Privacy Preserving Access Control Policies Using Two Level Encryption In Public Cloud

A. Karuna
Asst. Prof, *Dept. Of CSE*
University College of Engineering JNTUK, Kakinada
karunagouthana@gmail.com

T. Venkata Satish
M. Tech. Dept. Of CSE
University College of Engineering JNTUK, Kakinada
sateeshnava@gmail.com

***Abstract*—** With many features of cloud computing, many organizations have been considering moving their information systems to the cloud storage. Cloud storage is a service model, in which data is stored, maintained, managed and backup remotely, And made available to the users over the network. However, an important problem in public cloud is preserving confidentiality of stored data from both unauthorized access and the storage provider i.e. cloud. So, in order to make confidentiality the data owners first encrypts the file before uploading them on the cloud storage, and re-encrypting whenever user credentials change .That thus incur high communication and computational cost. We propose an approach based on two level encryption. Under such approach the data owner performs coarse grained encryption before uploading to the cloud server, and then cloud performs complete access control policy encryption on top of the encrypted data by the owner. Our system handles both confidentiality of data and preserves privacy of users from unsecured cloud.

Keywords- *cloud storage, Privacy ,Access conrol,Two level Encryption.*

*****

I. INTRODUCTION

With the advent technologies of cloud computing, data is stored in cloud. However privacy and security represent major concerns for assuring the confidentiality of data against the cloud. Encryption is a commonly adopted technology to ensure the confidentiality of data. Encryption alone however is not sufficient to support the enforcement of fine-grained organizational access control policies (ACP). Nowadays many organizations regulating ACP's means which users can access which data or records, Such control is often based on security-relevant properties of users, called as identity attributes, using access control language. These access control systems are referred to as "attribute based access control" (ABAC) systems. Support fine-grained access control which is essential for high-assurance data confidentiality and security.

The figure 1: shows the traditional encryption approaches have been proposed for fine-grained access control over the encrypted data [2], [3]. , these approaches combined data items or records based on their Access control policy, and using different symmetric key each group items or records are encrypted. After that, the keys for the encrypted data items are given only to the users that they are allowed to access. However, such approaches have various limitations.

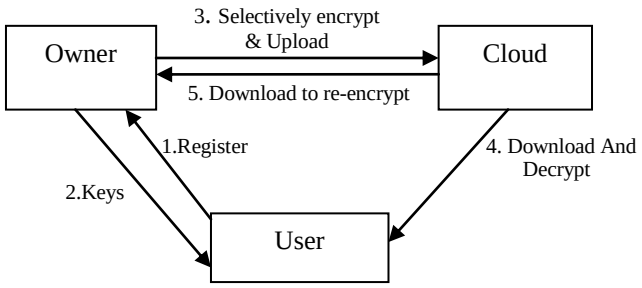

Fig. 1: Traditional Approach

1. Whenever user dynamic changes, the data owner tends to download and decrypt the file.
2. Re-encrypt it with the new keys of updated data, and upload the encrypted data.
3. The owner needs to establish a private communication channel for issuing the new keys to the users.
4. User identity attributes confidentiality is not considered. SO cloud can learn user private information and their organization.
5. They are in efficient in supporting fine grained attribute based access control (ABAC).

The goal of this article is to provide an overview of our proposal to enforce fine-grained access control on sensitive data stored in untrusted public clouds, while at the same time it assures the confidentiality of the owner data from the public cloud and preserving the privacy of users who are authorized to access the data.

The rest of the article is organizes as follows Section II describes drawbacks of existing cryptograph technique and propose a new approach to address limitation of existing system and Section III overall view and architecture of Two Layer Encryption approach.

II. SINGLE LAYER ENCRYPTION

In Recent time proposed approaches based on broadcast key management scheme address the some of the limitation of the traditional encryption. We refer to these approaches as a single layer encryption (SLE) approaches. SLE assures the privacy of the users and support fine grained ACPs.

However, while SLE addresses some limitation of traditional approaches, it still requires the data owner needs to enforce all the ACPs through selective encryption and upload encrypted file to the public cloud. The data owner first encrypts the file before uploading it to the cloud, and re-encrypts the file whenever user dynamic changes. So, the data owner has to download all affected data that thus incurs high communication and computation cost.

For *e.g.*, whenever user credentials are changed in the system, the data owner tends to download the affected data from the cloud, generate new encryption key, and re-encrypt the downloaded data with new key, and then uploads the re-encrypted file to the cloud.

The Fig :2, shows the conventional data outsourcing scenario of SLE approach where the data owner enforces all the ACPs through selective encryption and uploads encrypted file to the public cloud.

- The System consists five different phases

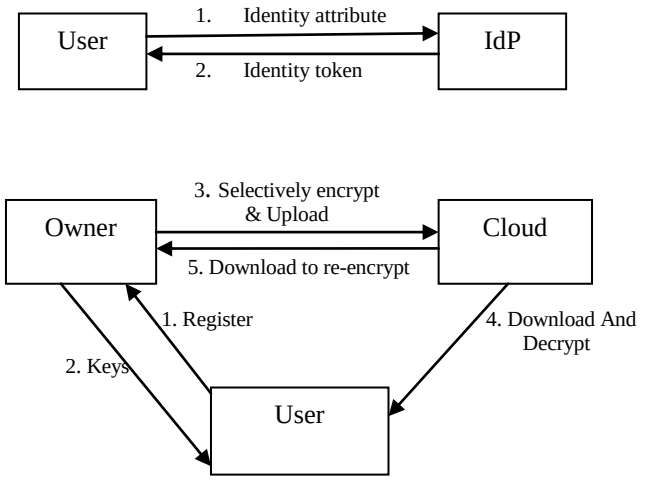

Fig. 2: Single Layer Encryption approach

In this paper, we propose a novel approach to address these limitations. The approach is based on Two Layer of encryption called as *two-level encryption* (TLE). Under such an approach the data Owner performs a coarse grained encryption on the data in order to assure the confidentiality of the data from the un-trusted cloud, whereas the Cloud performs a fine-grained encryption on top of the encrypted data performed by the data owner, encryption based on the ACPs. A challenging issue in this approach is how to decompose the ACP. So, that fine-grained ACP enforcement can be given to the cloud while at the same time assures the privacy of the identity attributes of users and confidentiality of the data. In order to delegate as much as possible access control policy enforcement to the Cloud, one needs to decompose the ACP. So, that the Owner only needs to manage the minimum number of attribute conditions in these policies that assure the confidentiality of data from the un-trusted Cloud. Each policy should be decomposed into two sub ACPs such that the combination of the two sub policies result in the original policy (ACP). The two-level encryption should be performed such that the owner first encrypts the file based on the one set of sub policies and then Cloud re-encrypts the already encrypted data using the other set of policies (ACP). The two encryptions together enforce the original policies as users should perform two decryptions in order to access the data.

For consider an example, if the ACP is (“role=doctor” ∧ “yos>2”) ∨ (“role=doctor” ∧ “type=junior), the ACP can be decomposed into two sub ACPs “role=doctor” and “type=junior” ∨ yos>2”.Here yos stands for year of service, and type(junior, assistant and senior). Notice that the decomposition is consistent; that is, (“role=doctor” ∧ “yos>2”) ∨ (“role=doctor” ∧ “type=junior”) = “role=doctor” ∧ (“type=junior” ∨ “ÿos>2”). The data owner enforces using the earlier one by encrypting the data for the users satisfying the former and the cloud enforces the latter one by re-encrypting the data already encrypted by the owner for the users satisfying the latter. Since the cloud does not handle Condition 1, it cannot decrypt owner encrypted file/data. So the confidentiality of owner data is preserved. Notice that users must satisfy the original ACP in order to access the file by performing two decryptions. Since the data owner handles minimum number of attributes the overhead of managing attributes at the data owner is reduced.

III. TWO LEVEL ENCRYPTION

In this section the proposed system is briefly described. Two level Encryption (TLE) approach. By name itself says there are two ways of encryption schemes. Like the SLE system described in Section II, the TLE system consists of the four entities, Data Owner, User, IdP and Cloud. However, unlike the SLE approach, the Cloud and the data owner collectively enforce ACPs by performing two encryptions on each data item. Under this approach one to reduce the burden on owner side and delegates as much as access control policies to the cloud. It provides a better way to handle user dynamic changes.

- Owner, the data owner define ACP and decompose into two sub ACPs, and uploads encrypted data to the cloud.
- The Cloud re-encrypts the encrypted data of owner using the second ACP, and stores the data.
- Idp, the Identity provider module maintain the meta data of encrypted data with details like file name, domain name, secret key along with user privileges. And IdP also deals with creating new data owner and users.
- User, The user uses one more identity tokens to gain access to the encrypted data hosted in the cloud.

Fig :3 shows user registration, the user registers with his/her user name along with domain name, and email address. After that IdP provide file request permission to user from data owner for downloading files.

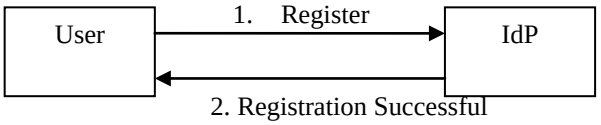

Fig. 3: User Registration

Fig.:4 shows Data Owner Registration. The data owner registers with attributes like username, email address and file permission controls.

Fig. 4: Data Owner Registration

The TLE approach has many advantages, whenever the user policy changes, only the top layer of the encrypted file needs to be updated. Since the top layer encryption is performed at the cloud side, no data transmission is required between the Owner and the cloud. And no need to establish a private communication channel with users for issuing the new keys. This two level encryption allows one to reduce the burden on the data Owner side, and delegates as much as possible access control policies to the Cloud.

Fig 5: Shows the Actual system diagram of two layer encryption approach.

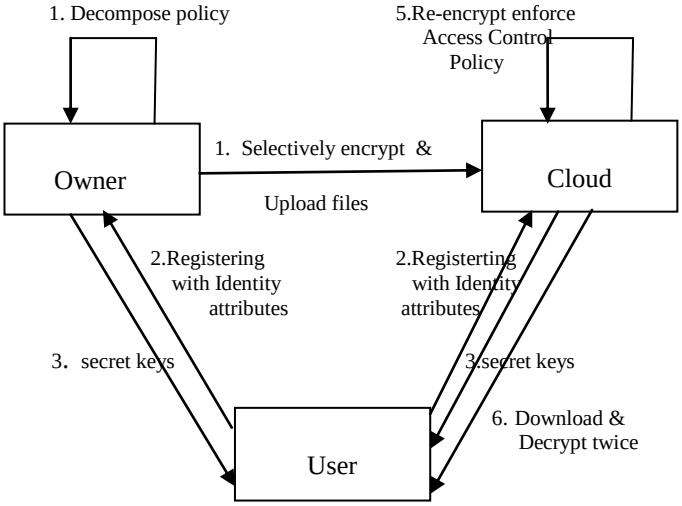

Fig. 5: Two Level Encryption

The Two Level Encryption (TLE) approach has consisted of following phases.

Policy Decomposition: The data owners first decompose each access control policy into two sub ACPs such that the data owner enforces minimum number of attribute conditions to assure the confidentiality from the cloud. And make sure that the decomposed ACPs are consistent, so that the two sub ACPs together enforce the original ACPs as described in section II.

User Registration: Users register/request with their identity attributes in order to obtain secrets to decrypt file that they are allowed to access from both cloud and owner.

File Encryption and Uploading: The data Owner first encrypts the file using Owner's sub ACPs in order to hide the data from the Cloud. And then uploads the file along with public information like file name, domain name, sub domain cloud server address. Here we use AES Algorithm for encrypting the file, and KeyGen algorithm is used for key distribution. Then the cloud re-encrypts the data based on the key generation.

File Downloading and Decryption: User module can download the encrypted data by providing file name, user name and his/her attributes at the time of registration from the cloud. Here user decrypt the file twice, first removing the encryption layer added by the cloud by providing attributes, It decrypts the access control encryption. And then decrypts the file by using AES algorithm i.e. inner layer encryption performed the data owner.

Encryption Evolution Management: After the data owner performs initial encryption to the data, and upload to the cloud. whenever user dynamics changed affected data need to be re-encrypted with a new key .Unlike SLE approach when user credentials are changed or ACPs modified , the data owner no need to involve ,the cloud re-enforce the affected data with revoked users.

IV. Experimental Result

The experiment performed on a windows7 with a dual core processor 5400 @ 2.70GHz, 1 GB of RAM. Our proposed prototype system is implemented in Java. In this we use AES-256 implementation for encryption. And KeyGen algorithm is used for key distribution to the user .here we use 16 bit key .first 10 bits for public key and remaining six bits for private key.

V. Coclusion

Privacy and Security is a key issue for cloud storage. An effective way for protecting privacy is to set the privacy policies that are the agreement between data owner and the storage provider. In this paper, we propose a Two level Access control scheme to enable data sharing in cloud environment for to achieve both data confidentiality and privacy protection. So, the proposed system provides the confidentiality of user data from the cloud storage. As future we would like to further extend our scheme with new features and a complete solution that guarantees both security and privacy protection for a remote file storage system. And we also plan further try to reduce the computation cost.

References

[1] Mohamed Nabeel and Elisa Bertino ," Privacy Preserving Delegated Access control In Public Clouds" In IEEE Transaction On Knowledge And Data Engineering, 2013. *(references)*

[2] E. Bertino and E. Ferrari " Secure and selective disseminationof XML documents," ACM Trans. Inf .Syst. Secr.,Vol 5, no. 3.

[3] G. Mikalu and Suciu, "Control access to published data using cryptography ,"

[4] M. Nabeel, E. Bertino, M. Kantarcioglu, and B. M. Thuraisingham, "Towards privacy preserving access control in the cloud," in Proceedings of the 7th International Conference on Collaborative

Computing: Networking, Applications and Worksharing, ser. CollaborateCom '11, 2011, pp. 172–180.

[5] N. Shang, M. Nabeel, F. Paci and E. Bertino, "A privacy preserving approach to policy-based content dissemination," in ICDE '10: Proceedings of the 2010 IEEE 26th International Conference on Data Engineering, 2010.

[6] M. Nabeel and E. Bertino, "Towards attribute based group key management," in Proceedings of the 18th ACM conference on Computer and communications security, Chicago, Illinois, USA, 2011

[7] J.BetheJ. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy ttribute-based encryption. In *SP 2007: Proceedings of the 28th IEEE Symposium on Security and Privacy*, pages 321–334, 2007.

[8] Nabeel, N. Shang, and E. Bertino. Privacy preserving policy based content sharing in public clouds.*IEEE Transactions on Knowledge and Data Engineering*, 99, 2012.