ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 2 February,2014 Page No. 3938-3942

EFFICIENT USER REVOCATION FOR DYNAMIC GROUPS IN THE CLOUD

Kanya Devi J, Kanimozhi S
Assistant Professor
Department of Computer Science and Engineerin Sri Shakthi Institute of Engineering and Technology
Coimbatore-62
jkanya@siet.ac.in
PG scholar
Department of Computer Science and Engineering Sri Shakthi Institute of Engineering and Technology
Coimbatore-62
kanilidigomtech@gmail.com

***Abstract*:**

Mona, secure data sharing in a multi-owner manner for dynamic groups preserves data, identity privacy from an untrusted cloud and allows frequent change of the membership. In RLS while the number of invoked users grows larger, the length of RL increases. To send all user revocation details to the group members for sharing purpose, leads to communication overhead .To address this issue, in this paper, By leveraging group signature and dynamic broadcast encryption techniques and for overall security Elliptic curve cryptography(ECC) algorithm is used, so that any cloud user can anonymously share data with others. The storage overhead and encryption computation cost of the scheme are independent with the number of revoked users.

Index Terms: Cloud computing, access control, dynamic groups, and data sharing

1. INTRODUCTION

When you store your photos online on your personal computer and in a social networking site, it's a "cloud computing" service. In an organization, you want to use an online invoicing service instead of updating the in-house one you have been using for many years, that online service is a cloud computing service.

Cloud computing is mainly used for resource-sharing and with very low-maintenance. The cloud service providers (CSPs), such as Amazon, are able to provide a various services to cloud users with the help of powerful various datacenters. Cloud Providers provides a fundamental service is data storage (Storage as-a service). An organisation allows its group members in the same group or department to store and share files in the cloud. By utilizing the cloud, the group members can be completely released from its local data storage and maintenance. A significant risk arises in confidentiality of those stored files. So, the users are not fully trusted the cloud servers operated by cloud provider while sensitive data stored in the cloud.

To preserve data privacy and confidentiality, a basic solution is to encrypt data files, and then upload the encrypted data into the cloud [7].

The main issues of building secure cloud storage service on top of a public cloud infrastructure where the service provider is not completely trusted by the group users. S.Kamara [7] described several architectures such as consumer architecture wishes to upload data, to verify the integrity of the data and to retrieve the data from the cloud. By invoking the data processor to upload the data, invoking the data verifier to verify the integrity of the data and invoking the token generator to retrieve the data that combine recent and non-standard cryptographic primitives in order to achieve the goal. To increase the adoption of cloud storage, designed a virtual private storage services are based on cryptographic techniques. Service should provide confidentiality and integrity. The main benefits of a public storage services are availability, reliability, efficient retrieval, and data sharing.

When preparing data to store in the cloud, the data processor begins by indexing it and encrypting it with a *symmetric encryption* scheme (e.g., AES) under a unique key refer to single writer/single reader (SWSR). It then encrypts the index using a *searchable encryption* scheme and encrypts the unique key with an *attribute-based encryption* scheme under an appropriate policy. Finally, it encodes the encrypted data and index in such a way that the data verifier can later verify their integrity using a proof of storage.

Asymmetric searchable encryption (ASE) schemes where the party searching over the data is different from the party that generates and refer to many writer/single reader (MWSR).It is very inefficient. Attribute-based encryption scheme each user in the system is provided with a decryption key that has a set of attributes associated with it.

A user can then encrypt a message under a public key and a policy. Decryption will only work if the attributes associated with the decryption key match the policy used to encrypt the message.

S.Yu [12] described access policies based on data attributes and allowing the data owner to delegate most of the computation tasks involved in fine grained data access control to untrusted cloud servers without disclosing the underlying data contents. The issue is mainly caused by the operation of user revocation, which inevitably requires the data owner to re-encrypt all the data files accessible to the leaving user, or even needs the data owner to stay online to update secret keys for users. Achieve this goal, by uniquely combining techniques of attribute-based encryption (ABE), proxy re-encryption, and lazy re-encryption.

Data confidentiality is also achieved since cloud servers are not able to learn the plaintext of any data file in our construction. For further reducing the computation overhead on cloud Servers and thus saving the data owner's investment, take advantage of the lazy re-encryption technique and allow Cloud Servers to "aggregate" computation tasks of multiple system operations.

V.Goyal [11] described an efficient system that was expressive; it allowed to encrypt or to express an access predicate in terms of any monotonic formula over attributes. The techniques provide a framework for directly realizing provably secure CP-ABE systems. In this, the ciphertext distributes shares of a secret encryption exponent across different attributes according to the access control LSSS matrix M. A user's private key is associated with a set S of attributes and he will be able to decrypt a ciphertext if his attributes "satisfy" the access matrix associated with the ciphertext.

2. RELATED WORK

M.Kallahalla [6] described storage systems and individual storage devices themselves become networked, they must defend both against the usual attacks on messages traversing an untrusted potentially public network as well as attacks on the stored data itself. This is a challenge because the primary purpose of networked storage is to enable easy sharing of data, which is often at odds with data security.

To protect stored data, it is not sufficient to use traditional network security techniques that are used for securing messages between pairs of users or between clients and servers. Thinking of a stored data item as simply a message with very long network latency is a misleading analogy. Since the same piece of data could be read by multiple users, when one user places data into a shared storage system, the eventual recipient of this "message (stored data item) is often not known in advance. In addition, because multiple users could update the same piece of data, a third user may from time-to-time update "the message" before it reaches its eventual recipient. Stored data must be protected over longer periods of time than typical message round-trip times.

Existing secure storage solutions (encryption-wire) require the creators of data to trust the storage server to control all users' access to this data as well as return the data intact. Most of these storage systems cater to single users, and very few allow secure sharing of data any better than by sharing a pass word. So, introduces a new secure file system, Plutus, which strives to provide strong security even with an untrusted server. The main feature of Plutus is that all data is stored encrypted and all key distribution is handled in a decentralized manner. All cryptographic and key management operations are performed by the clients, and the server incurs very little cryptographic overhead. Plutus uses to provide basic file system security features such as to detect and prevent unauthorized data modifications, to differentiate between read and write access to files, and to change users' access privileges.

Plutus is an encrypt-on-disk system where all the key management and distribution is handled by the client. The advantage of doing this over existing encrypt-on-wire systems is that we can protect against data leakage attacks on the physical device, such as by an untrusted administrator, a stolen laptop, or a compromised server; allow users to set arbitrary policies for key distribution (and therefore file sharing); and enable better server scalability because most of the computationally intensive cryptographic operations are performed at end systems, rather than in centralized servers. Plutus cryptography is performed on clients, not servers. So Plutus has superior scalability along with stronger security.

3. GROUP SIGNATURE

Chaum and van Heyst [4] first introduced the concept of group signatures. In general, a group signature scheme allows any member of the group to sign messages while keeping the identity secret from verifiers. The variant of the short group signature scheme [1] will be used to achieve anonymous access control, as it supports efficient member-ship revocation.

D. Boneh [1] described short signatures in the scheme are approximately the size of a standard RSA signature with the same security. Security of the group signature is based on the Strong Diffie-Hellman assumption and a new assumption in bilinear groups called the Decision Linear assumption.

The Decision Linear problem gives rise to the linear encryption (LE) scheme, a natural extension of ElGamal encryption. Unlike ElGamal encryption, linear encryption can be secure even in groups where a DDH-deciding algorithm exists. In this scheme, a user's public key is a triple of generators; her private key is the exponents. To encrypt a message, choose random values, and output the triple.

To recover the message from an encryption, the user computes. By a natural extension of the proof of security of ElGamal, LE is semantically secure against a chosen-plaintext attack.

A number of revocation mechanisms for group signatures have been described. All these mechanisms can be applied to the system. The Revocation Authority (RA) publishes a Revocation List (RL) containing the private keys of all revoked users. Consequently the Revocation List can be derived directly from the private keys of revoked users. The list RL is given to all signers and verifiers in the system. It is used to update the group public key used to verify signatures. The given RL, anyone can compute this new public key, and any unrevoked user can update her private key locally so that it is well formed with respect to this new public key. Revoked users are unable to do so.

Bellare et al described three properties that a group signature scheme must satisfy: Correctness, which ensures

that honestly-generated signatures verify and trace correctly; Full-anonymity, which ensures that signatures do not reveal their signer's identity; and Full-traceability, which ensures that all signatures, even those created by the collusion of multiple users and the group manager, trace to a member of the forging. Consequently we get a group signature whose length is under 200 bytes — less than twice the length of an ordinary RSA signature (128 bytes) with comparable security. Signature generation requires no bilinear pairing computations, and verification requires a single pairing.

4. DYNAMIC BROADCAST ENCRYPTION TECHNIQUES

Broadcast encryption [5] enables a broadcaster to transmit encrypted data to a set of users so that only a privileged subset of users can decrypt the data. A. Fiat [5] described a broadcaster encrypts messages and transmits these to a group of users who are listening to a broadcast channel and use their private keys to decrypt transmissions.

Cecile described dynamic broadcast encryption scheme involves two authorities: a group manager and a broadcaster. The group manager grants new members access to the group by providing to each new member a public label lab and a decryption key dk. The generation of (lab, dk) is performed using a secret manager key. The broadcaster encrypts messages and transmits these to the whole group of users through the broadcast channel.

In a public-key broadcast encryption scheme, the broadcaster does not hold any private information and encryption is performed with the help of a public group encryption key ek containing. When the broadcaster encrypts a message, some group members can be revoked temporarily from decrypting the broadcast content thanks to a one-time revocation mechanism. The KEM-DEM methodology, broadcast encryption is viewed as the combination of a specific key encapsulation mechanism (a Broadcast-KEM) with a symmetric encryption (DEM) that remains implicit. It leaves as an open problem to realize dynamic public-key broadcast encryption with an encryption key substantially. Finally, expect our trapdoor mechanism to find other cryptographic applications in the future.

5. SYSTEM MODEL AND ITS DESIGN GOALS

5.1 *SYSTEM MODEL*

We consider a cloud computing architecture by combining with an example that an organisation uses a cloud to enable its employees in the same group or department to share files. The system model consists of three different entities: the cloud server, a group manager, and a large number of group members (i.e., the employees) as illustrated in Fig. 1

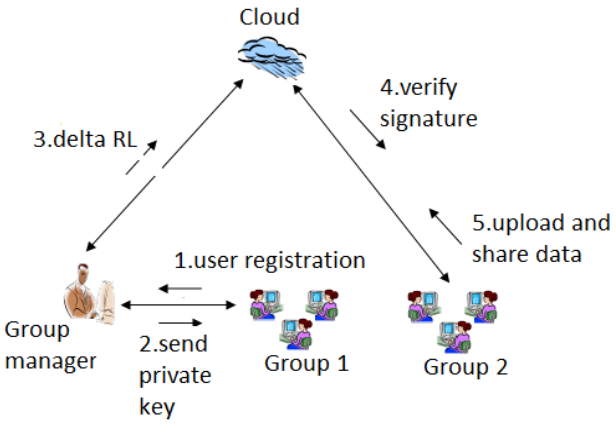

Fig.1 System model

Cloud server is operated by cloud service providers and the fundamental service provides by them as storage as a service (SaaS). However, the cloud is not fully trusted by the group members. We assume that the cloud server is honest and trust them.

So that cloud server will not maliciously delete or modify user data, by achieving data auditing schemes.

Group manager is responsible for system parameters generation, registering the user, revocating the group member and revealing the real identity incase of any dispute occur. In the given example, the group manager is acted by the administrator of the organisation and group manager is fully trusted by the other parties.

Group members are the registered users they will stockpile their private data into the cloud server and share the data among the group members. In our example, the employee plays the role of group members. It allows the group members to be dynamically changed, due to the staff resignation and the participation of new employee in the organisation.

5.2 DESIGN GOALS

Access control: Cloud Server allows only the authorized group member to store their private data in the cloud offered by cloud service providers as SaaS and it won't allow unauthorized group member to store their data in the cloud.

Data confidentiality: Data owner will store their data in the cloud and share the data among the group members. Who upload the data have rights to modify and delete their data in the cloud.

Traceability: In case of any dispute occurs it can easily traceable. If other group member delete the other group members data can be easily noticeable.

6. THE PROPOSED SCHEME:

The group manager is responsible for system parameters generation, user registration, user revocation and traceability.

6.1 User Registration

User registered with their details such as identity (user name, password and email-id). Group manager select random number, base point , parameters and performs modulo with prime number, by using ECC (Elliptic Curve Cryptography) generate an private key. For registered users they will obtain private key, that private key is used for group signature and file decryption. The Group manager adds the user identity (ID) to the group user list that will be used in traceability phase.

6.2 User Revocation

User Revocation is performed by the group manager. Delta Revocation List is publicly available based on those, group members are allowed to encrypt the data and make that data confident against revoked users. Revoked users are maintained in the revoke user list and make publicly available in the cloud. Delta RL is bounded by signature to declare its validity. Upon receiving the resignation request from the group member, group member will be in revoked user list.

6.3 File Generation

Group members will store their data in real cloud. Aspose real cloud (SaaS) is provided by cloud service provider mainly for storage. The group members will request with group id and based on the Delta RL allow the data owner to upload the data in the cloud, if their signature is true. If it's a revoked user, cloud server will not allow generating the data and signature verification status false. When generating the data, hash id will be generated that will be used for deleting the data.

Data owner	File name	Hash id	Hash code	date
Name	name	F(∂)	C1,c2,c	t_{data}

6.4 File Access

To access the data that are stored in the cloud, group member will give request as group id, data id. Cloud server will verify their signature, if the group member in the same group then allow to access file. Group member have rights to access data, but not having rights to delete or modify the data that are stored in the cloud. If any request from revoked user, cloud server won't allow accessing the data.

6.5 File Deletion

File that are stored in the cloud can be deleted by either group member (i.e., the member who uploaded the file into the server) or by group manager. It allows data owners to delete their own files that are stored in the cloud. If any delete request from the group member, cloud server will verify the signature and delete the data file that are stored in the cloud.

6.6 Traceability

Group manager will reveal their real identity in case of any dispute occurs. If any malpractice happened inside the organisation it can be easily traceable. If any group members are modify or delete the data file of other groups, it can easily identify which member doing such activities.

6.7 Delta RLS

In existing RLS, revoked user details such as private key are updated manually for every day. Revoked users can access the cloud, hacking is possible. But in Delta RLS set a ttp value (threshold value), when it reaches the threshold value revoked users are updated automatically. Revoked users can't able to access the cloud hacking attack is reduced and communication overhead is also reduced.

7. RESULT ANALYSIS

Data generation and file access operations between Mona (RLS) and Delta RLS.

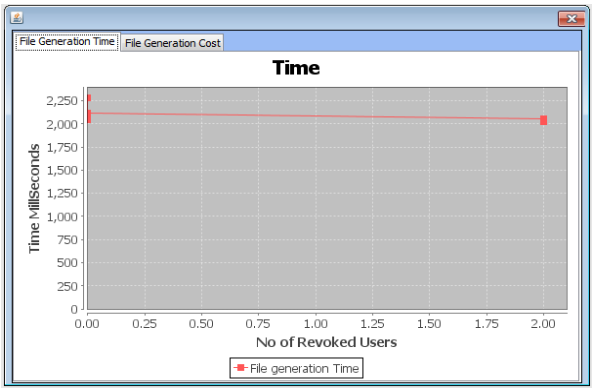

Fig 7.1 File Generation of RLS

In RLS, revoked users are updated for every one day. If any user revoked, they have an chance for accessing the cloud after the users revoked. In Figure 7.1 Three files are generated by cloud users and no users are revoked, after some time two files are generated and users are revoked. If any no. of users revoked they are updated at the end of the day.

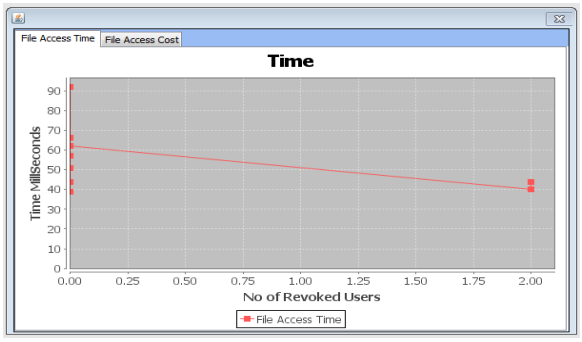

Fig 7.2 File access of RLS

In Figure7.2 number of files is accessed by cloud users and revoked users are updated.

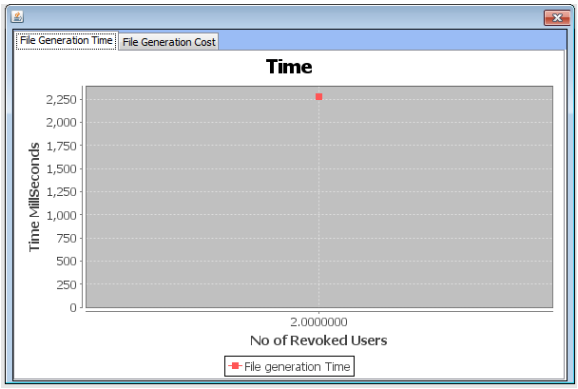

Fig 7.3 File generation of Delta RLS

In Delta RLS, revoked users are periodically updated. So there is no chance of accessing the cloud after the cloud users revoked. In Figure 7.2 one file is generated at the time of two users revoked.

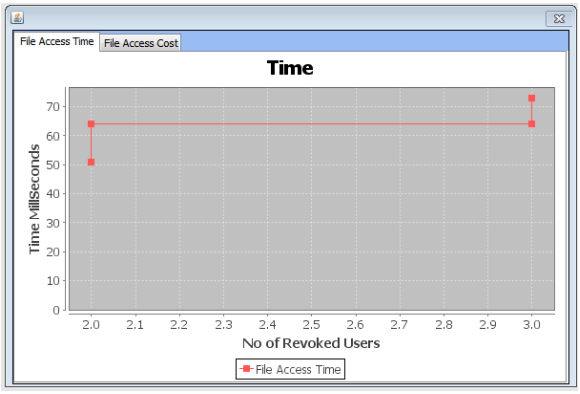

Fig 7.4 File access of Delta RLS

In Figure 7.4 two files are accessed at the time of two users revoked and revoked users are periodically updated. After some time again one user is revoked and two files are accessed at a time. When compared both RLS and Delta RLS, Delta RLS is more secure and no users are allowed to access the cloud after the users revoked.

8. CONCLUSION

In this paper, securely share the data file among the dynamic groups. Without revealing their identity members in the same group can share the data efficiently. Elliptic curve cryptography is used for over all security. When compared to other algorithm key size is very small, it is not able to hack easily. Delta RL is used for efficient revocation without updating private keys of remaining users. In future, concentrate on key management, how to revoke the private keys from the group members.

9. REFERENCES

[1] D. Boneh, X. Boyen, and H. Shacham, "Short Group Signature," Proc. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO),pp. 41-55, 2004.

[2] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO), pp. 213-229, 2001.

[3] D. Boneh, B. Lynn, and H. Shacham, "Short Signature from the Weil Pairing," Proc. Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology, pp. 514-532, 2001.

[4] D. Chaum and E. van Heyst, "Group Signatures," Proc. Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT), pp. 257-265, 1991.

[5] A. Fiat and M. Naor, "Broadcast Encryption," Proc. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO), pp. 480-491, 1993.

[6] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu "Plutus: Scalable Secure File Sharing on Untrusted Storage," Pro USENIX Conf. File and Storage Technologies, pp. 29-42, 2003.

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

[8] S.Kulkarni and Bezawada Bruhadeshwar, "Rekeying and Storage Cost for Multiple User Revocation," Department of Computer Science and Engineering, Michigan State University, East Lansing, MI48824USA.

[9] R. Lu, X. Lin, X. Liang, and X. Shen, "Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing," Proc. ACM Symp Information, Computer and Comm. Security, pp. 282-292, 2010.

[10] D. Naor, M. Naor, and J.B.Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers," Proc. Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO), pp. 41-62, 2001.

[11] B. Waters, "Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization," Proc. Int'l Conf. Practice and Theory in Public Key Cryptography Conf. Public KeyCryptography, http://eprint.iacr.org/2008/290.pdf, 2008.

[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.