A Secure Architecture for Access Control with Hashing Technique in Public Cloud

Revathi. M[1], Divyashree. B. A[2]
M. Tech (CNE) II year[1], Professor[2]
Department of Computer Science,
BNMIT (Affiliated to VTU), Bangalore, India

Abstract— **Traditional approaches are inefficient to support fine-grained encryption. Under such approaches, data owner is responsible for encrypting the data before uploading them into cloud and re-encrypting the data whenever there is a change in user's sub access control policy. Data owners thus incur high communication and computation costs. In order to minimize the overhead at the data owner, fine-grained access control has to be delegated to the cloud. The cloud performs a fine-grained encryption and the data owner performs coarse-grained encryption. Cloud encrypts the data that is already encrypted by the data owner. Efficient cryptographic algorithms such as Residual Number System (RNS) and Elliptical Curve Cryptography (ECC) are implemented. Both RNS and ECC reduce the computational cost. Integrity check is also been used. Thus the proposed system preserves user's privacy and ensures data confidentiality.**

Keywords— **Cloud Computing, Encryption, Fine-grained access control, Hashing, Privacy.**

I. INTRODUCTION

Cloud computing is an emerging technology which provides robust computational power at reduced cost to the society. It enables users with limited resources to outsource large computational task to the cloud. It also provides software that can be used in pay-per-use basis. It provides resources that can be easily deployed. In spite of the benefits of the cloud, major issue is the security. Thus the data has to be encrypted before outsourcing it.

Though, encryption assures the confidentiality of the data against the cloud, the use of traditional encryption approaches is not sufficient to support the enforcement of fine-grained access control policies.

Many organizations have today access control policies (ACPs) directing which clients can access which data. ACPs are expressed in terms of the properties of the users, referred to as identity attributes, using access control languages known as eXtensible Access Control Markup Language (XACML). Such an approach is referred to as attribute based access control (ABAC). The ABAC supports fine-grained access control [12] which assures data security and privacy.

II. RELATED WORK

In Single Layer Encryption (SLE), the Owner enforces all access control policies (ACPs) through selective encryption and uploads encrypted data to the untrusted Cloud. Whenever the user sub access control policies changes, Owner has to download the data from the Cloud, decrypt it, re-encrypt it using a new key and then the re-encrypted data is uploaded to Cloud. Thus there is a high computation and communication overhead at the Owner. In Attribute based encryption for fine grained access control of encrypted data [2], a scheme known as Key Policy-Attribute Based Encryption (KP-ABE) was proposed where each ciphertext is labeled with a set of attributes and private key is related to the access structure. KP-ABE scheme does not hide the attributes under which data has to be encrypted. If it were possible to hide the attributes then it would lead to keyword based search on encrypted data. In Ciphertext policy attribute based Encryption [3], private key of a user will be associated with number of attribute. An access structure is specified while encrypting a message. A user will be able to decrypt a ciphertext if that user's attributes pass through the ciphertext's access structure.

Improved proxy re-encryption schemes with applications to secure distributed storage [4], the data owner encrypts group of content using a symmetric key. But this encryption scheme does not protect the symmetric keys. Scalable secure file sharing on untrusted storage [6], a cryptographic system known as Plutus was proposed to secure file sharing on untrusted servers. In this system, files which share a similar attribute are grouped together and each file group is associated with a symmetric key.

III. PROPOSED SYSTEM

The proposed approach applies two layers of encryption to each data before uploading them into the cloud. In two layer encryption (TLE) approach, the data owner performs a coarse grained encryption [11] over the data to guarantee data confidentiality. Then the Cloud performs fine grained encryption over the data that is encrypted by the Data Owner.

Consider hospital as an example. Hospital acts as Data Owner which stores Electronic Health Records (EHRs) in public cloud and makes it available to hospital employees. Here Domain attribute specifies roles (doctor, nurse, receptionist, cashier, etc) of employees and Sub-Domain attribute specifies type (assistant, junior, senior) of an employee. Fine-grained access control specifies which user can access which data item. In hospital, a doctor must be able to view a patient's record and cashier must be able to view billing information.

Figure 1 shows the architecture of two layer encryption approach. Data owner is one who stores the file into Cloud; these files are in turn accessed by the user. Data Owner specifies the access control policy for each file. Domain and Sub-Domain Attributes of a user are used to set the Access Control Policies (ACPs).

User sends identity attributes such as email id, name, domain and sub-domain values to Identity Provider (IdP). Identity Provider issues identity token to the user. Identity token is the hash value of the above identity attributes.

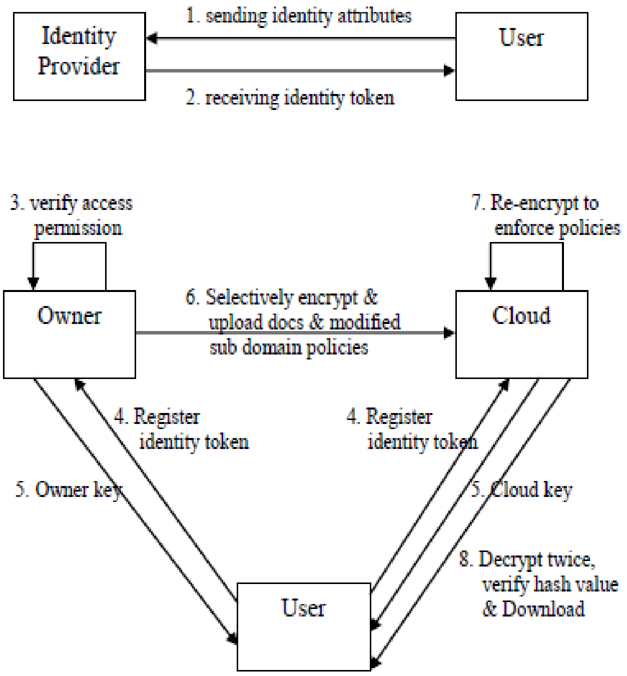
Identity Provider
1. sending identity attributes
User
2. receiving identity token
3. verify access permission
7. Re-encrypt to enforce policies
Owner
6. Selectively encrypt & upload docs & modified sub domain policies
Cloud
4. Register identity token
4. Register identity token
5. Owner key
5. Cloud key
8. Decrypt twice, verify hash value & Download
User

Figure 1. Two Layer Encryption Architecture

User registers identity token with both Data Owner and Cloud. Data Owner uses RNS algorithm [7] to encrypt the file and then uploads encrypted file along with sub domain policies to the Cloud. Here the sub domain policies are delegated [1] to the cloud. Cloud in turn encrypts the file using ECC algorithm [8].

User sends appropriate secret keys (owner and cloud key) to decrypt the file. Hash value of the file is verified. If the hash value matches then the file will be downloaded to the system else file will not be downloaded.

2.1 Modules

A. Identity token issuance

Identity Provider issues identity tokens to the users based on their identity attributes.

B. Identity token registration and secret key generation

Users register their identity tokens to obtain secret keys in order to decrypt the data later.

C. Data Encryption and Uploading

Owner encrypts the data using owner key and uploads the encrypted data along with sub domain details to the Cloud. Cloud in turn encrypts the data using Cloud key.

D. Data Decryption and Downloading

User can download encrypted data from the Cloud. First, the Outer Layer Encryption (OLE) is removed using cloud key and the Inner Layer Encryption (ILE) is removed using owner key and finally, hash value of the file is verified. If hash codes matches then file will be downloaded to user's system.

E. Encryption Evolution Management

After some time, the access control policies may change. Further, already encrypted data may go through frequent changes [5], [10]. In such circumstances, it might be needed to re-encrypt [9] the already encrypted data. Cloud generates a new cloud key and performs re-encryption without the intervention of Data Owner.

VI. RESULT AND DISCUSSIONS

Encryption 1 and Encryption 2 shown in figure 2 are the results of proposed and traditional system respectively. The x-axis in the graph shows the number of files uploaded and y-axis shows the time taken for encryption.

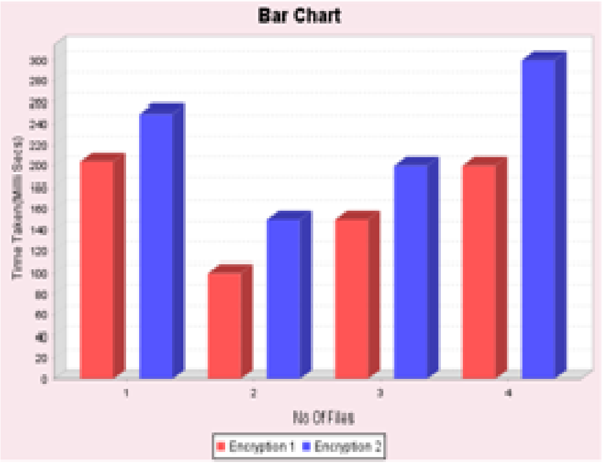

Figure 2. Two Layer Encryption System Result

Encryption 2 (traditional approach) enforces the ACPs through a single encryption. In this approach, the Owner itself performs the attribute based encryption based on ACPs. Encryption 1 (two layer encryption approach) enforces the ACPs through two encryptions. Here the Data Owner performs coarse-grained encryption and Cloud performs fine-grained encryption. A user can access the data only if he is able to decrypt both encryptions.

In Encryption 2, as the user sub domain policies changes, Data Owner performs re-encryption and issues new keys to the user. This creates high computational workload at the Data Owner. In Encryption 1, the sub domain policies are delegated to the Cloud. Hence whenever user's sub domain policies changes, Cloud performs re-encryption without intervention of the Data Owner. Thus, the computational overhead at the Data Owner is reduced in the proposed system, which is shown in the graph.

V. CONCLUSION AND FUTURE ENHANCEMENT

Traditional approaches are inefficient to support the enforcement of fine-grained access control policies and are difficult to manage all the encryption keys. Such approaches incur high communication and computation cost to manage encryption keys whenever user sub access control policies changes. Thus a two layer encryption approach has been proposed to solve the above problem. The proposed system delegates sub access control policies to the Cloud. Whenever sub domain policies changes, Cloud performs re-encryption on behalf of Data Owner.

The approach is based on attribute based key management scheme which protects the privacy of users and it also uses efficient cryptographic algorithms such as RNS and ECC. Finally, hash code of the file is verified for integrity. The experimental results shows that the two layer encryption approach had reduced the overhead at the Owner. Thus the proposed system does not incur high communication and computation cost.

In the future work, alternative designs for two layer encryption approach have to be determined. Also the relationship between the access control policies can be determined to further reduce the computational cost.

REFERENCES

[1] M. Nabeel and E. Bertino, "Privacy preserving delegated access control in the storage as a service model," in IEEE International Conference on information Reuse and Integration(IRI), 2012.

[2] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in CCS '06: Proceedings of the 13th ACM conference on Computer and communications security. New York, NY, USA: ACM, 2006, pp. 89–98.

[3] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in SP '07: Proceedings of the 2007 IEEE Symposium on Security and Privacy. Washington, DC, USA: IEEE Computer Society, 2007, pp. 321–334.

[4] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," ACM Transaction on Information System Security, vol. 9, pp. 1–30, February 2006.

[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," in Proceedings of the 33rd International Conference on Very Large Data Bases, ser. VLDB '07. VLDB Endowment, 2007, pp. 123–134.

[6] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang and K. Fu, "Scalable secure file sharing on untrusted storage," in Proc. Of FAST'03, 2003.

[7] Bajard J. C, Didier. L. S and Kornerup. P, "An RNS montgomery modular multiplication algorithm," IEEE TRANSACTIONS ON COMPUTERS 47, 7 (1998), 766-776.

[8] I. Blake, G. Seroussi and N. Smart, "Elliptic Curves in Cryptography," London Mathematical Society, Lecture Note Series 265, Cambridge University Press, 1999.

[9] Cheng-Kang Chu, Jian Weng, Sherman S. M. Chow, Jianying Zhou and Robert H. Deng, "Conditional Proxy Broadcast Re-Encryption," in Proc. Of the 14th Australasian Conference on Information Security and Privacy, 2009, pp. 327–342.

[10] Shucheng Yu, Cong Wang, and Kui Ren, "Attribute Based Data Sharing with Attribute Revocation," In ASIACCS 2010: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, pages 261-270, 2010.

[11] 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. Collaborate Com '11, 2011, pp. 172–180.

[12] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing," Proc. IEEE INFOCOM, pp. 534-542, 2010.