International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 14
November 2017

Data Sharing In Dynamic Groups Avoiding Anti-Collusion

Koritala Aleakhya & Dr. P. Mahipal Reddy
*Pg Scholar, Department of CNIS, Vaagdevi college of Engineering, Autonomous, Warangal.
** Assistant Professor, Department of CNIS, Vaagdevi college of Engineering, Autonomous, Warangal.
koritalaaleakhya@gmail.com

ABSTRACT: *According to the rapid growth and essentiality to ensure security in the cloud. In this paper, we propose a Secure AntiCollusion data sharing schema for dynamic groups in the cloud using identity-based encryption. Customers can achieve a thriving and moderated methodology for sharing information between individuals gathered in the cloud with low maintenance characters and low administration cost. Then, security certifications will be given to the sharing information files as they are outsourced. Due to the ongoing change in registration, sharing information while ensuring protection is still a test problem, especially for an unreliable cloud because of the attack by agreement. In addition, for existing plans, the security of the key dispersion depends on the secure communication channel, so again, having such a channel is a solid feeling and is difficult to practice. In this article, we propose a secure information sharing plan for elementary individuals. First, we offer a secure route for key dispersal without secure matching channels, and customers can safely acquire their private keys from the collection administrator. In addition, the plan can perform accurate access control, any client in the collection can use the source in the cloud and refused customers can no longer return to the cloud after being rejected. Third, we can protect the plan against deception attacks, which means that rejected clients can not get the first record of information regardless of whether they handle the untrusted cloud. In this methodology, using the polynomial capability, we can realize a protected client rejection plan. Keywords: Access control, privacy protection, key distribution, cloud computing,*

I. INTRODUCTION

Cloud computing, with the characteristics of intrinsic data sharing and reduced maintenance, allows a better use of resources. In cloud computing, cloud service providers offer an abstraction of infinite storage space for clients that host data [1]. It can help customers reduce their financial costs for data management by migrating the local management system to cloud servers. However, security issues are becoming the biggest constraint as we are now outsourcing potentially sensitive data storage to cloud providers. To preserve data confidentiality, a

common approach is to encrypt data files before clients download encrypted data to the cloud [2]. Unfortunately, it is difficult to design a secure and efficient data sharing scheme, especially for dynamic groups in the cloud. Kallahalla et al. [3] presented a cryptographic storage system that allows secure data sharing on unreliable servers based on the techniques of dividing files into groups of files and encrypting each group of files with a file block key. However, the keys of the file block must be updated and distributed for user revocation, therefore, the system has a significant additional cost of key distribution. Other schemas for sharing data on untrusted servers have been proposed in [4], [5]. However, the complexities of user participation and revocation in these systems increase linearly with the number of data owners and users revoked. Yu et al. [6] combined encryption techniques based on key policy attributes [7], proxy re-encryption, and lazy reencryption to gain access control to fine-grained data without disclosing the content of the data. However, the single-owner way can hinder application implementation, where any member of the group can use the cloud service to store and share data files with others. Lu et al. [8] proposed a secure provenance scheme by taking advantage of group signatures and encryption techniques based on encryption-policy attributes [9]. Each user gets two keys after registration while the attribute key is used to decrypt the data that is encrypted by attribute-based encryption and the group signing key is used for privacy and traceability preservation . However, revocation is not supported in this scheme. Liu et al. [10] presented a secure multi-proprietary data sharing scheme, named Mona. It is claimed that the system can perform accurate access control and that revoked users will no longer be able to access the share data once revoked. However, the system will easily suffer from collusion attack by the revoked user and the cloud [13]. The revoked user may use his private key to decrypt the encrypted data file and obtain the secret data after revocation by conspiring with the cloud. In the file access phase, first, the revoked user sends his request to the cloud, and then the cloud responds to the corresponding encrypted data file and the revocation list to the revoked user without verification. Then the revoked user can calculate the decryption key using the attack algorithm. Finally, this attack can cause revoked users to obtain sharing data and to disclose other secrets of legitimate members. Zhou et al. [14] presented a scheme of secure access control over encrypted data in cloud storage by invoking a role-based encryption technique. It is argued that the system can achieve effective user revocation that combines role-based access control policies with encryption to secure the storage of large

data in the cloud. Unfortunately, inter-entity checks are not involved, the system easily suffers from attacks, for example, a collusion attack. Finally, this attack can lead to the disclosure of sensitive data files. Zou et al. [15] presented a convenient and flexible key management mechanism for trusted collaborative computing. By exploiting the access control polynomial, it is designed to provide effective access control for dynamic groups. Unfortunately, the secure way to share the permanent personal secret between the user and the server is not supported and the private key will be disclosed once the mobile secret p Permanent staff will have been obtained by the attackers. Nabeel et al. [16] proposed a content sharing scheme based on a privacy policy in public clouds. However, this system is not secure due to the low protection of the commitment in the identity token issuance phase.

II. RELATED WORK

Distributed computing is the conveyance of PC benefits over the Internet. Regardless of whether they understand it or not, many individuals utilize distributed computing administrations for their own needs. Here, keeping up information protection and character classification is a genuinely troublesome undertaking in multi-inhabitant information sharing. In this article, we propose a safe multi-proprietor information sharing plan by exploiting bunch marking and utilizing dynamic communicate encryption procedures that all individuals can share and information with different clients. What's more, here, the quantity of clients repudiated is free of the cost of capacity per head and encryption. In this article, the primary objective is to guarantee information security and show the adequacy of our plan in tests [13]. Distributed computing is a rising registering worldview in which IT foundation assets are given as Internet administrations. As promising as it might be, this worldview presents numerous new difficulties for information security and access control when clients outsource touchy information to share it on cloud servers that are not in an indistinguishable space from information proprietors. . To protect the privacy of touchy client information against untrusted servers, existing arrangements normally apply cryptographic techniques by uncovering the unscrambling keys just to approved clients. In any case, in doing as such, these arrangements definitely present an overwhelming registering overhead on the information proprietor for key circulation and information administration when fine granularity information get to control is wanted, and along these lines not develop well. The issue of at the same time acquiring the precision, adaptability, and classification of

access control information is as yet uncertain [6]. In the distributed computing condition, putting away delicate information is a more troublesome errand. The cost of secrecy is high when we scramble whole delicate information. Encryption information does not function admirably in the cloud application. It has turned into the test to safeguard delicate information in the cloud. We in this manner dissect the information that must be scrambled and the others are most certainly not. And furthermore separate the information into various parts and put away in an alternate cloud condition. Each piece of the datasets contains the tokens. The capacity server distinguishes the information utilizing emblematic keys. This enables protection to save information assaults from assailants [5]. Security and protection are real worries in the reception of cloud advances for information stockpiling. One way to deal with moderate these worries is the utilization of two-layer encryption (TLE) which incorporates coarse granularity and fine granularity get to control encryption. Be that as it may, in this approach, information proprietors acquire high correspondence and figuring costs. To beat this issue, the information proprietor makes encryption identified with protection; while the Trusted Third Party (TTP) performs fine re-encryption on exclusive scrambled information that tackles this issue by utilizing limit based access control with TTP to guarantee that legitimate clients get to the outsourced information. In this article, we have proposed an encryption strategy at TTP level to ensure the protection and respectability of outsourced information in the cloud condition [16].

III. PROPOSED METHOD

Symmetric Key Management The first and oldest, based on key administration engineering of the 1970s, uses similar information encryption innovation to oversee keys and scramble information. In these frameworks, called "symmetric key" frameworks in light of the fact that a similar key is used to encode and decrypt data, the key manager produces another key for each message at the request of the sender. The key is stored in a database next to the list of collectors. At the time the beneficiary confirms, the key is retrieved from the database and the name of the collector is coordinated against the list of approved beneficiaries. In the case of everything looks, the descrambling key is sent to the receiver. Symmetric key systems have become the centerpiece of the inside just encryption and confirmation frameworks. Until now, Kerberos frameworks and Windows space controllers were based on symmetric key administration. The ability to quickly interpret key passwords and, to a large extent, the rapid

execution of symmetric key encryption computations make these frameworks attractive for internal applications that do not need to incorporate any external clients into the encryption process. High storage costs - Many symmetric keyframes, but not all, require that a database containing the key for each message be available in the framework. While some advocates of symmetric keyframes will require that this database not be a significant obstruction, this key database must be reproduced, descended and supervised for the most part. Since this database contains basic security data (to be specific, the keys), these expenses are magnified. High Availability Requirements - Because the sender must request a key for each key manager message, the key supervisor is committed to each encryption operation. This implies that the key administrator must be exceptionally accessible and that the size of the key manager will limit the size of the entire information or information encryption framework. It also tends to increase the impact of capacity needs. B. Public Key Infrastructure (PKI) Key Management In the mid-1980s, an evolution of digital developments led to new and important types of encryption computations. These calculations, called "public keys" or "asymmetric", use an alternative key to scramble the information that they use to decrypt the information. The well-known Diffie-Hellman and RSA calculations are the best-known cases of open key calculations. While they are unable to encode substantial amounts of information, unbalanced calculations are ideally suited to key monitoring because they can quickly scramble smaller sized items. The basic thinking behind using open key calculations to monitor mass encryption keys is that a recipient produces two keys: an open key and a private key. To encode information, the sender creates a mass encryption key, scrambles the mass key with the recipient's open key, and sends the information next to the newly encrypted mass key. The recipient obtains the information, decrypts the mass key with his private key, and then uses that key to decode the data. On the surface, executives in the light of the key company's general base, or PKI, seem to understand the most pressing flaws of symmetric key frameworks: there is no requirement for a key database messages and the key server must not be reached for each message. Be that as it may, PKI has two notable obstacles to its ancestor, the symmetric key administration: 1) the creation of the private key at the beneficiary makes recovery of keys difficult to achieve and 2) the sender must find an open key for each beneficiary and confirms its legitimacy. Since the recipients create these keys themselves, the server will probably be unable to provide keys for all recipients. This failure to discover the

keys for each recipient has made the administration of scrambled messaging frames, for example PGP and S / MIME impossible to encrypt. . This is the problem that authentications had to illuminate.

IV. CONCLUSION AND FUTURE WORK

In this article, we design a secure anti-collusion data sharing scheme for dynamic groups in the cloud. In our schema, users can securely obtain their private keys from the group manager and secure communication channels. In addition, our system is able to effectively support dynamic groups, when a new user joins the group or a user is revoked from the group, the private keys of other users do not need to be recalculated and put up to date. In addition, our system can perform a secure user revocation, revoked users can not get the original data files once they are revoked even if they conspire with the unreliable cloud. In this article, I can use an identity-based encryption algorithm, but in the future more new secure encryption techniques will be used and revoked users may be available but they will not be able to obtain the original data files.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, Joseph AD, Katz R., Konwinski A., Lee G., Patterson D., Rabkin A., Stoica I., and Zaharia M., of cloud computing, ‖ Common. ACM, vol. 53, no. 4, pp. 50-58, April 2010.

[2] S. Kamara and K. Lauter, "Cryptographic Cloud Storage", in Proc. Int. Conf. Financial Cryptography Data Security, January 2010, pp. 136-149.

[3] Mr. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, -Plutus: Scalable Secure File Sharing on Unreliable Storage, ‖ in Proc. USENIX Conf. File Storage Technol., 2003, pp. 29-42.

[4] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, -Sirius: Securing remote unsecured storage, ‖ in Proc. Netw. Distribute Syst. Security Symp., 2003, pp. 131-145.

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

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

[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters, -Attribute-based Encryption for Fine Granular Access Control of Encrypted Data, in Proc. ACM Conf. Comput. Common. Security, 2006, pp. 89-98.

[8] Lu R., X. Lin, X. Liang, and X. Shen, -Safe Source: The Essence of the Bread and Butter of Legal Computing in Cloud Computing, ‖ in Proc. ACM

Symp. Inf., Comput. Common. Security, 2010, pp. 282-292.

[9] B. Waters, encryption based on the -Ciphertext-policy attribute: An expressive, efficient and surely secure realization, ‖ in Proc. Int. Conf. Theory of Practice Public Key Cryptography Conf. Public key cryptography, 2008, pp. 53-70.

[10] X. Liu, Y. Zhang, Wang B., and J. Yang, -Mona: Secure multi-proprietary data sharing for dynamic groups in the cloud, ‖ IEEE Trans. Distribute in parallel Syst., Vol. 24, no. 6, pp. 1182-1191, June 2013.

[11] D. Boneh, X. Boyen, and E. Goh, - Encryption based on hierarchical identity with a constant-size ciphertext, ‖ in Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn., 2005, pp. 440-456.

[12] C. Delerablee, P. Paillier, and D. Pointcheval, "Encryption of secure dynamic diffusion by integral collusion with Ci paradigms or decryption keys of constant size", in Proc. 1st Int. Conf. PairingBased Cryptography, 2007, pp. 39-59.

[13] Z. Zhu, Z. Jiang, and R. Jiang, -The attack on mona: secure multi-owner data sharing for dynamic groups in the cloud, ‖ in Proc. Int. Conf. Page 5841 Inf. Sci. Cloud Comput., December 7, 2013, pp. 185-189.

[14] L. Zhou, V. Varadharajan, and M. Hitchens, -Achieving access control based on secure role on encrypted data in cloud storage, ‖ IEEE Trans. Inf. Forensics Security, vol. 8, no. 12, pp. 1947-1960, dec. 2013.

[15] X. Zou, Y.-S. Dai, and E. Bertino, "A Practical and Flexible Key Management Mechanism for Trusted Collaborative Computing," in Proc. IEEE Conf. Comput. Commun., 2008, pp. 1211-1219.

[16] M. Nabeel, N. Shang, and E. Bertino, -Privacy preserving policy-based content sharing in public clouds, ‖ IEEE Trans. Know. Data Eng., Vol. 25, no. 11, pp. 2602-2614, Nov. 2013.

[17] D. Dolev and A. C. Yao, "On Public Key Protocol Security", IEEE Trans. Inf. Theory, vol. IT-29, no. 2, pp. 198-208, March 1983.

[18] B. Dan and F. Matt, "identity-based encryption from weil matching," in Proc. 21st Annu. Int. Cryptol. Conf. Adv. Cryptol., 2001, vol. 2139, pages 213-229.

[19] B. Den Boer, -Diffie-Hellman is as loud as discrete log for some prime numbers