International Journal of Advance Research in Science and Engineering
Vol. No.5, Issue No. 02, February 2016
www.ijarse.com
IJARSE
ISSN 2319 - 8354

A SURVEY ON PRESERVING SECURITY OF USERS WITH AUTHORIZED ACCESS CONTROL IN PUBLIC CLOUD

Minaj Mulla[1], Aliya Inamdar[2],

[1,2]Dept. of Computer Science Engineering, S.I.E.T College of Engineering and Technology, Vijaypur, Karnataka, (India)

ABSTRACT

Security and privacy represent major concerns in the cloud, the conventional approaches for data encryption to enforce fine grained access control is insufficient for the confidential data hosted in the cloud.it uses fine-grained encryption to encrypt the data, in this approach the data owner itself need to encrypt the data and upload it to the cloud and if users changes their credentials then again re-encrypting the data. Thus the data owner incurs high communication and computational cost. A better should minimize the overhead at the data owner. In the proposed paper the overhead is reduced at the data owner by applying coarse-grained encryption by the owner and fine-grained encryption by the cloud. The challenging issue is how to decompose ACPs to perform two layer encryption. The proposed paper also shows that the problem is NP complete and provided the appropriate algorithm and also GKM group key management scheme is used that supports ACPs.

I. INTRODUCTION

Security and privacy represent major concerns in the adoption of cloud technologies for data storage. An approach to mitigate these concerns is the use of encryption. However, whereas encryption assures the confidentiality of the data against the cloud, the use of conventional encryption approaches is not sufficient to support the enforcement of fine-grained organizational access control policies (ACPs). Many organizations have today ACPs regulating which users can access which data; these ACPs are often expressed in terms of the properties of the users, referred to as identity attributes, using access control languages such as XACML. Such an approach, referred to as attribute-based access control (ABAC), supports fine-grained access control (FGAC) which is crucial for high-assurance data security and privacy. Supporting ABAC over encrypted data is a critical requirement in order to utilize cloud storage services for selective data sharing among different users. Notice that often user identity attributes encode private information and should thus is strongly protected from the cloud, very much as the data themselves. Approaches based on encryption have been proposed for fine-grained access control over encrypted data [2], [3],those approaches group data items based on ACPs and encrypt each group with a different symmetric key. Users then are given only the keys for the data items they are allowed to access. Extensions to reduce the number of keys that need to be distributed to the users have been proposed exploiting hierarchical and other relationships among data items.

II. LITERATURE SURVEY

Fine-grained access control allows one to enforce selective access to the content based on expressive policy specifications. Research in FGAC can be categorized into two dissemination models: push-based and pull-based models. Our work focuses on the pull-based model.

In the push-based approaches [1], [2] subdocuments are encrypted with different keys, which are provided to users at the registration phase. The encrypted subdocuments are then broadcasted to all users. However, such approaches require that all [1] or some [2] keys be distributed in advance during user registration phase. This requirement makes it difficult to assure forward and backward key secrecy when user groups are dynamic. Further, the rekey process is not transparent, thus shifting the burden of acquiring new keys on users. Shang et al. [3] proposes approach to solve such problem. It lays the foundation to make rekey transparent to users and protect the privacy of the users who access the content. How-ever, it does not support expressive access control policies as in our approach and also it is not directly applicable to pull base dc approaches.

Under the pull-based model, the content publisher is required to be online in order to provide access to the con-tent. Recent research efforts [4], [6], [7], [8] have pro-posed approaches to construct privacy preserving access control systems using a third-party storage service. In such approaches, the data owner has to enforce the ACPs and the privacy of the users from the content publisher is not protected. Further, in some approaches, multiple encryptions of the same document are required which is inefficient.

A major drawback of all the above approaches is that they do not consider the management of encrypted data hosted in a third party when users are added or removed from the system or when the ACPs/subdocuments are updated. All the approaches require the data owner to handle encryption. Di Vimercati et al. [5] first identifies this problem and proposes an initial solution. While their solution improves over existing solutions, such solution does not support expressive attribute based policies and does not protect the privacy of the users.

Attribute based encryption. The concept of attribute-based encryption has been introduced by Sahai and Waters [9]. The initial ABE system is limited only to threshold policies in which there are at least k out of n attributes common between the attributes used to encrypt the plaintext and the attributes users possess. Pirretti et al. [10] gave an implementation of such a threshold ABE system using a variant of the Sahai-Waters Large Universe construction [9]. Since this initial threshold scheme, a few variants have been introduced to provide more expressive ABE systems. Goyal et al. [11] introduced the idea of key-policy ABE (KP-ABE) systems and Bethen-court et al. [12] introduced the idea of cipher text-policy ABE (CP-ABE) systems. Even though these constructs are expressive and provably secure, they are not suitable for group management and especially in supporting forward security when a user leaves the group (i.e. attribute revocation) and in providing backward security when a new user joins the group. Some of the above schemes suggest using an expiration attribute along with other attributes. However, such a solution is not suitable for a dynamic group where joins and departures are frequent.

Proxy re-encryption (PRE). In a proxy re-encryption scheme [13] one party A delegates its decryption rights to another party B via a third party called a “proxy.” More specifically, the proxy transforms a cipher text computed under party A’s public key into a different cipher text which can be decrypted by party B with B’s private key. In such a scheme neither the proxy nor party B alone can obtain the plaintext. Recently Liang et al.

[14] has extended the traditional PRE to attribute based systems and independently Chu et al. [15] has extended the traditional PRE to support conditions where a proxy can re-encrypt only if the condition specified by A is satisfied. These improved PRE techniques alone or combined with ABE schemes [16] could be utilized to implement dele-gated access control in the cloud. However, they do not protect the identity attributes of the users who access the system and are difficult to manage.

III. METHODOLOGY

3.1 Single Layer Encryption Approach

The SLE scheme [6] consists of the four entities, Owner, Usr, IdP and cloud. They play the following roles:

- Owner, the data owner defines ACPs, and uploads encrypted data to the Cloud, the cloud storage service.
- The Cloud hosts the encrypted data of the Owner.
- IdP, the identity provider, a trusted third party, issues identity tokens to users based on the attribute attributes that users have. An identity token is a signed Pedersen commitment that binds the identity attribute value to aUsr while hiding it from others. There can be one or more certified IdPs. We assume that all IdPs issue identity tokens in the same format.
- Usr, the user, uses one or more identity tokens to gain access to the encrypted data hosted in the cloud.

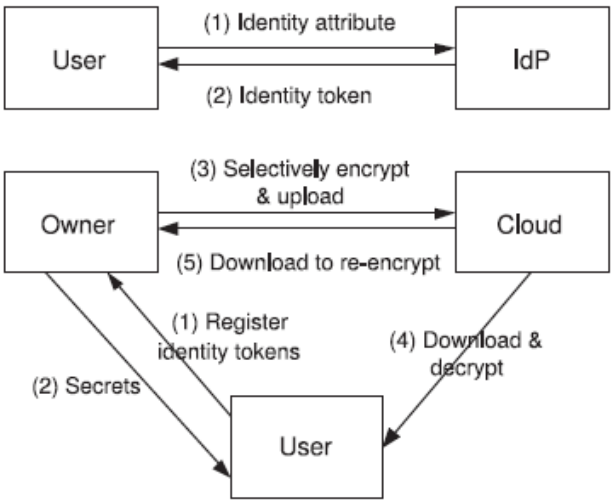

Figure 1.1 Single layer encryption approach

As shown in Figure 1.1, the SLE approach follows the conventional data outsourcing scenario where the Owner enforces all ACPs through selective encryption and uploads encrypted data to the untrusted cloud. The system goes through five different phases. We give an overview of the five phases below:

- **Identity token issuance**.:IdPs issue identity tokens to Usrs based on their identity attributes.
- **Identity token registration**: Usrs register all their identity tokens to obtain secrets in order to later decrypt the data that they are allowed to access.
- **Data encryption and uploading:** Based on the secrets issued and the ACPs, the Owner encrypts the data using the keys generated using the AB-GKM::KeyGen algorithm and uploads to the cloud.
- **Data downloading and decryption:**.Usrs download encrypted data from the cloud and decrypt using the key derived from the AB-GKM::KeyDer algorithm.

- **Encryption evolution management:** Over time, either access control polices or user credentials may change. Further, already encrypted data may go through frequent updates. In such situations, it may be required to re-encrypt already encrypted data. The Owner alone is responsible to perform such re-encryptions. The Owner downloads all affected data from the cloud, decrypts them and then follows the data encryption and upload step.

We now give an overview of our solution to the problem of delegated access control to outsourced data in the cloud. The TLE system consists of the four entities, Owner, Usr, IdP and cloud. However, unlike the SLE approach, the Owner and the cloud collectively enforce ACPs by performing two encryptions on each data item. This two layer enforcement allows one to reduce the load on the Owner and delegates as much access control enforcement duties as possible to the cloud. Specifically, it provides a better way to handle data updates, and user dynamics changes.

3.2 Two Layer Encryption Approach

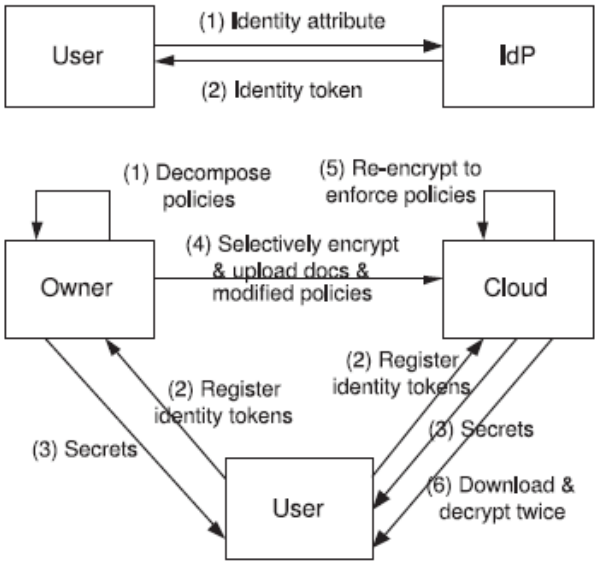

Figure 1.2 Two layer encryption approach.

Figure 1.2 shows the system diagram of the TLE approach. The system goes through one additional phase compared to the SLE approach. We give an overview of the six phases below:

- **Identity token issuance**. IdPs issue identity tokens to Usrs based on their identity attributes.
- **Policy decomposition**. The Owner decomposes each ACP into at most two sub ACPs such that the Owner enforces the minimum number of attributes to assure confidentiality of data from the cloud. It is important to make sure that the decomposed ACPs are consistent so that the sub ACPs together enforce the original ACPs. The Owner enforces the confidentiality related sub ACPs and the cloud enforces the remaining sub ACPs.
- **Identity token registration**. Usrs register their identity tokens in order to obtain secrets to decrypt the data that they are allowed to access. Usrs register only those identity tokens related to the Owner's sub ACPs and register the remaining identity tokens with the cloud in a privacy preserving manner. It should be noted that the cloud does not learn the identity attributes of Usrs during this phase.

- **Data encryption and uploading**: The Owner first encrypts the data based on the Owner's sub ACPs in order to hide the content from the cloud and then uploads them along with the public information generated by the AB-GKM::KeyGen algorithm and the remaining sub ACPs to the cloud. The cloud in turn encrypts the data based on the keys generated using its own AB-GKM::KeyGen algorithm. Note that the AB-GKM::KeyGen at the cloud takes the secrets issued to Usrs and the sub ACPs given by the Owner into consideration to generate keys.
- **Data downloading and decryption**: Usrs download encrypted data from the cloud and decrypt the data using the derived keys. Usrs decrypt twice to first remove the encryption layer added by the cloud and then by the Owner. As access control is enforced through encryption, Usrs can decrypt only those data for which they have valid secrets.
- **Encryption evolution management**: Over time, user credentials may change. Further, already encrypted data may go through frequent updates. In such situations, data already encrypted must be re-encrypted with a new key. As the cloud performs the access control enforcing encryption, it simply re-encrypts the affected data without the intervention of the Owner.

3.3 Two Layer Encryption execution steps

The system consists of the four entities, Owner, Usr, IdP and cloud. Let the maximum number of users in the system be N, the current number of users be n (< N), and the number of attribute conditions N_a.

IV. IDENTITY TOKEN ISSUANCE

IdPs are trusted third parties that issue identity tokens to Usrs based on their identity attributes. It should be noted that IdPs need not be online after they issue identity tokens. An identity token, denoted by IT has the format fnym, id-tag, c, s g, where nym is a pseudonym uniquely identifying a Usr in the system, id-tag is the name of the identity attribute, c is the Pedersen commitment for the identity attribute value x and s is the IdP's digital signature on nym, id-tag and c.

V. POLICY DECOMPOSITION

Using the policy decomposition Algorithm 3, the Owner decomposes each ACP into two sub ACPs such that the Owner enforces the minimum number of attributes to assure confidentiality of data from the cloud. The algorithm produces two sets of sub ACPs, $ACPB_{Owner}$ and $ACPB_{cloud}$. The Owner enforces the confidentiality related sub ACPs in $ACPB_{Owner}$ and the cloud enforces the remaining sub ACPs in $ACPB_{cloud}$.

VI. IDENTITY TOKEN REGISTRATION

Usrs register their IT s to obtain secrets in order to later decrypt the data they are allowed to access. Usrs register their IT s related to the attribute conditions in ACC with the Owner, and the rest of the identity tokens related to the attribute conditions in ACB=ACC with the cloud using the AB-GKM: Sec Genalgorithm.When Usrs register with the Owner, the Owner issues them two sets of secrets for the attribute conditions in ACC that are also

present in the sub ACPs in $ACPB_{cloud}$. The Owner keeps one set and gives the other set to the cloud. Two different sets are used in order to prevent the cloud from decrypting the Owner encrypted data.

VII. DATA ENCRYPTION AND UPLOAD

The Owner encrypts the data based on the sub ACPs in $ACPB_{Owner}$ and uploads them along with the corresponding public information tuples to the cloud. The cloud in turn encrypts the data again based on the sub ACPs in $ACPB_{cloud}$. Both parties execute AB-GKM::KeyGen algorithm individually to first generate the symmetric key, the public information tuple PI and access tree T for each sub ACP. We now give a detailed description of the encryption process.

VIII. DATA DOWNLOADING AND DECRYPTION

Usrs download encrypted data from the cloud and decrypt twice to access the data. First, the cloud generated public information tuple is used to derive the OLE key and then the Owner generated public information tuple is used to derive the ILE key using the AB-GKM::KeyDer algorithm.

IX. ENCRYPTION EVOLUTION MANAGEMENT

After the initial encryption is performed, affected data items need to be re-encrypted with a new symmetric key if credentials are added/removed. Unlike the SLE approach, when credentials are added or revoked, the Owner does not have to involve. The cloud generates a new symmetric key and re-encrypts the affected data items.

X. CONCLUSION

Conventional approaches for data encryption to manage the secret keys and uploading the data to the on to the cloud even after changing the user credentials. Such an approach cause high communication and computation cost. The proposed paper uses TLE a two layer encryption which prevents the overhead at the data owner by performing two layer encryption which also minimizes the information exposure risk due to colluding Users and cloud. The challenging issue is how to decompose ACPs to perform two layer encryption. The policy decomposition problem is NP-Complete and provided approximation algorithms. Based on the decomposed ACPs, The paper proposes a novel approach to privacy preserving fine-grained delegated access control to data in public clouds. The approach is based on a privacy preserving attribute based key management scheme that protects the privacy of users while enforcing attribute based ACPs.

REFERENCES

[1] E. Bertino and E. Ferrari, “Secure and Selective Dissemination of XML Documents,” ACM Trans. Information and System Security,vol. 5, no. 3, pp. 290-321, 2002.

[2] G. Miklau and D. Suciu, “Controlling Access to Published Data Using Cryptography,” Proc. 29th Int’l Conf. Very Large Data Bases (VLDB ’03), pp. 898-909, 2003.

[3] N. Shang, M. Nabeel, F. Paci, and E. Bertino, "A Privacy-Preserving Approach to Policy-Based Content Dissemination," Proc. IEEE 26th Int'l Conf. Data Eng. (ICDE '10), 2010.

[4] M.Nabeel, E. Bertino, M. Kantarcioglu, and B.M. Thuraisingham, "Towards Privacy Preserving Access Control in the Cloud," Proc.Seventh Int'l Conf. Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom '11), pp. 172-180, 2011.

[5] S.D.C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P.Samarati, "Over-Encryption: Management of Access Control Evolution on Outsourced Data," Proc. 33rd Int'l Conf. Very Large Data Bases (VLDB '07), pp. 123-134, 2007.

[6] S. Coull, M. Green, and S. Hohenberger, "Controlling Access to an Oblivious Database Using Stateful Anonymous Credentials,"Proc. 12th Int'l Conf. Practice and Theory in Public Key Cryptography,pp. 501-520, 2009.

[7] J. Camenisch, M. Dubovitskaya, and G. Neven, "Oblivious Transfer with Access Control," Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), pp. 131-140, 2009.

[8] K.P.N. Puttaswamy, C. Kruegel, and B.Y. Zhao, "Silverline:Toward Data Confidentiality in Storage-Intensive Cloud Applications," Proc. Second ACM Symp. Cloud Computing (SOCC '11), pp. 10:1-10:13, 2011.

[9] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," Proc 24th Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques(Eurocrypt '05), pp. 457-473, 2005.

[10] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure Attribute-Based Systems," Proc. 13th ACM Conf. Computer and Comm. Security (CCS '06), pp. 99-112, 2006.

[11] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data,"Proc. 13th ACM Conf. Computer and Comm. Security (CCS '06),pp. 89-98, 2006.

[12] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," Proc. IEEE Symp.Security and Privacy (SP'07), pp. 321-334, 2007.

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

[14] X. Liang, Z. Cao, H. Lin, and J. Shao, "Attribute Based Proxy Re-Encryption with Delegating Capabilities," Proc. Fourth Int'l Symp.Information, Computer, and Comm. Security (ASIACCS '09), pp. 276-286, 2009.

[15] C.-K. Chu, J. Weng, S. Chow, J. Zhou, and R. Deng, "Conditional Proxy Broadcast Re-Encryption," Proc. 14th Australasian Conf.Information Security and Privacy, pp. 327-342, 2009.

[16] J.-M. Do, Y.-J. Song, and N. Park, "Attribute Based Proxy Re-Encryption for Data Confidentiality in Cloud Computing Environments,"Proc.First Int'l Conf. Computers, Networks, Systems and Industrial Eng., pp. 248-251, 2011.