INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

EXPRESSIVE, EFFICIENT DATA SHARING IN CLOUD STORAGE

MISS G. V. KAPSE, PROF. S. S. SHEREKAR, DR. V. M. THAKARE

SGBAU, Amravati, India.

Accepted Date: 15/03/2016; Published Date: 01/05/2016

Abstract*:* To avoid unauthorized access in cloud data sharing and storage, data should be encrypted before outsourcing. And for that attribute based encryption mostly used, instead of attribute based encryption here role based policies can be generated and depend on that policies encryption can be done. In case of data with multiple owners, all the owners must have identical access policy and revocation must be done with the permission of all owners. Other major issue is key generation and management. To overcome key generation and management issue explore a policy based encryption technique in which access key generation for user is depended on access policies assigned to every user together with the attributes. The data stored in the cloud is encrypted using a key generation technique and based on the access permissions assigned to the data and attributes of the owners those share their data with security and integrity by the use of policy based encryption technique. This new mechanism is applicable in many real-world applications is presented in this paper.

Keywords: Cloud computing, data sharing, data storage, multi-owner, data encryption.

Corresponding Author: MISS G. V. KAPSE

Access Online On:

www.ijpret.com

How to Cite This Article:

G. V. Kapse, IJPRET, 2016; Volume 4 (9): 634-646

PAPER-QR CODE

INTRODUCTION

Cloud computing has been drastically change the shape of modern computing environment. The growth of electronic personal data leads to a trend that data owners choose to remotely outsource their data to clouds and enjoy the high-quality retrieval and storage service without worrying about the burden of local data management and maintenance. However, secure share and search for the outsourced data is a difficult to defeat, which may easily occurs to leakage of sensitive personal information. Expressive data sharing and storing with security is importance. Cloud storage has emerged as a challenging solution for providing ubiquitous, convenient, and nowadays, it is easy to apply for free accounts for email, photo album, and file sharing and remote access, with storage size of more than about 25 GB. Along with the current wireless technology, users can able to access all of their files and emails by a mobile phone in any corner of the world. Business users are attracted by cloud storage due to its various benefits, including lower cost, greater agility, and better resource utilization. In enterprise settings, the rise in demand for data out sourcing, this is assisted in the strategic management of corporate data. It is also used as a core technology in the background of various online services for personal applications. Though enjoying the benefits of sharing data via cloud storage, users are also increasingly worried about inadvertent data leaks in the cloud. Such data leaks, caused by a hacker or a misbehaving cloud operator, and can caused to seriously to personal privacy or business secrets. To address users worry over potential data leaks in cloud storage, so the data owner prefer to encrypt all the data before uploading them to the cloud, such that whenever required the encrypted data may be retrieved and decrypted by those who have the decryption keys. However, the encryption of data makes it difficult for users to search and then selectively retrieve only the data containing given keywords [4]. A common solution is encryption in which Multi-owner information exchange is a model for sharing business data of large organizations, which allows owners to create, manage and control data in cloud environment.

Cloud storage permits many users having different roles and access permissions to share and store their data. In policy based data sharing, every user has an access policy for the system and each file has definite file access policy which can differ for different users. In a distributed environment multiple replicas of data is stored to improve availability. Hence the integrity and access control are major issues. A cloud system may have different cloud service providers (CSPs) to improve the performance of the system. Based on availability and work load, the system selects a CSP for the client accessing it. Number of clients may access the system simultaneously; hence the availability of data is a major issue. It can be improved by CSPs with data duplication. The data owners may want to set some restrictions to clients who are trying to access the data. In this scenario, the distributed data should keep all details about the different access control policies set to data. But again the authorized clients should be categorized according to permission; it may be an issue in a distributed system with number of clients.

This system mainly allows public users and custom users. In the case of public users the policy is same for everyone and only public files will be accessed by them, it is not a challenging problem. The second category i.e. the custom users are somewhat a major problem. Custom users are selected users who have special permissions for accessing some files or data. The permissions may be same or different for each custom user. The system must keep track of all the access policies of custom users and provide data or file according to those policies. Here proposing a system that deals multiple owners and policies to provide secured data sharing in cloud storage.

II) BACKGROUND

A searchable attribute based proxy re-encryption system when compared with the existing systems only supporting either searchable attribute-based functionality or attribute-based proxy re-encryption; this primitive supports both abilities and provides flexible keyword update service. In particular, the system enables a data owner to efficiently share his data to a specified group of users matching a sharing policy and meanwhile, the data will maintain its searchable property but also the corresponding search keyword can be updated after the data sharing. It is also proved chosen cipher-text secure in the random oracle model [1]. A novel public verification mechanism was to audit the integrity of multi-owner data in an un-trusted cloud by taking the advantage of multi-signatures. With this mechanism, the verification time and storage overhead of signatures on multi-owner data in the cloud are independent with the number of owners. In addition, it demonstrates the security of scheme with rigorous proofs, compared to the straightforward extension of previous mechanisms [2]. Key-Aggregate cryptosystem for scalable data sharing in cloud storage described public-key cryptosystems that produce constant-size cipher texts such that efficient delegation of decryption rights for any set of cipher texts was possible. The speciality was 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. Key-aggregate encryption scheme consists of five polynomial-time algorithms. It considered how to “compress” secret keys in public-key cryptosystems which support delegation of secret keys for different cipher-text 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.

This scheme gave the first public-key patient-controlled encryption for flexible hierarchy [3]. The design of KASE scheme draws its insights from both the multi-key searchable encryption scheme and the key-aggregate data sharing scheme. Specifically, in order to create an aggregate searchable encryption key instead of many independent keys, it adapted the idea presented in. Each searchable encryption key was associated with a particular index of document, and the aggregate key is created by embedding the owner's master-secret key into the product of public keys associated with the documents. In order to implement keyword search over different documents using the aggregate trapdoor, author employed a similar process as in. The cloud server can used this process to produce an adjusted trapdoor for every document [4]. Proxy re-encryption scheme worked more efficiently for the group data sharing in cloud environment especially when the size of the group was large and the membership of the group was dynamic compared with the existing CPRE schemes, in which whenever the membership of the group was changed, the condition value should be changed and the re-encryption key should be generated for each data shared by the updated group [5].

This paper introduces Multi-owner data sharing in cloud storage using policy based encryption and these are organizes as follows. **Section I** Introduction. **Section II** discusses Background. **Section III** discusses previous work. **Section IV** discusses existing methodologies. **Section V** discusses attributes and parameters and how these are affected. **Section VI** proposed method and outcome result possible. Finally **section VII** Conclude this paper.

III) PREVIOUS WORK DONE

Kaitai Liang, et al. [1] introduced a novel and practical notion, searchable ABPRE. This notion guarantees that the keyword search ability of a cipher-text can be remained after the sharing of the cipher-text. Liang designed a concrete searchable Key-Policy (KP) ABPRE system satisfying the above notion. Prove the scheme chosen ciphertext secure in the Random Oracle Model (ROM). As of independent interest, author's protocol supports keyword update. This property brings a convenience to data owner in the sense that the ciphertext keyword can be freely modified based on data shared record. This system had better efficiency regarding to keyword search and decryption phases.

Wang, et al. [2] proposed a novel multi-signature scheme with the property of block less verifiability, using this scheme as a building block, designed an efficient public verification mechanism on the integrity of multi-owner data in an un-trusted cloud. With this mechanism, each block in cloud data was signed by multiple owners and is attached with one multi-signature so that the size of verification metadata on each block was constant and independent with the number of owners. The entire data was valid only if all the owners signed it correctly. In addition, a public verifier was able to audit the integrity of multi-owner data with only a combination of all the blocks instead of downloading the entire data from the cloud, and the verification time was also irrelevant with the number of owners. Cheng-Kang Chu, et al. [3] proposed several concrete KAC schemes with different security levels and extensions and also studied how to make a decryption key more powerful in the sense that it allows decryption of multiple cipher-texts, without increasing its size and gave the framework and definition for key aggregate encryption. Then it described how to use KAC in a scenario of its application in cloud storage. Baojiang Cui, et al. [4] defined a generic framework for key aggregate Searchable encryption (KASE) and provides requirements for designing a valid KASE scheme. The design of KASE scheme draws its insights from both the multi-key searchable encryption scheme and the key-aggregate data sharing scheme. Specifically, in order to create an aggregate searchable encryption key instead of many independent keys. Each searchable encryption key was associated with a particular index of document, and the aggregate key was created by embedding the owner's master-secret key into the product of public keys associated with the documents. In order to implement keyword search over different documents using the aggregate trapdoor, it employed a similar process as in. The cloud server can used this process to produce an adjusted trapdoor for every document. **T**he idea of proxy re-encryption has been introduced to support secure data sharing among group members in cloud environment. However, in this scheme, a malicious user can collude with the server to decrypt unauthorized messages. Junggab Son, et al. [5] proposed the conditional proxy re-encryption (CPRE) aimed to fix this problem by introducing a condition value into message encryption process and re-encryption key generation. CPRE significantly inefficient when the membership of the group changes very actively and the size of the group large since a new condition value selected and re-encryption keys had to be generated for each user whenever the group membership is changed. In CPRE the condition value not associated with re-encryption keys. Whenever a group membership was changed, only a new condition value distributed to the users via cloud server. As a result, the overhead of each user becomes significantly reduced at each membership change.

IV) EXISTING METHODOLOGIES

Author can construct a straightforward solution to verify the integrity of multi-owner data in the cloud by asking each owner to independently compute a signature on each block in multi-owner data with previous method and storing all the *d* signatures of each block on the cloud. Where *d* is the number of owners. However, this straightforward solution is not efficient, because not only the verification time but also the storage overhead of signatures on the entire data is linearly increasing with an increase of the number of owners.

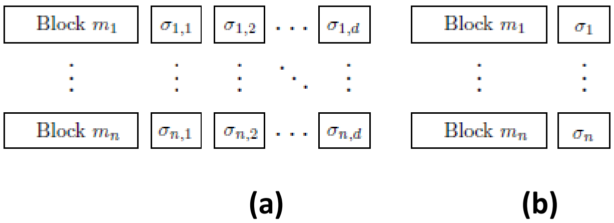

Fig.1. Comparison between the straightforward solution and this mechanism: (a) The straightforward solution, where d signatures are attached to each block and (b) this mechanism, where one multi-signature is attached to each block [2].

A canonical application of KAC is data sharing. The key aggregation property is especially useful when expected 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. This described the main idea of data sharing in cloud storage using KAC, illustrated in Fig.2. Suppose Alice wants to share her data $m_1, m_2, ..., m_v$ on the server. She first performs Setup (1^ $^{(\lambda)}$,n) to get param and execute **KeyGen** to get the public/master-secret key pair (pk, msk). 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 m_i by C_i= Encrypt (pk, *i*, m_i). The encrypted data are uploaded to the server. With param and pk, people who cooperate with Alice can update Alice's data on the server. Once Alice is willing to share a set *S* of her data with a friend Bob, she can compute the aggregate key K_S for Bob by performing Extract (msk, *S*). Since K_S is just a constant-size key, it is easy to be sent to Bob via a secure e-mail. After obtaining the aggregate key, Bob can download the data he is authorized to access. That is, for each *i* € *S*, Bob downloads C_i (and some needed values in param) from the server. With the aggregate key K_S, Bob can decrypt each C_i by Decrypt (K_S, *S*, *i*, C_i) for each i € *S*.

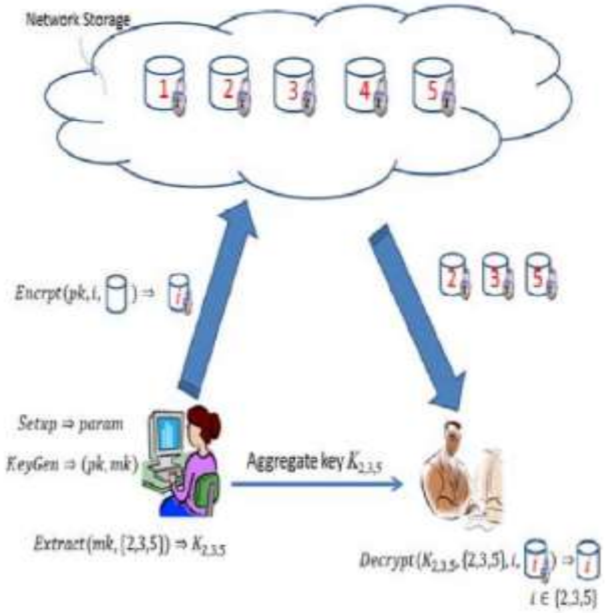
Network Storage
1
2
3
4
5
Encrpt(pk, i,) ⇒ i
2
3
5
Setup ⇒ param
KeyGen ⇒ (pk, mk)
Aggregate key $K_{2,3,5}$
Extract(mk, {2,3,5}) ⇒ $K_{2,3,5}$
Decrypt($K_{2,3,5}$, {2,3,5}, i, i) ⇒ i
i ∈ {2,3,5}

Fig.2. Using KAC for data sharing in cloud storage [3].

The KASE framework is composed of seven algorithms. Specifically, to set up the scheme, the cloud server would generate public parameters of the system through the **Setup** algorithm, and these public parameters can be reused by different data owners to share their files. For each data owner, he/she should produce a public/master-secret key pair through the **Keygen** algorithm. Keywords of each document can be encrypted via the **Encrypt** algorithm with the unique searchable encryption key. Then, the data owner can use the master-secret key to generate an aggregate searchable encryption key for a group of selected documents via the **Extract** algorithm. The aggregate key can be distributed securely to authorized users who need to access those documents. After that, as shown in fig. 3, an authorized user can produce a keyword trapdoor via the **Trapdoor** algorithm using this aggregate key, and submit the trapdoor to the cloud. After receiving the trapdoor, to perform the keyword search over the specified set of documents, the cloud server will run the **Adjust** algorithm to generate the right trapdoor for each document, and then run the **Test** algorithm to test whether the document contains the keyword.

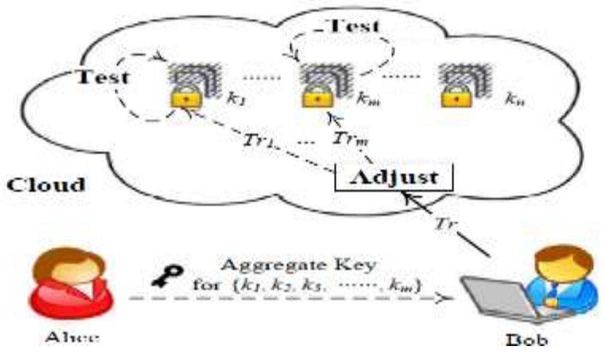

Fig.3. Framework of key-aggregate searchable encryption [4].

V) ANALYSIS AND DISCUSSION

Liang defined a searchable attribute-based proxy re-encryption with keyword update, and proposed a concrete construction satisfying the notion. And proved the scheme CCA secure in the ROM. The scheme is the first of its type to integrate searchable attribute-based encryption with attribute-based proxy re-encryption, which was applicable to many real-world applications [1]. In efficient public verification on the integrity of multi-owner data in the cloud, with the techniques of multi-signatures, only a multi-signature is required to be attached to each block for multiple owners. As a result, not only signature storage but also verification time are both independent with the number of owners in mechanism. A public verifier is able to efficiently check the correctness of multi-owner data. Security analyses demonstrate that mechanism is secure, and experimental results show this mechanism has a better performance on verifying multi-owner data than the straightforward solution extended by previous work [2]. With more mathematical tools, cryptographic schemes are getting more versatile and often involve multiple keys for a single application. This scheme considered 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. This scheme gave the first public-key patient-controlled encryption for flexible hierarchy [3]. Analysis and evaluation confirm that KASE scheme can provide an effective solution to building practical data sharing system based on public cloud storage. In a KASE, the owner only needs to distribute a single key to a user when sharing lots of documents with the user and the user only needs to submit a single trapdoor when he queries over all documents shared by the same owner. However, if a user wants to query over documents shared by multiple owners, he must generate multiple trapdoors to the cloud [4]. A content sharing scheme that is safe in the cloud computing environment, based on a conditional proxy re-encryption scheme. The scheme proposed can significantly reduce burden of a client due to two characteristics. First, re-encryption process is delegated to a cloud server and second, the number of re-encryption keys to be required for sharing is minimized [5].

Data sharing and Storage techniques	Advantages	Disadvantages
Searchable Attribute-Based Mechanism	1)Provides flexible keyword update service 2) Reduce the size of search token	1)Additional encryption/ decryption burden on user 2) Lack of privacy and confidentiality
Efficient Public Verification on the Integrity of Multi-Owner Data	1) The storage cost of signatures and verification time of multi-owner data are independent with the number of owners. 2) A public verifier is able to check the correctness of data stored on the cloud server without retrieving all the blocks.	It is computationally impractical to generate a forgery of a multi-signature.
Key-Aggregate Cryptosystem	Public-key of a user can be set as identity string of the user like email address	The predefined bound of the number of maximum ciphertext classes.
Key-Aggregate Searchable Encryption (KASE)	Employing caching techniques in the group data sharing system to further improve the efficiency of the keyword search procedure.	KASE cannot be applied in Federated clouds directly
Secure Data Sharing in Cloud Environment	1) Whenever a group membership is changed, only a new condition value is distributed to the users via cloud server. 2)Client efficiency 3)Managing	1) A cloud cannot obtain plaintext by carrying out multiple re-encryptions. 2) A client can unable to recover the plaintext with

	group efficiency	chosen cipher text attack and without condition values.

Table 1: Comparison between various data sharing and data storage techniques in cloud computing

VI) PROPOSED METHODOLOGY

Here proposing the architecture with multiple owners and multiple users. The owners of a unique data or file belong to an organization or institution. For example, in an organisation the confidential data may be handled by only directors of company and possibility of having multiple directors. In such case the security and integrity of data is major issue. In this proposed method the data may have multiple owners, the owners register into system as a group but having unique access keys and passwords. Any member belongs to the group can store and share the data. The policies of shared files are set by any owners from group and should need approval of all the owners. That means, any change in file policy should be done by the group permission. The file shared can be private, public or custom based on the set access policy.

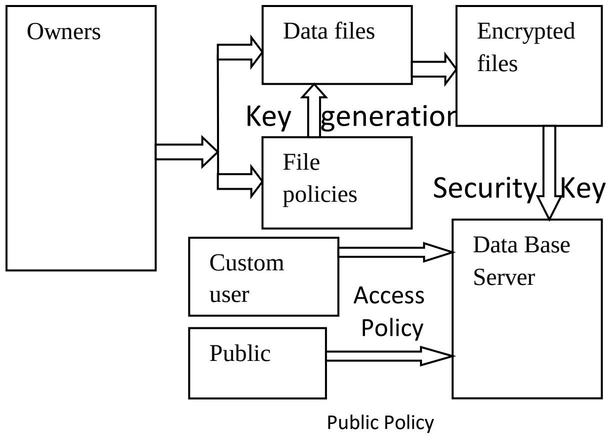

Fig. 4: Architecture

A. Access Key Generation

Access key is generated for every user who registers in the system. System collects required attributes from user like e-mail, user-name etc. Using these attributes from user and some other features a unique key is generated and from that key, using a pattern function, a six digit code is generated and that will be the passed to the user. At the time of login, system verifies this code along with the user-name and password. As the system is mainly designed for a specific organization, the users of the system can be grouped according to the terms of organization.

1) ***Owners:*** Owners are group member who upload data to cloud for secure storage. Each owner member in a group have own private login including user-name, password and an access key together with a group key. The group key is common for all owners in a particular group.

2) ***Public Users:*** Any user can be a public user and the defined procedure is same for all users. No specific policy is set to access the public files, only the file policy is required.

3) ***Custom Users*****:** In an organization, different types of clients and many kinds of employees are accessing the information. So the access permissions should vary and owners can select a category or particular user to assign a particular access policy. Specific access keys are generated depend upon file policy and user type. That particular access key and policy is checked before giving access to those files.

B. File Policy and Encryption Key

File policies are generated for every file based on the confidentiality of the file. The member of group may store the file as private, public or custom and can set the permissions as read, edit, download and delete.

The first category is belongs to the confidentiality and type of information. For example, some data is strictly very confidential and only owners are able to view. Some data can be edited by any owner and some data can be edited with the permission of all owners in group. Same way the data for clients can be accessed by them and managed by particular employees.

The second category in policy is the access rights or permissions for users. Here user means all those who access the system. The permissions may read, edit, download or delete. By combining these two factors file policies are generated. For example client P can read and download File Q as a custom user or any user can read File R as public user.

The big challenge is generation of encryption key depends on these policies. For generating the encryption key, the file access policy like private, public and custom is used additionally file attributes, attributes of owners and access permissions. Therefore the encryption method is a mixture of Attribute based Encryption and Policy based Encryption. A new pattern function is used to make key, the pattern differs for private, public and custom files. The encryption key breaks and forming ‘n’ codes and saves those in ‘n’ key managers. The combination code is stored in another location.

C. Decryption

Before accessing any file, the file policies and user policies must match. If both match, then according to the access key of user the system give the permissions and allowed that user to retrieves the combination code. Using the combination code, the key codes are retrieved and combined to make key, in this way decryption is done with that key.

D. File Revocation

File revocation is the making of the file permanently unavailable. This can be done by deleting the file policies and encryption keys. If ‘m’ codes out of ‘n’ are deleted the key cannot be restored and decryption may impossible. When a file is trying to access, firstly the file policies should checked, if there are no file policy then the file itself inaccessible. The system two times ensures the inaccessibility of a file.

OUTCOME AND POSSIBLE RESUL

Design a secure data sharing with policies for dynamic groups of owners in a cloud storage system. In which, a user is able to share data with others in the system as well as a group and also capable of storing and sharing their data. Also this system supports efficient file revocation and policy changing. More specially, efficient file revocation can be done through file policy revocation and deletion of some key codes. So users cannot decrypt files stored in the cloud. Furthermore, the storage overhead and the encryption computation cost are constant.

VII) CONCLUSION

In consideration of the practical problem of privacy-preserving in data sharing system based on public cloud storage which requires a data owner to distribute a hundred or thousands of keys to users to enable them to access his/her documents, here propose the concept of policy based encryption, in which access key generation for user is based on access policies assigned to each user along with the attributes having high security. Experimental results confirm that this work can provide a promising solution to building practical data sharing system based on public cloud storage.

FUTURE SCOPE:

One of promising future work is to introduce the efficient encryption scheme on top of this policy based encryption and consider the problem of security, computational overhead as well. Supporting user revocation is an important issue in the real application, and that is a big challenge in the application of Policy based encryption schemes. Making this scheme compatible with existing schemes that support efficient user abrogation is one of future works.

REFERENCES

1. Kaitai Liang and Willy Susilo,"Searchable Attribute-Based Mechanism with Efficient Data Sharing for Secure Cloud Storage", IEEE transactions on information forensics and security, VOL. 10, NO. 09, September 2015.
2. Boyang Wang, Hui Li, Xuefeng Liu, Fenghua Li, and Xiaoqing Li, "Efficient Public Verification on the Integrity of Multi-Owner Data in the Cloud", Journal of communications and networks, VOL. 16, NO. 06, December 2014.
3. 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 systems, VOL. 25, NO. 02, February 2014.
4. Baojiang Cui, Zheli Liu and Lingyu Wang, "Key-Aggregate Searchable Encryption (KASE) for Group Data Sharing via Cloud Storage", IEEE Transactions computers, VOL. 06, NO. 02, June 2014.
5. Junggab Son, Hyunbum Kim, Donghyun Kim,"On Secure Data Sharing in Cloud Environment", ACM, VOL. 09, NO. 11, January 2014.