Preserving Privacy of Public Clouds through Fine-Grained and Delegated Access Control Approaches

M.Revanth
M.Tech CS, Narayana Engineering College, Gudur, AP, India.

N. Koteswar Rao
Associate Professor, IT, Narayana Engineering College, Gudur, AP, India.

Abstract – Current approaches to enforce fine-grained access control on confidential data hosted in the cloud are based on fine-grained encryption of the data. Under such approaches, data owners are in charge of encrypting the data before uploading them on the cloud and re-encrypting the data whenever user credentials or authorization policies change. Data owners thus incur high communication and computation costs. A better approach should delegate the enforcement of fine-grained access control to the cloud, so to minimize the overhead at the data owners, while assuring data confidentiality from the cloud. We propose an approach, based on two layers of encryption that addresses such requirement. Under our approach, the data owner performs a coarse-grained encryption, whereas the cloud performs a fine-grained encryption on top of the owner encrypted data. A challenging issue is how to decompose access control policies (ACPs) such that the two layer encryption can be performed. We show that this problem is NP-complete and propose novel optimization algorithms. We utilize an efficient group key management scheme that supports expressive ACPs. Our system assures the confidentiality of the data and preserves the privacy of users from the cloud while delegating most of the access control enforcement to the cloud.

Index Terms – Delegated access, Fine-grained access, Privacy preserving, Public Cloud.

1. INTRODUCTION

In order to preserve security and privacy of data items stored in the cloud, access control policies must be enforced to users that define which user can access which data. These access control policies are derived from the identity attributes of the users. But providing identity attributes to owners or clouds could reveal their identity. This may contain personal information about users which can be a threat to the privacy of users therefore must be protected from the cloud.[1] As a solution to this issue users can register at a key management module to retrieve tokens. These tokens further be used to derive security keys using which the users re-encrypt the data. Data owners encrypt the data using ACP's, so that only users who satisfy the policies will be given the key to decrypt them. This approach can have several limitations as follows:

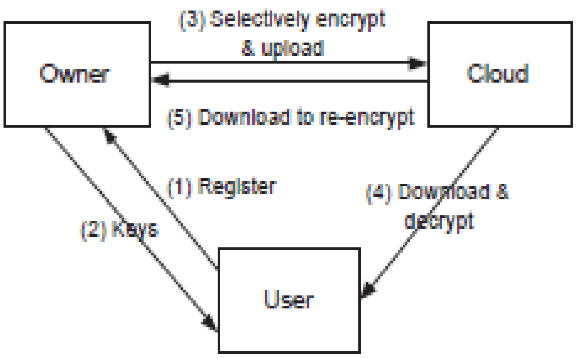
• Data Owners does not keep the copy of data, therefore when the user profile or the policies are updated, the data owner needs to download the data again from the cloud to re-encrypt them with new keys.

•New keys are to be communicated with the users.

The goal of this paper is to provide an overview of our approaches to enforce delegated access and fine-grained access control on sensitive data stored in untrusted public clouds, while at the same assuring the confidentiality of the data from the cloud and preserving the privacy of users who are authorized to access the data. We compare these approaches and discuss about open issues.

2. RELATED WORK

Approaches based on encryption have been proposed for fine-grained access control over encrypted 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 [2] [3]. Such approaches however have several limitations: As the data owner does not keep a copy of the data, whenever the user dynamics or ACPs change, the data owner needs to download and decrypt the data, re-encrypt it with the new keys, and upload the encrypted data. Notice also that this process must be applied to all the data items encrypted with the same key. This is inefficient when the data set to be re-encrypted is large. In order to issue the new keys to the users, the data owner needs to establish private communication channels with the users. The privacy of the identity attributes of the users is not taken into account. Therefore the cloud can learn sensitive information about the users and their organization [4]. They are either unable or inefficient in supporting fine-grained ABAC policies. It requires the data owner to enforce *all* the ACPs by fine-grained encryption, both initially and subsequently after users are added/revoked or the ACPs change. All these encryption activities have to be performed at the owner that thus incurs high communication and computation cost [7] [8].



- The main drawback of this scheme is the high resource costs it requires for the implementation.[5][6]
- Also computing hash value for even a moderately large data files can be computationally burdensome for some clients (PDAs, mobile phones, etc).
- Data encryption is large so the disadvantage is small users with limited computational power (PDAs, mobile phones etc.).

3. PROPOSED WORK

A challenging issue in the TLE approach is how to decompose the ACPs so that fine-grained ABAC enforcement can be delegated to the cloud while at the same time the privacy of the identity attributes of the users and confidentiality of the data are assured. The TLE approach has many advantages. When the policy or user dynamics changes, only the outer layer of the encryption needs to be updated. Since the outer layer encryption is performed at the cloud, no data transmission is required between the data owner and the cloud. Further, both the data owner and the cloud service utilize a broadcast key management scheme whereby the actual keys do not need to be distributed to the users. Instead, users are given one or more secrets which allow them to derive the actual symmetric keys for decrypting the data.

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, user dynamics, and policy changes. The system goes through one additional phase compared to existing approach.

Identity token issuance, IdPs are trusted third parties that issue identity tokens to Users based on their identity attributes. It should be noted that IdPs need not be online after they issue identity tokens.

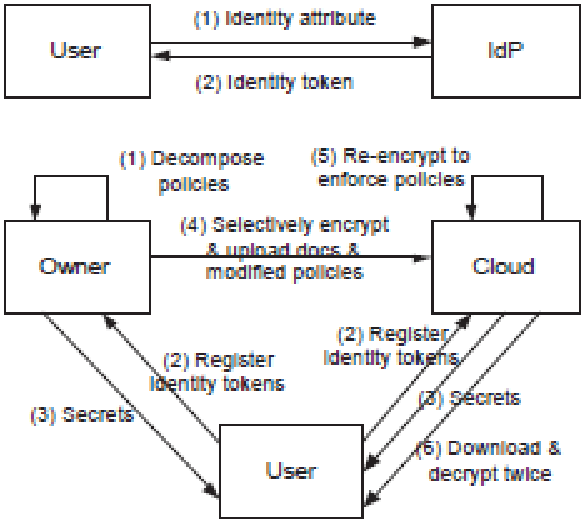

User
(1) Identity attribute
IdP
(2) Identity token
(1) Decompose policies
(5) Re-encrypt to enforce policies
(4) Selectively encrypt & upload docs & modified policies
Owner
Cloud
(2) Register identity tokens
(2) Register identity tokens
(3) Secrets
(3) Secrets
User
(6) Download & decrypt twice

Fig: Overall System Architecture

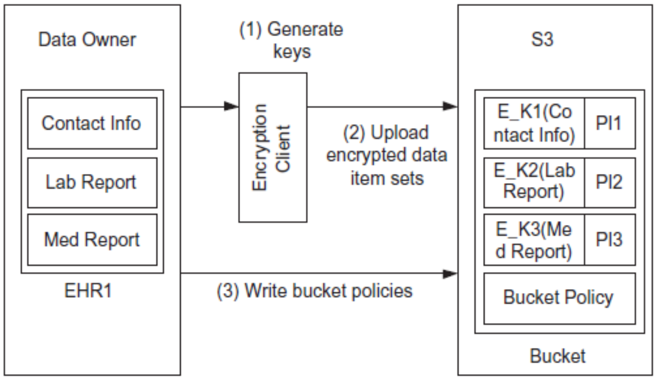


Fig: Overall Involvement of the Owner

Identity token registration, Users register their token to obtain secrets in order to later decrypt the data they are allowed to access. Users register their tokens 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. When Users 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.

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 Users and the sub ACPs given by the Owner into consideration to generate keys.

Data downloading and Decryption, Users 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. These two keys allow a User to decrypt a data item only if the User satisfies the original ACP applied to the data item.

Encryption Evolution Management, over time, either ACPs or 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.

4. TWO-LAYER ENCRYPTION APPROACH TO PRIVACY-PRESERVING ABAC

Our basic approach follows the conventional data outsourcing scenario where the Owner enforces *all* the access control policies through selective encryption and uploads encrypted data to the untrusted Cloud. We refer to this approach as single layer encryption (SLE). The SLE approach supports fine-grained attribute-based access control policies and preserves the privacy of users from the Cloud. This section, we provide an overview of an approach, based on two layers of encryption, that addresses such requirement. Under such approach, referred to as *two-layer encryption* (TLE), the Owner performs a coarse grained encryption, whereas the Cloud performs a fine grained encryption on top of the data encrypted by the coarse grained encryption. A challenging issue in this approach is how to decompose the ABAC policies such that the two-layer encryption can be performed. In order to delegate as much access control enforcement as possible to the Cloud, one needs to decompose the ABAC policies so that the Owner only needs to manage the minimum number of attribute conditions in these policies that assures the confidentiality of data from the Cloud. Each policy should be decomposed into two sub policies such that the conjunction of the two sub policies result in the original policy. The two-layer encryption should be performed such that the Owner first encrypts the data based on one set of sub policies and the Cloud re-encrypts the encrypted data using the other set of policies.

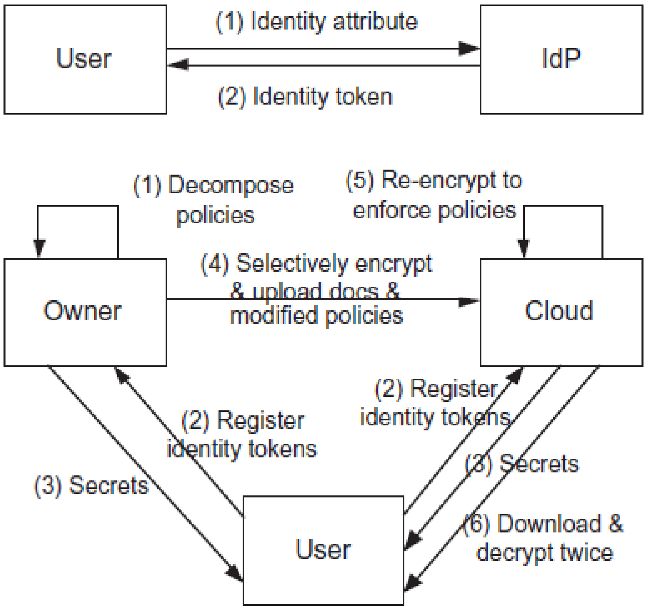

User
(1) Identity attribute
IdP
(2) Identity token
(1) Decompose policies
(5) Re-encrypt to enforce policies
Owner
(4) Selectively encrypt & upload docs & modified policies
Cloud
(2) Register identity tokens
(2) Register identity tokens
(3) Secrets
(3) Secrets
User
(6) Download & decrypt twice

Fig: Two Layer Encryption Approach

Comparison

Property	ABE	SLE	TLE
Cryptosystem	Asymmetric	Symmetric	Symmetric
Secure attribute based group communication	Yes	Yes	Yes
Efficient revocation	No	Yes	Yes
Delegation of access control	No	No	Yes

5. CONCLUSION

Current trends in computing infrastructures like Service Oriented Architectures (SOAs) and cloud computing are further pushing publishing functions to third-party providers to achieve economies of scale. However, recent surveys by IEEE and Cloud Security Alliance (CSA) have found that one of the key resistance factor for companies and institutions to move to the cloud is represented by data privacy and security concerns. Our AB-GKM based approaches address such privacy and security concerns in the context of efficient and flexible sharing and management of sensitive content. Compared to state of the art ABE based approaches, our approaches support efficient revocation and management of users which is a key requirement to construct scalable solutions.

REFERENCES

[1] J. Camenisch, M. Dubovitskaya, R. R. Enderlein, and G. Neven. Oblivious transfer with hidden access control from attribute-based encryption. In SCN 2012: Proceedings of the 8th International Conference on Security and Cryptography for Networks, pages 559–579, 2012.

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

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

[4] G.Ateniese, R. Burns, R.urtmola, J.Herring, L. Kissner, Z. Peterson, and D.Song, “Deduplication in cloud storage using side channels in cloud services,” Oct 2008.

[5] K. D. Bowers, A. Juels, and A. Oprea, ”Hail: A high availability and integrity layer for cloud storage,” in Proc. Of CCS’09, 2009, pp. 187-198.

[6] S. Kamara and K. Lauter, “Cryptographic Cloud Storage,” Proc. Int‟l Conf. Financial Cryptography and Data Security (FC), pp. 136-149, Jan. 2010.

[7] X.Liu,B.Wang,Y.Zhang, and J.Yan,”Mona: Secure Multi Owner Data Sharing for Dynamic Groups in the Cloud,”IEEE Computer Society,vol. 24,no. 6,June. 2013.

[8] G. Miklau and D. Suciu, “Controlling access to published data using cryptography,” in VLDB ’2003: Proceedings of the 29th international conference on Very large data bases. VLDB Endowment, 2003, pp. 898–909.