ISSN: 2321-7782 (Online)
Impact Factor: 6.047

Volume 4, Issue 6, June 2016

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

A Review on Aggregate Key Cryptosystem for Data Sharing In Clouds Storage

Abhijeet C. Ghabade
Department of Computer Engineering,
Sinhgad Institute of Technology,
Lonavala – India

Abstract: Cloud storage can be a storage cloud that can be accessed on line connected to multiple information and resources. Smart cloud storage accessibility and reliableness, sturdy protection, disaster recovery, and will offer the lowest price. Cloud storage to safely, i.e. critical pragmatism fast, flexibly share information with others. Information sharing is a critical pragmatism in cloud storage. Throughout this paper we firmly, rapidly, a way to show and share with others information flexibly cloud storage. We measure cipher strength for a set of attainable texts square Cryptography rights of economical construction of new cipher strength constant size delegations texts describe public key cryptosystems. Innovation that will set any of a combination secret keys and, as a key Although the overall ability being as encompassing all of the keys of compact construction. Alternative words, key holder cipher strength determined in the versatile text selections are a constant cloud storage size will be unharnessed, but unlike the combination key encrypted files remain confidential out of the set. This compact combination keys are sent to others or handily often live much} end with secure storage very limited credit. We have within our customary models offer formal security analysis plans. We describe our plans of conjointly are optional application.

Keywords: Cloud storage, data sharing, key-aggregate encryption, patient-controlled encryption.

I. INTRODUCTION

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 drop box 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 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 drop box through the aggregate key. This is shown in figure1.

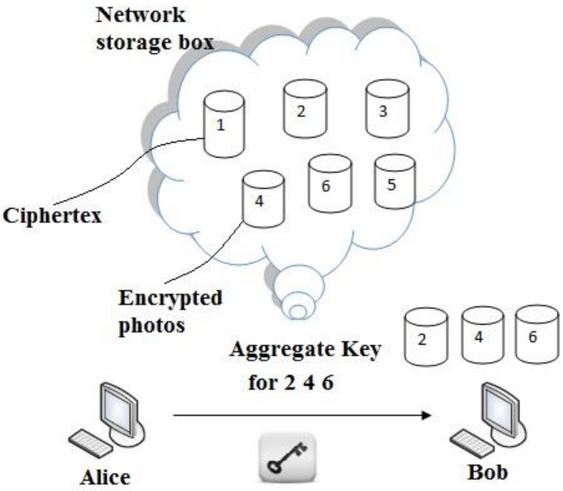

Fig 1. File Sharing Between Alice And Bob.

II. Literature Survey

Cloud 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 users 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 auditability, 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 uploader [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 are k attributes are overlay among the ciphertext and a private key the decryption is granted [5].

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].

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].

In a multi attribute-authorities numbers of attributes are analyzed regarding the decryption key and the user must get a particular key related to the attribute while decrypting a message. The decryption keys are allocated independently to users those who have attribute identity without interaction between each other. Multi-authority attribute-based encryption allows real time deployment of attribute based privileges as different attributes are issued by different authorities. The attribute authorities ensure the honesty of the user privilege so the confidentiality is maintained by central authority [8].

III. Proposed System Design and Performance Analysis

KEY-AGGREGATE ENCRYPTION:

A key aggregate encryption has five polynomial-time algorithms as

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.

2. KeyGen Phase

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

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.

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

Output = ciphertext C

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**.**

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.

Our approaches allow the compression factor F (F = n in our schemes) to be a tunable parameter, at the cost of O(n)-sized system parameter. Encryption can be done in constant time, while decryption can be done in O(jSj) group multiplications (or point addition on elliptic curves) with 2 pairing operations, where S is the set of ciphertext classes decryptable by the granted aggregate key and jSj n. As expected, key extraction requires O(jSj) group multiplications as well, which seems unavoidable.

However, as demonstrated by the experiment results, we do not need to set a very high n to have better compression than the tree-based approach.

Note that group multiplication is a very fast operation. Again, we confirm empirically that our analysis is true. We implemented the basic KAC system in C with the Pairing-Based Cryptography (PBC) Library8 version 0.4.18 for the underlying elliptic-curve group and pairing operations. Since the granted key can be as small as one G element, and the ciphertext only contains two G and one GT elements, we used (symmetric) pairings over Type-A (super singular) curves as defined in the PBC library which offers the highest efficiency among all types of curves, even though Type-A curves do not provide the shortest representation for group elements.

In our implementation, p is a 160-bit Solinas prime, which offers 1024-bit of discrete-logarithm security. Setup algorithm, while outputting (2n + 1) elements by doing (2n 2) exponentiations, can be made efficient by preprocessing function offered by PBC, which saves time for exponentiating the same element (g) in the long run. This is the only “low-level” optimization trick we have used. All other operations are implemented in a straightforward manner.

In particular, we did not exploit the fact that e^(g1; gn) will be exponentiated many times across different encryptions. However, we pre-computed its value in the setup stage, such that the encryption can be done without computing any pairing.

The execution times of Setup, KeyGen, Encrypt are independent of the delegation ratio r. In our experiments, KeyGen takes 3:3 milliseconds and Encrypt takes 6:8 milliseconds. As expected, the running time complexities of Extract and Decrypt increase linearly with the delegation ratio r (which determines the size of the delegated set S).

Our timing results also conform to what can be seen from the equation in Extract and Decrypt — two pairing operations take negligible time, the running time of Decrypt is roughly a double of Extract. Note that our experiments dealt with up to 65536 number of classes (which is also the compression factor), and should be large enough for fine-grained data sharing in most situations. Finally, we remark that for applications where the number of ciphertext classes is large but the non-confidential storage is limited, one should deploy our schemes using the Type-D pairing bundled with the PBC, which only requires 170-bit to represent an element in G. For n = 216, the system parameter requires approximately 2:6 megabytes, which is as large as a lower-quality MP3 file or a higher-resolution JPEG file that a typical cell phone can store more than a dozen of them. But we saved expensive secure storage without the hassle of managing a hierarchy of delegation classes.

IV. Conclusion

The data privacy may be a central question of cloud storage. With additional mathematical tools, crypto graphical schemes have gotten additional versatile and infrequently involve multiple keys for one application. During this paper, we have a tendency to take into account a way to “compress” secret keys in public-key cryptosystems that support delegation of secret keys for various cipher text categories in cloud storage. Regardless of that one in all the ability set of categories, the delegate will forever get associate degree combination key of constant size. Our approach is additional versatile than hierarchical key assignment which may solely save areas if all key-holders share the same set of privileges. Though the parameter is downloaded with cipher texts, it might be higher if its size is freelance of the utmost range of cipher text categories. On the opposite hand, once one carries the delegated keys around in a very mobile device while not victimization special trustworthy hardware, the secret\'s prompt to escape, coming up with a leakage-resilient cryptosystem , nonetheless permits economical and versatile key delegation is additionally a motivating direction. Outsourcing knowledge of knowledge of information} to server could cause leak the non-public data of user to everybody. Cryptography may be a one resolution that provides to share elect knowledge with desired candidate. Sharing of decipherment keys in secure method plays vital role. Public-key cryptosystems provides delegation of secret keys for various cipher text categories in cloud storage. The delegate gets firmly associate degree combination key of constant size.

ACKNOWLEDGEMENT

We would like to express our sincere thanks to our beloved Principle Dr. M.S. Gaikwad and project co_ordinator Prof. M.S. Chaudhari for facilities provided for preparing this project

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, "Proc. 10th Int'l Conf. Applied to Cryptography and Network Security (ACNS),vol. 7341, pp. 526-543, 2012.
2. B. Wang, S.S.M. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," Proc. IEEE 33rd Int'l Conf. Distributed Computing Systems (ICDCS),2013
3. S.S.M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R.H. Deng, "Dynamic Secure Cloud Storage with Provenance,"Cryptography and Security,pp. 442-464, Springer, 2012.
4. J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," Proc. ACM Workshop Cloud Computing Security(CCSW'09),pp.103-114,2009.

AUTHOR(S) PROFILE

Abhijeet Ghabade, received the B.E (Bachelor of Engineering) degree in Information Technology from Rajarshi Shahu College of Engg.(pune) in 2010. During 2012-2016 He research with sinhgad institute of technology Department of computer Engineering, Lonavala, India to study cloud computing and network security.