ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 19, April 2016

Privacy Policy Interface of User Uploaded File in Cloud Storage Using Blowfish and RSA Algorithm

Anjali.K.Sunandan[1] , Nibedita Lahkar[2] , Prabakeran[3]
UG Scholars [1, 2], Assistant professor [3], *KCG college of technology, Tamil Nadu, India*
anjalisunandhan@gmail.com [1], nibe12lahkar@gmail.com [2]

***Abstract*— Cloud computing technology is widely used so that the data can be outsourced on cloud can accessed easily. Different members can share that data through different virtual machines but present on single physical machine. The need is to share data securely to users. The cloud service provider and users authentication is necessary to make data security. Privacy preserving in cloud is important make sure the users identity is not revealed to everyone. On cloud anyone can share data. Cryptography helps the user to share the data to in safe manner. So user encrypts data and uploads on server. The encryption and decryption keys may be different for each set of data. Only those set of decryption keys are shared that the selected data can be decrypted. Here a public-key cryptosystems which generate a ciphertext which is of constant size. So that the ciphertext transfer of the decryption rules for each set number. The difference is one can collect a set of secret keys and make them as small size as a single key with holding the same ability of all the keys that are formed in a group. This compact aggregate key can be efficiently sent to others or to be stored in a smart card with little secure storage.**

Index Terms—Cloud storage, Attribute base encryption, Identity base encryption, data sharing, key- aggregate encryption.

1.INTRODUCTION

Cloud storage

Cloud storage is a model of data storage in which the digital data is stored in logical pools, the physical storage spans multiple servers and the physical environment is typically owned and managed by a hosting company.

Attribute base encryption

Attribute based encryption is a type of public key encryption in which the secret key of a user and the ciphertext are dependent upon attributes.

Identity base encryption

Identity based encryption is an important primitive of ID based cryptography. As such it is a type of public key encryption in which the public key of a user is some unique information about the identity of user.

Data sharing

The ability to share the same data resource with multiple application or users. It implies that the data are stored in one or more server in the network.

Key aggregate encryption

Encryption keys also come with two flavors symmetric key or asymmetric key. Using symmetric encryption, when Alice wants the data to be originated from a third party, she has to give the encrypted her secret key; obviously, this is not always desirable. By contrast, the encryption key and decryption key are
different in public-key encryption

Cloud computing is widely increasing technology; data can be saved on cloud remotely and can have access to huge applications with quality services which are shared among customers. As increase in outsourcing of data the cloud computing serves does the management of data [1].

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's 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].

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.

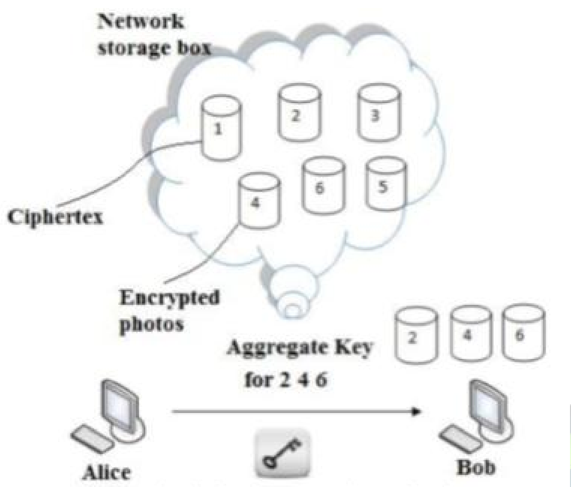

Fig 1 File sharing between Alice and Bob

Suppose some day she wants to share few photos with her friend Bob, either she can encrypt all photos with one key and send to him or she can create encrypt with different keys and send it. The un-chosen data may be leaked to Bob if the single key generated for encryption so create distinct keys of data and send single key for sharing.

A new way for public-key encryption is used called as key-aggregate cryptosystem (KAC)[1]. The encryption is done through an identifier of Ciphertext known as class, with public key. The classes are formed by classifying the ciphertext. 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 dropbox through the aggregate key. This is shown in figure1.

2. LITERATURE SURVEY

[1]Cheng-Kang Chu et.al explained that attribute-based encryption (ABE) allows each ciphertext 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 ciphertext can be decrypted by this key if its associated attribute conforms to the policy. For example, with the secret key for the policy (2V3V6V8), one can decrypt ciphertext tagged with class 2,3,6 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 ciphertext size is not constant [1]

[2]. S.S.M.Chow et.al explained that loud 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 users 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 [2].

[3]. C. Wang et.al explained that 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 [3].

[4]. B. Wang et.al explained that 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 [4].

[5].Sherman S.M. Chow, et.al explained that 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 are k attributes are overlay among the ciphertext and a private key the decryption is granted [5].

[6].V.Goyal et.al explained that 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].

[7]Y. Sun et.al explained that 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 ciphertext is decrypted using secret key [7].

3. PROPOSED IDEA

The aggregation cryptosystem consists of efficient Key Aggregate Cryptosystem algorithm. The data owner set up the general public parameter using Setup and creates a public/private key and combines using KeyGen. The secret file is encrypted utilizing RSA and Blowfish algorithm. The information owner will make use the master-secret to come up with aggregate decipherment key for a collection of data files. The generated keys may be passed to delegates securely (via secure e-mails or secure devices). Finally, any user with aggregate key will decrypt the data file and download it.

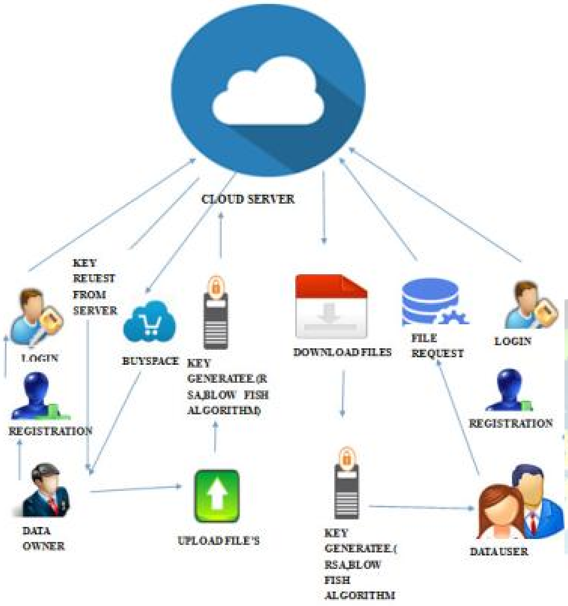

Fig 3. Architecture Diagram

MODULES

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 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.

Input= public key pk, an index i, and message m

Output = ciphertext C.

3.3 KeyGen Phase

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

3.4 Extract Phase

This is executed by the data owner for delegating the decrypting power for a certain set of ciphertext classes to a delegate.

Input = master-secret key mk and a set S of indices corresponding to different classes

Outputs = aggregate key for set S denoted by kS.

3.5 Decrypt Phase

This is executed by the candidate who has the decryption authorities. Decrypt (kS, S, i, C), the decryption algorithm takes input as public parameters pk, a ciphertext C, i denoting ciphertext classes for a set S of attributes.

Input = kS and the set S, where index i = ciphertext class

Outputs = m if i element of S

4. DATA SHARING

KAC in meant for the data sharing. The data owner can share the data in desired amount with confidentiality. KCA is easy and secure way to transfer the delegation authority. The aim of KCA is illustrated in Figure 2.

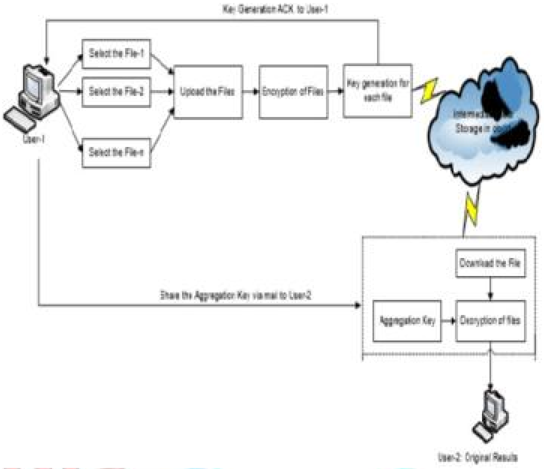

Fig 2. System Architecture.

For sharing selected data on the server user-1 first performs the Setup. Later the public/master key pair (pk, mk) is generated by executing the KeyGen. The msk master key is kept secret and the public key pk and param are made public. Anyone can encrypt the data m and this data is uploaded on server. With the decrypting authority the other users can access those data. If user-1 is wants to share a set S of her data with a user-2 then she can perform the aggregate key KS for user-2 by executing Extract (mk, S). As kS is a constant size key and the key can be shared through secure e-mail. When the aggregate key has got user-2 can download the data and access it.

4. RESULTS AND CONCLUSION

In the project, the results and discussions are used for comparing the existing system with the proposed model by using the performance analysis. In this below figure, the performance analysis with existing and proposed system by comparing with compression factor and delegation ratio yields the more efficient result than the previous methods. In this paper, we discussed the public-key encryption methodology for protecting the privacy of data from the attackers who may obtain the data by legal or other means, data stored by users and confidential information. The main aim of this approach is to obtain the aggregate key of constant size empowered with the decryption rights for the number of files is possible by the valid user. Along with the privacy of data, the confidentiality is also preserved by encrypting the user data before dumping into the cloud. Protection of the users' data privacy in cloud storage is an important aspect. With the help of mathematical tools, the encryption schemes are becoming more versatile and have started involving many encryption and decryption keys for a single application. But this project introduced the unique concept of the aggregation of the keys involved in decryption process. The cost of storing and transmitting the cipher texts is lowered as they are constant-sized. This compact aggregate key can be conveniently sent to others or be stored in a smart card with very limited secure storage. It is modelled in such a way keeping different security levels and extensions. Storing the delegated keys in the mobile devices which have no trusted software, there is a possibility that the key gets disclosed. So designing a leakage-proof cryptosystem which supports flexible and efficient key delegation is an interesting direction. In this project, the RSA and Blowfish algorithm is used for encrypting the files. A more secured and efficient algorithm can be used in future so as to cope up with the speed and for security purpose. The measures to avoid data de-duplication in the cloud can also be one of the enhancements for this project. Christo Ananth et al. [11] proposed a system in which the complex parallelism technique is used to involve the processing of Substitution Byte, Shift Row, Mix Column and Add Round Key. Using S- Box complex parallelism, the original text is converted into cipher text. From that, we have achieved a 96% energy efficiency in Complex Parallelism Encryption technique and recovering the delay 232 ns. The complex parallelism that merge with parallel mix column and the one task one processor techniques are used. In future, Complex Parallelism single loop technique is used for recovering the original message.

5. FUTURE ENHANCEMENT

The 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 article, 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 delegate 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. A limitation in our work is the predefined bound of the number of maximum ciphertext classes. In cloud storage, the number of ciphertexts usually grows rapidly. So we have to reserve enough ciphertext classes for the future extension. Otherwise, we need to expand the public-key as we described. Although the parameter can be downloaded with ciphertexts, it would be better if its size is independent of the maximum number of ciphertext classes. On the other hand, when one carries the delegated keys around in a mobile device without using special trusted hardware, the key is prompt to leakage, designing a leakage- resilient cryptosystem yet allows efficient and flexible key delegation is also an interesting direction

6. REFFERENCE

[1]Cheng-Kang Chu, Sherman S. M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng,(2014) Senior Member,IEEE. IEEE Transactions on Parallel and Distributed Systems. Volume: 25

[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 Dataon 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] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," ACM Trans. Information and System Security, Volume 9, Issue 1, pp. 1-30, 2006

[9] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," Theory and Applications of Cryptographic Techniques, Volume 3494, pp. 457-473, 2005.

[10] R. S. Sandhu, "Cryptographic Implementation of a Tree Hierarchy for Access Control," Information Processing Letters, Volume 27, Issue 2, pp. 95-98, 1988

[11] Christo Ananth, H.Anusuya Baby, "Encryption and Decryption in Complex Parallelism", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 3, Issue 3, March 2014,pp 790-795

[12] D. Boneh and M.K. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Advances in Cryptology, volume 2139, pp. 213-229, 2001.