A Random Access Multi CSP for Dynamic Group Algorithm for Group Shared Data in Cloud Computing

K.T. Archana[1], R. Nithya[2], C. Jayakumar[3], T. Dharanika[4]
[1,2,3,4] *Assistant Professor, Department of Computer Science and Engineering, Narasu`s Sarathy Institute of Technology, Salem*

Abstract- **A secure way for key distribution without any secure communication channels, and the users can securely obtain their private keys from group manager. Second, scheme can achieve fine-grained access control, any user in the group can use the source in the cloud and revoked users cannot access the cloud again after they are revoked. Protect the scheme from collusion attack, which means that revoked users cannot get the original data file even if they conspire with the untrusted cloud. In our approach, by leveraging polynomial function, we can achieve a secure user revocation scheme. Finally, our scheme can achieve fine efficiency, which means previous users need not to update their private keys for the situation either a new user joins in the group or a user is revoked from the group. We propose a random access multi csp for dynamic group algorithm for overcoming the above drawback. Through this the data guarding will be ensured and the Qos will be maintained. The attackers will find difficult to gather server details because of changing the cloud server frequently. This improves networking performance and secure data from being attacked.**

I. INTRODUCTION

Cloud computing, with the characteristics of intrinsic data sharing and low maintenance, provides a better utilization of resources. In cloud computing, cloud service providers offer an abstraction of infinite storage space for clients to host data. It can help clients reduce their financial overhead of data managements by migrating the local managements system into cloud servers. However, security concerns become the main constraint as we now outsource the storage of data, which is possibly sensitive, to cloud providers. To preserve data privacy, a common approach is to encrypt data files before the clients upload the encrypted data into the cloud . Unfortunately, it is difficult to design a secure and efficient data sharing scheme, especially for dynamic groups in the cloud.

II.RELATED RESEARCH

Kallahalla et al. presented a cryptographic storage system that enables secure data sharing on untrustworthy servers based on the techniques that dividing files into filegroups and encrypting each file_group with a file-block key. However, the file-block keys need to be updated and distributed for a user revocation, therefore, the system had a heavy key distribution overhead. Other schemes for data sharing on untrusted servers have been proposed in . However, the complexities of user participation and revocation in these schemes are linearly increasing with the number of data owners and the revoke users.

Yu et al. exploited and combined techniques of key policy attribute-based encryption , proxy re-encryption and lazy re-encryption to achieve fine-grained data access control without disclosing data contents. However, the single-owner manner may hinder the implementation of applications, where any member in the group can use the cloud service to store and share data files with others.

Lu et al. proposed a secure provenance scheme by leveraging group signatures and cipher text-policy attribute based encryption techniques . Each user obtains two keys after the registration while the attribute key is used to decrypt the data which is encrypted by the attribute-based encryption and the group signature key is used for privacy preserving and traceability. However, the revocation is not supported in this scheme.

Liu et al. presented a secure multi-owner data sharing scheme, named Mona. It is claimed that the scheme can achieve fine-grained access control and revoked users will not be able to access the sharing data again once they are revoked. However, the scheme will easily suffer from the collusion attack by the revoked user and the cloud. The revoked user can use his private key to decrypt the encrypted data file and get

the secret data after his revocation by conspiring with the cloud.

In the phase of file access, first of all, the revoked user sends his request to the cloud, and then the cloud responds the corresponding encrypted data file and revocation list to the revoked user without verifications.

Next, the revoked user can compute the decryption key with the help of the attack algorithm. Finally, this attack can lead to the revoked users getting the sharing data and disclosing other secrets of legitimate members.

Zhou et al. Achieve efficient user revocation that combines role-based access control policies with encryption to secure large data storage in the cloud. Unfortunately, the verifications between entities are not concerned, the scheme easily suffer from attacks, for example, collusion attack. Finally, this attack can lead to disclosing sensitive data files.

Zou et al. presented a practical and flexible key management mechanism for trusted collaborative computing. By leveraging access control polynomial, it is designed to achieve efficient access control for dynamic groups. Unfortunately, the secure way for sharing the personal permanent portable secret between the user and the server is not supported and the private key will be disclosed once the personal permanent portable secret is obtained by the attackers.

Nabeel et al. proposed a privacy preserving policy based content sharing scheme in public clouds. However, this scheme is not secure because of the weak protection of commitment in the phase of identity token issuance.

III.PROBLEM DEFINITION

We propose a random access multi csp for dynamic group algorithm for overcoming the above drawback. Through this the data guarding will be ensured and the qos will be maintained. The attackers will find difficult to gather server details because of changing the cloud server frequently. This improves networking performance and secure data from being attacked.

A. Advantages

1. Attackers find difficult to gather details from the server.
2. Attacker may mislead and confused when the server is changed.
3. Different servers possess different operational protocols which cannot be easily identifiable.

IV. SYSTEM DESIGN

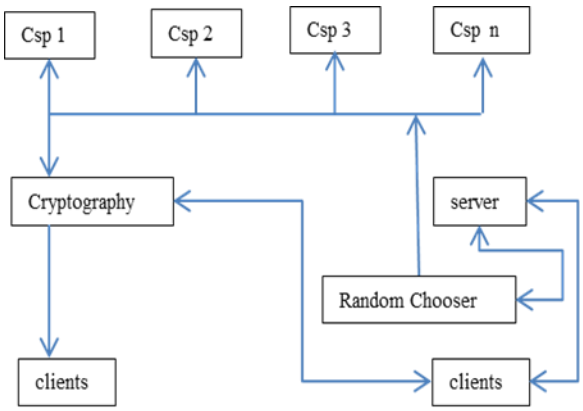

Fig. 1. Block Diagram

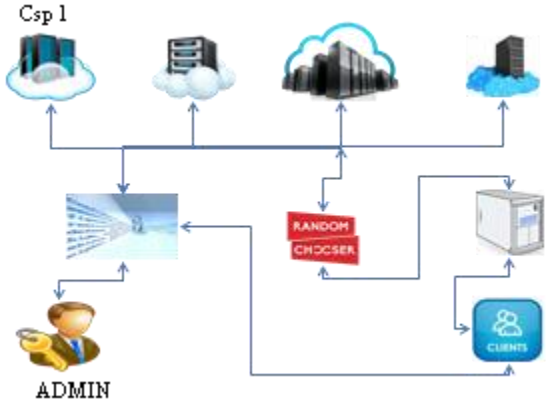

Fig. 2. Architectural Diagram

V.IMPLEMENTATION

1) Cloud server creation
2) Cryptography implementation
3) Dynamic cloud storage
4) Random access multi csp for dynamic group algorithm

A. Cloud server creation

Cloud server is created and the client settings are done so that they can access file from the server. The files will be stored by the admin and this will be retrieved and used by the clients.

B. Cryptography implementation

The file stored by the admin will be encrypted and stored. These files will be encrypted using asymmetric algorithm. The elliptic curve cryptography algorithm provides public key and private key. The private key generated will be sent to the user via email.

C. Dynamic cloud storage

The files stored will be transferred to other cloud storage servers in periodic interval of time. This process will be done by changing the service providers regularly. The copy of data will be there in another cloud to provide qos and fault tolerance.

D. Random access multi csp for dynamic group algorithm

Finally the data fetched from the cloud providers will be done in a random manner depending upon the algorithms choice. The duplicated data from other cloud will also be provided to the user randomly.

E. Cloud server creation

Cloud server is created and the client settings are done so that they can access file from the server. The files will be stored by the admin and this will be retrieved and used by the clients.

F. Cryptography implementation

The file stored by the admin will be encrypted and stored. These files will be encrypted using asymmetric ecc algorithm. The elliptic curve cryptography algorithm provides public key and private key. The private key generated will be sent to the user via email.

G. Dynamic cloud storage

The files stored will be transferred to other cloud storage servers in periodic interval of time. This process will be done by changing the service providers regularly. The copy of data will be there in another cloud to provide qos and fault tolerance.

H. Random access multi csp for dynamic group algorithm

Finally the data fetched from the cloud providers will be done in a random manner depending upon the algorithms choice. The duplicated data from other cloud will also be provided to the user randomly

VI.CONCLUSION

We have provided secured storage through alternating and storing data in dynamic way. The users can securely obtain their private keys from group manager Certificate Authorities and secure communication channels. Also, our scheme is able to support dynamic groups efficiently, when a new user joins in the group or a user is revoked from the group, the private keys of the other users do not need to be recomputed and updated. Moreover, our scheme can achieve secure user revocation, the revoked users can not be able to get the original data files once they are revoked even if they conspire with the untrusted cloud.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A view of cloud computing,” Commun. ACM, vol. 53, no. 4, pp. 50–58, Apr. 2010.

[2] S. Kamara and K. Lauter, “Cryptographic cloud storage,” in Proc. Int. Conf. Financial Cryptography Data Security, Jan. 2010, pp. 136–149.

[3] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving secure, scalable, and fine-grained data access control in cloud computing,” in Proc. ACM Symp. Inf., Comput. Commun. Security, 2010, pp. 282–292.

[4] Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in Proc. ACM Conf. Comput. Commun. Security, 2006, pp. 89–98.

[5] R. Lu, X. Lin, X. Liang, and X. Shen, “Secure provenance: The essential of bread and butter of data forensics in cloud computing,” in Proc. ACM Symp. Inf., Comput. Commun. Security, 2010, pp. 282–292.

[6] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, “Improved proxy re-encryption schemes with applications to secure distributed storage,” in Proc. Netw. Distrib. Syst. Security Symp., 2005, pp. 29–43.

[7] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, “Plutus: Scalable secure file sharing on untrusted storage,” in Proc. USENIX Conf. File Storage Technol., 2003, pp. 29–42.

[8] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, "Sirius: Securing remote untrusted storage," in Proc. Netw. Distrib. Syst. Security Symp., 2003, pp. 131–145.