Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 3, March 2015, pg.658 – 662

RESEARCH ARTICLE

Secure Data Sharing in Web by using Key Aggregator Cryptosystem

L.Alexstephen
Final year Student
Department of CSE
IFET college of Engineering
Villupuram-TN, India
Alexstephencse@gmail.com

R.Rajmohan
Assistant Professor
Department of CSE
IFET college of Engineering
Villupuram-TN, India
rjmohan89@gmail.com

D.Dinagaran
Assistant Professor
Department of CSE
IFET college of Engineering
Villupuram-TN, India
ddinagaran@gmail.com

Abstract: Data sharing is an important functionality in cloud storage. In this article, we show how to securely, efficiently, and flexibly share data with others in cloud storage. We describe new public- key cryptosystems which produce constant-size cipher texts such that efficient delegation of decryption rights for any set of cipher texts are possible. The novelty 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. In other words, 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. This compact aggregate key can be conveniently sent to others or be stored in a smart card with very limited secure storage. We provide formal security analysis of our schemes in the standard model. We also describe other application of our schemes. In particular, our schemes give the first public-key patient-controlled encryption for flexible hierarchy, which was yet to be known.

Keywords- kac, key encryption, decryption, aggregator

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 important functionality of cloud storage, because user can share data from anywhere and anytime to anyone [1]. 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. 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.

2. SYMMETRICKEY ENCRYPTION

SYMMETRIC-KEY ENCRYPTION WITH COMPACT KEY Bengali et al. presented an encryption scheme which is originally proposed for concisely transmitting large number of keys in broadcast scenario. 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 ciphertext 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., . However, sharing of decryption power is not a concern in these schemes.

IBE WITH COMPACT KEY 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) [2] [7]. 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 [3]. Guo et al. 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. 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. We mentioned, our schemes feature constant ciphertext size, and their security holds in the standard model.

In fuzzy IBE , 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.

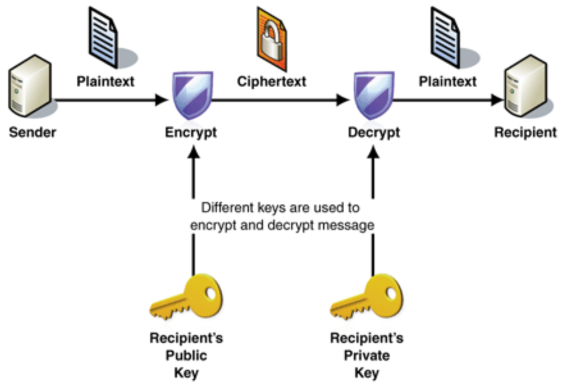


Fig.1key exchange using in the data sharing

3. 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 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 ciphertext classes [6]. 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 ciphertext, public-key, 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 ciphertext 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.

4. PROPOSED ARCHITECTURE

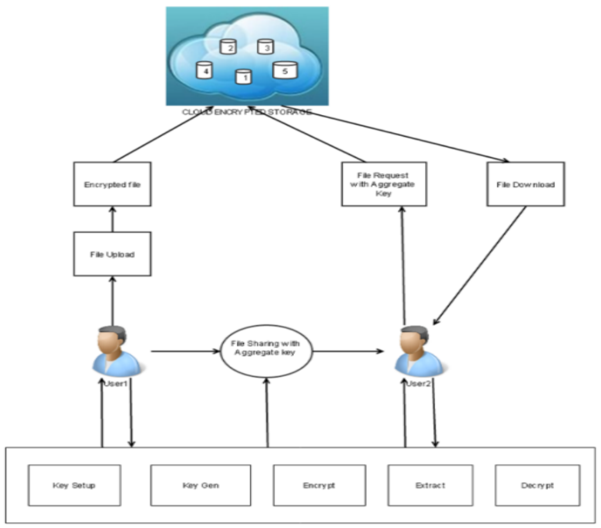


Fig.2 proposed system

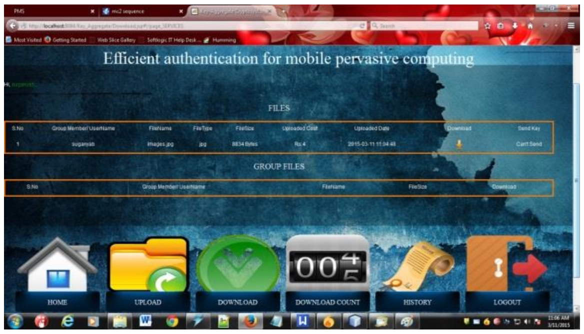
5. PROPOSED FRAMEWORK

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 delegates 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. Key aggregate encryption schemes consist of five polynomial time algorithms as follows:
1. Setup (1λ , n) : The data owner establish public system parameter via Setup. On input of a security level parameter 1λ and number of ciphertext classes n , it outputs the public system parameter param
2. KeyGen: 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 ciphertext as C.
4. 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 Ks.
5. Decrypt (Ks, S, I, C): It is executed by a delegate who received, an aggregate key Ks generated by Extract. On input Ks, set S, an index i denoting the ciphertext class ciphertext C belongs to and output is decrypted result m.

6. EXPERIMENT WORK AND SETUP

Our experimental setup is windows7,8 operating system. Software is using as netbeans 7.3.1.our server is mysql, and platform based on java, connectivity based on jdbc Driver.

7. RESULT ANALYSIS



S.No	Group Member/ UserName	FileName	FileType	FileSize	Uploaded Cost	Uploaded Date	Download	Send Key
1	suganyab	images.jpg	jpg	8834 Bytes	Rs.4	2015-03-11 11:04:48		Can't Send

S.No	Group Member/ UserName	FileName	FileSize	Download

Fig.3 User Interface

Data to be uploading and download in the web when user can be registration to upload and download to visualize. User to view key based to exchange the data using to view so user easy to download and secure.

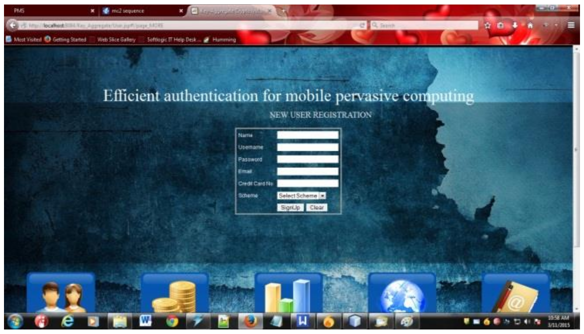


Fig.4 User Interface

8. CONCLUSION AND FUTURE WORK

Many key cryptosystem can use but some of the algorithm used to share data to be secure. key can be using to identified but so algorithm can be used many so in future it may be modify and reuse not only modified in may change to reuse to view to using advanced algorithm.

References

[1] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, "Prisense: Privacy- Preserving Data Aggregation in People-Centric Urban Sensing Systems," Proc. IEEE INFOCOM, pp. 758-766, 2010.

[2] Z. Erkin and G. Tsudik, "Private Computation of Spatial and Temporal Power Consumption with Smart Meters," Proc. Int'l Conf. Applied Cryptography and Network Security (ACNS '12), pp. 561-577, 2012.

G. Acs and C. Castelluccia, "I Have a Dream!: Differentially Private Smart Metering," Proc. 13th

Int'l Conf. Information Hiding (IH '11), pp. 118-132, 2011.

[3] M. Jawurek and F. Kerschbaum, "Fault-Tolerant Privacy- Preserving Statistics," Proc. 12th Privacy Enhancing Technologies Symp.(PETS '12), 2012.

[4] M. Shao, Y. Yang, S. Zhu, and G. Cao, "Towards Statistically Strong Source Anonymity for Sensor Networks," Proc. IEEE INFOCOM, 2008.

[5] C. Castelluccia, "Efficient Aggregation of Encrypted Data in Wireless Sensor Networks," Proc. Second Ann. Int'l Conf. Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '05), pp. 109-117, 2005.

[6] M. Bellare, "New Proofs for NMAC and HMAC: Security without Collision-Resistance," Proc. 26th Ann. Int'l Conf. Advances in Cryptology (CRYPTO '06), pp. 602-619, 2006.

[7] [C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," Proc. Third Conf. Theory of Cryptography (TCC '06), 2006. [32] Q. Li and G. Cao, "Providing Privacy-Aware Incentives for Mobile Sensing," Proc. IEEE PerCom, 2013.