efficiently implemented with the aggregate key.

II. KEY-AGGREGATE ENCRYPTION

A. Asymmetric key Encryption:

Asymmetric encryption (also known as public-key encryption) is a cryptography technique that uses public

and private key pairs to encrypt and decrypt data respectively. The private key is a closely guarded secret, while the public key can be freely distributed over untrusted networks. You do not worry who has your public key (you could print it on a 100 foot tall banner if you so desired), but keep your private key secrete. The disadvantage of symmetric key encryption is that it assumes that the two parties involved have already agreed upon an encryption key in a secure manner. Any insecurity in the key exchange mechanism compromises the security of the data. These disadvantages of symmetric key are overcome in asymmetric key encryption. Conversely the disadvantage of asymmetric encryption algorithms is that they are more computationally expensive and hence slower to work with.

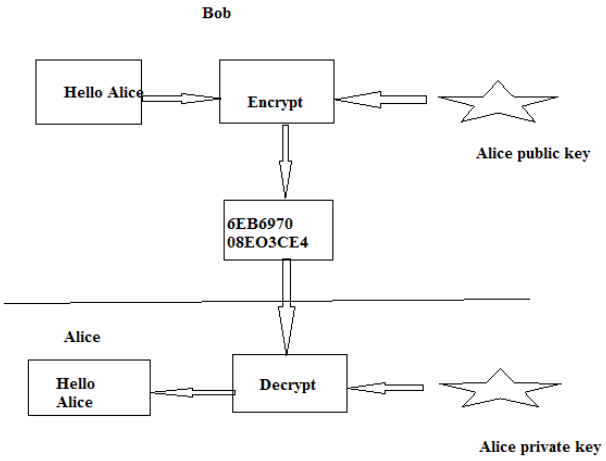

Fig. 1 : An Ansymmetric Key-Encryption and Decryption of data.

An asymmetric key encryption scheme, anyone can encrypt messages using the public key, but only the holder of the paired private key can decrypt. Security depends on the secrecy of the private key.

III. System Architecture

A. *Framework*

The user1 establishes the public system parameter via setup and generate master secrete key via KeyGen. Messages can be encrypted via encrypt by anyone who also decides what ciphetext class is associated with the plaintext message to be encrypted. The user1 can use the master secret key to generate an aggregate description key for a set of ciphertext classes via Extract. The generated keys can be passed to delegates securely. Finally, any user with an aggregate key can decrypt any ciphertext provided that the cipher text's class is contained in the aggregate key via Decrypt.

Setup (1^λ,n): Executed by user1 to setup an account on an untrusted server. On input a security level parameter 1^λ and the number of ciphertext classes n, it output the public system parm, which is omitted from the input of the other algorithm for brevity.

1) KeyGen():

Executed by user1 to randomly generate public/ master - secrete key(pk,msk).

2) Encrypt (pk ,i, m):

Executed by anyone who want to encrypt the data On input a public-key pk, an index i denoting the ciphertext class, and a message m, it outputs a ciphertext C.

3) Extract (msk, S):

Executed by user1 by delegating the decrypting power for a certain set of ciphertext classes to a deligatee On input the master-secrete key msk and a set S of indices corresponding to different classes, it outputs the aggregate key for set S denoted by Ks.

4) Decrypt (Ks, S, i, C):

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 Є S .

5) Correctness:

For any integers λ and n, any set S Є {1,……,n}, any index I Є S and any message m, Pr[Decrypt(Ks, S, i, C) =m: param←setup(1^λ, n),(pk, msk)←KeyGen(), Encrypt(pk,I,m), Ks←Extract(msk,S)]=1.

6) Compactness:

For any integers λ, n, any set S, any index I Є S and any message m; param← Setup(1^λ, n) , (pk, msk) ← KeyGen(), Ks←Extract(msk, S) and C ←Encrypt(pk, I, m); |Ks| and |C| only depend on the security parameter λ but independent of no of classes n.

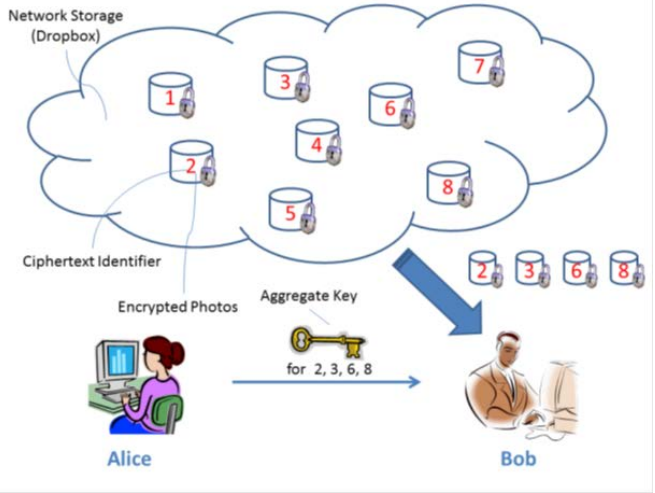

Fig. 2: System Architecture

In this our key aggregate cryptosystem, when user1 want to share his important data with another side user2 via secure transmitting medium. Firstly the user who wants to share his data, it will encrypt that files by using aggregate key. Client user1 upload that files by using its public key on cloud based network system. After that sender will send message to receiver with its aggregate key and that file no.that he have to decrypt via a secure email. Then at receiver side the user will download the data by using master key. After downloading that files, the user will use the aggregate key to decrypt that file. In this way file sharing can be completed by using asymmetric key encryption.

IV. Related work

In this section we compare our basic KAC scheme with other possible various solutions on data sharing in secure cloud storage.

A. Compact key Symmetric-key encryption

In the symmetric-key schemes, the encryption and decryption and decryption keys are the same for both communicating parties. Thus communicating parties must have same key before they can achieve the secrete communication.

Cryptography is the art of using mathematics to encrypt and decrypt data. The primary goal in making private key symmetric ciphers useful is distribution of private keys, communicating parties would first should to be holding shared private key. Public key cryptography solves this conundrum by implementing encryption with two keys, a well-known public key and a private key. Only the receiver knows the private key value. The receiver's public key, on the other hand, is widely published by trusted sources.

On average, the number of keys increases with the number of branches. It is unlikely to come up with a hierarchy that can save the number of total keys to be granted for all individuals simultaneously.

B. Compact Key Identity based Encryption (IBE)

IBE is a type of public-key encryption in which the public-key of a user1 can be set as an identity string of the user1 (e.g., an email address). There is a trusted party called private key generator in IBE which holds a master-secret key and issues a secret key to each user1 with respect to the user identity. The encryptor can take the public parameter and a user identity to encrypt a message. The user2 can decrypt this ciphertext by his secret key. One of their schemes assumes random oracles but another does not. 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. This greatly increases the costs of storing and transmitting ciphertext, which is impractical in many situations such as shared cloud storage. As we mentioned, our schemes feature constant ciphertext size, and their security holds in the standard model.

C. RSA Asymmetric Algorithm:

Rivest-Shamir-Adleman is the most commonly used asymmetric algorithm (public key algorithm). It can be used both for encryption and for digital signatures. The security of RSA is generally considered equivalent to factoring, although this has not been proved.

RSA computation occurs with integers modulo n = p * q, for two large secret primes p, q. To encrypt a message m, it is exponentiated with a small public exponent e. For decryption, the recipient of the ciphertext c = me (mod n) computes the multiplicative reverse d = e-1 (mod (p-1)*(q-1)) (we require that e is selected suitably for it to exist) and obtains cd = m e * d = m (mod n). The private key consists of n, p, q, e, d (where p and q can be omitted); the public key contains only n and e. The problem for the attacker is that computing the reverse d of e is assumed to be no easier than factorizing n. The key size should be greater than 1024 bits for a reasonable level of security. Keys of size, say, 2048 bits should allow security for decades. There are actually multiple incarnations of this algorithm; RC5 is one of the most common in use, and RC6 was a finalist algorithm for AES.

V. Future work

The future work of our project is compact aggregate key can be conveniently sent to others or be stored in smart card .We provide additional security in future for the transmission of master key. We will divide the whole file into individual parts and provide separate aggregate key for each file part.

VI. Conclusion

We have been concluded that how to protect user's data privacy is a central question of cloud storage. With more mathematical tools, cryptographic schemes are getting more versatile and often involve multiple keys for a single

application. In this project, we consider how to compress secret keys in public-key cryptosystems which support delegation of secret keys for different ciphertext classes in cloud storage. No matter which one among the power set of classes, the delegatee can always get an aggregate key of constant size. Our approach is more flexible than hierarchical key assignment which can only save spaces if all key-holders share a similar set of privileges.

ACKNOWLEDGMENT

This work is supported by the our project guide
Prof. S. A. Kahate (ME. Comp).

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,? in Applied Cryptography and Network Security - ACNS 2012, ser. LNCS, vol. 7341. Springer, 2012, pp. 526, 543.

[2] L. Hardesty, Secure computers aren't so secure,? MIT press, 2009, http://www.physorg.com/news176107396.html.

[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] S. S. M. Chow, C.-K Chu, X. Huang, J. Zhou, and R. H. Deng, Dynamic Secure Cloud Storage with Provenance,? in Cryptography and Security: From Theory to Applications- Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday, ser. LNCS, vol. 6805. Springer, 2012, pp. 442?464.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, Aggregate and Veri_ably Encrypted Signatures from Bilinear Maps,? in Proceedings of Advances in Cryptology - EUROCRYPT ?03, ser. LNCS, vol. 2656. Springer, 2003, pp. 416?432.

[7] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, Dynamic and E_cient Key Management for Access Hierarchies,? ACM Transactions on Information and System Security (TISSEC), vol. 12, no. 3, 2009.

[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] F. Guo, Y. Mu, Z. Chen, and L. Xu, ?Multi-Identity Single-Key Decryption without Random Oracles,? in Proceedings of 33 Information Security and Cryptology (Inscrypt ?07), ser. LNCS, vol. 4990. Springer, 2007, pp. 384? 398.

[9] 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