International Journal of Scientific Research and Management (IJSRM)
||Volume||5||Issue||07||Pages||5833-5841||2017||
Website: www.ijsrm.in ISSN (e): 2321-3418
Index Copernicus value (2015): 57.47 DOI: 10.18535/ijsrm/v5i7.08

A Secure Anti-Collusion Data Sharing Scheme for Dynamic Groups in the Cloud by using Identity based Encryption

Mutyala Ramya Krishna[1], Vankamamidi S Naresh[2]

[1,2]Dept of CSE, Sri Vasavi Engineering College, Tadepalligudem, India

Abstract
As per the rapid growth and essentiality for providing security in cloud. In this we propose A Secure Anti-Collusion Data Sharing Scheme for Dynamic Groups in the Cloud by using Identity based Encryption. Clients can achieve a flourishing and moderate methodology for information sharing among gathering individuals in the cloud with the characters of low upkeep and little administration cost. Then, security certifications to the sharing information records will be given since they are outsourced. Due to the never-ending change of the enrolment, sharing information while giving protection saving is still a testing issue, particularly for an untrusted cloud because of the agreement attack. In addition, for existing plans, the security of key dispersion depends on the safe communication channel, then again, to have such channel is a solid feeling and is difficult for practice. In this paper, we propose a safe information sharing plan for element individuals. Firstly, we propose a safe route for key dispersion with no safe correspondence channels, and the clients can safely acquire their private keys from gathering administrator. Besides, the plan can accomplish fine-grained access control, any client in the gathering can utilize the source in the cloud and refused clients can't get to the cloud again after they are rejected. Thirdly, we can protect the plan from trickery attack, which implies that rejected clients can't get the first information record regardless of the possibility that they scheme with the untrusted cloud. In this methodology, by utilizing polynomial capacity, we can achieve a protected client denial plan.

***Keywords*:** Access control, privacy-preserving, key distribution, cloud computing,

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 [1]. 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 [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 enables secure data sharing on untrustworthy servers based on the techniques that dividing files into file groups 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 [4], [5]. However, the complexities of user participation and revocation in these schemes are linearly increasing with the number of data owners and the revoked users.

Yu et al. [6] exploited and combined techniques of key policy attribute-based encryption [7], 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. [8] proposed a secure provenance scheme by leveraging group signatures and ciphertext- policy attribute based encryption techniques [9]. 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 privacypreserving and traceability. However, the revocation is not supported in this scheme. Liu et al. [10] 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 [13]. 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, 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. [14] presented a secure access control scheme on encrypted data in cloud storage by invoking role-based encryption technique.

It is claimed that the scheme can 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. [15] 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. [16] 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.

II. Related Work

Cloud computing is the delivery of computing services over the Internet. Whether they realize it or not, many people use cloud computing services for their own personal needs. Here preserving the data privacy and identity privacy is somewhat difficult task in sharing a data in multi owner manner. In this paper we propose a secure multi owner data sharing schema by leveraging group signature and using dynamic broadcast encryption techniques any members can share and data with other users. And here numbers of revoked users are independent with the storage over head & encryption computation cost. In this paper the main goal is to provide the security for the data and demonstrate the efficiency of our schema in experiments[13].

Cloud computing is an emerging computing paradigm in which resources of the computing infrastructure are provided as services over the Internet. As promising as it is, this paradigm also brings forth many new challenges for data security and access control when users outsource sensitive data for sharing on cloud servers, which are not within the same trusted domain as data owners. To keep sensitive user data confidential against untrusted servers, existing solutions usually apply cryptographic methods by disclosing data decryption keys only to authorized users. However, in doing so, these solutions inevitably introduce a heavy computation overhead on the data owner for key distribution and data management when fine grained data access control is desired, and thus do not scale well. The problem of simultaneously achieving fine-grainedness, scalability, and data confidentiality of access control actually still remains unresolved[6].

In the cloud computing environment, storing sensitive data is more difficult task. The privacy preserve cost is high when we encrypt entire sensitive data. Also encrypt data are not performing well in cloud application. This is becomes the challenging to preserve the sensitive data in cloud. So we analyse the data which is need to be encrypted and other is not. And also split the data in different parts and stored it in different cloud environment. Each part of data sets are contains the tokens. The storage server identifies the data using token keys. This enables the privacy to preserve the data attacks from the attackers[5].

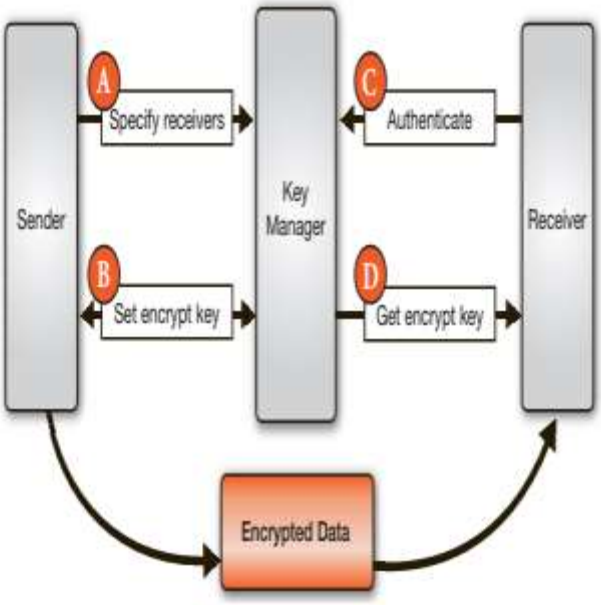
Security and privacy represent major concerns in the adoption of cloud technologies for data storage. An approach to mitigate these concerns is the use of Two Layer Encryption (TLE) which includes coarse-grained and fine grained access control encryption. But in this approach Data owners thus incur high communication and computation costs. To overcome this problem data owner performs a confidentiality related encryption; whereas the Trusted Third Party (TTP) performs a fine-grained re-encryption on top of the owner encrypted data which address this challenging issue using capability based access control with TTP to ensure valid users will access the outsourced data. In this paper, we proposed encryption method at TTP side to protect the privacy and integrity of outsourced data in cloud environment[16].

III. Proposed Method

A.Symmetric Key Management

The first and oldest—dating back to the 1970s—dedicated key management architecture uses the same data encryption technology to manage keys and scramble data. In these systems, called "symmetric key" systems because the same key is used to encrypt and decrypt information, the key manager generates a new key for every message at the sender's request. The key is stored in a database along with the list of receivers. When the receiver authenticates, the key is retrieved from the database and the receiver name is matched against the list of authorized recipients. If everything checks out, the decryption key is sent to the receiver.Symmetric key systems have become the centerpiece of internal-only encryption and authentication systems. Until recently, Kerberos systems and Windows domain controllers were based around symmetric key management. The ability to translate passwords readily into

keys and the extremely fast performance of symmetric key encryption algorithms make these systems attractive for internal applications that do not need to include any external users in the encryption process.



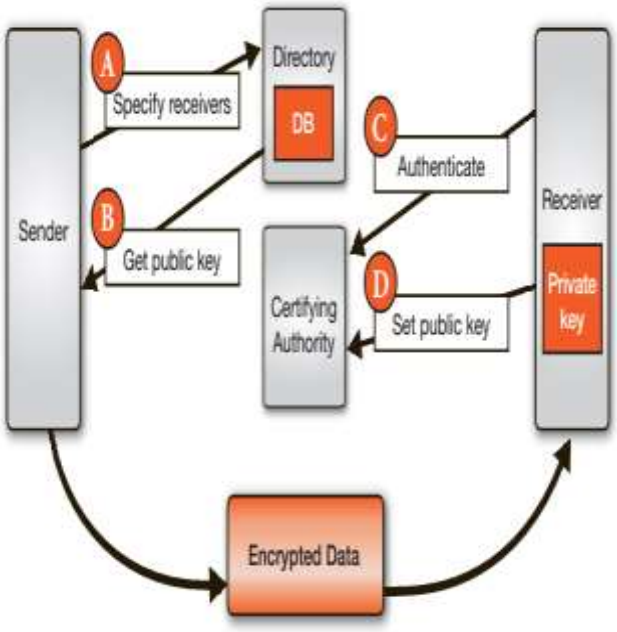
High Storage Costs– Many, but not all, symmetric key systems require that a database containing the key for every message is present in the system. While some proponents of symmetric key systems will insist that this database is not a significant impediment, this key database must be replicated, backed up, and generally managed. Because this database holds security critical information (namely, the keys), all of these costs are magnified.

High Availability Needs– Because the sender must request a key for each message from the key manager, the key manager is involved in every encryption operation. This means that the key manager must be highly available and the scale of the key manager will limit the scale of the entire messaging or data encryption system. This also tends to complicate the repercussions of the storage needs.

B.Public Key Infrastructure (PKI) Key Management

In the early 1980s, a series of mathematical innovations led to important new kinds of encryption algorithms. These algorithms, called "public key" or "asymmetric" systems use a different key to encrypt data than the one they use to decrypt data. The famous Diffie-Hellman and RSA algorithms are the best-known examples of public key algorithms. While they are ill-suited to encrypting large chunks of data, asymmetric algorithms are perfectly suited to managing keys, since they can readily encrypt smaller, key-sized objects. The essential idea behind using public key algorithms to manage bulk encryption keys is that a recipient generates a pair of keys: one public key and one private key. To encrypt data, the sender generates a bulk encryption key, encrypts the bulk key with the recipient's public key, and sends the data along with the newly encrypted bulk key. The recipient gets the data, decrypts the bulk key with his private key, and then uses that key to decrypt the data.On the surface, systems based on the public key infrastructure, or PKI, model seem to solve the most pressing flaws of symmetric key systems: there is no need for a per-message key database and the key server does not need to be contacted for each message. In reality, however, PKI has two significant disadvantages with respect to its predecessor, symmetric key management: 1) generating the private key at the recipient makes key recovery difficult to implement and 2) the sender must locate a public key for each recipient and verify its validity. Since the recipients generate these keys themselves, the server may not be able to supply keys for all recipients.This inability to find keys for every recipient made key management for encrypted email systems, such as PGP and S/MIME, impractical for encryption. This is the problem certificates were intended to solve. Certificates are authenticated data structures that tie a receiver's identity to a public key, without the need to request the key from a server. Because they are authenticated, certificates can be stored on distributed, untrusted directories, rather than the key server. This splits the key management server into a public facing directory and a certifying authority (CA). The CA is the only trusted component and the untrusted directory handles most of the transactions in the system, allowing for easier scalability

DOI: 10.18535/ijsrm/v5i7.08



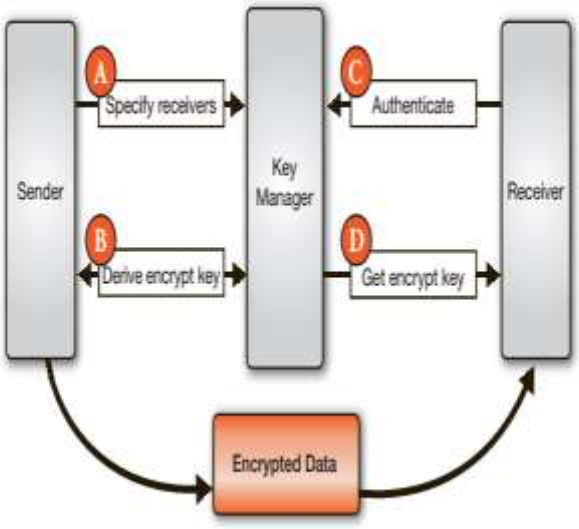
A New Approach to Key Management

Adi Shamir, one of the pioneers of public key cryptography, proposed a new type of public key algorithm in 1984. While public key systems have the inherent problem of distributing public keys and tying those public keys to a specific receiver, Shamir proposed mathematically generating the receiver's public key from his/her identity, then having the key server calculate the required private key. This system, called an "Identity-Based Encryption" (IBE) algorithm, would remove the need for public key queries or certificates. Because the key server generates the private key, key recovery no longer requires a separate private key database.

C. Identity-Based Encryption (IBE) system:

Using IBE radically simplifies key management because the sender does not need to contact the key server to get an encryption key. Instead, the encryption key is mathematically derived from the receiver's identity. The receiver must only contact the key server once to authenticate and get the required decryption key. The key server is able to construct the receiver's decryption key mathematically, eliminating the need for a database at the key server and making key recovery extremely straightforward. Since the sender does not need to contact the key server per unique recipient, encrypting information on a partner's key server is simple. The sender's policy can dictate which key server will be used to protect a message. That server can be one controlled by the sender's organization, one run by an outside service, or one located at the receiver's organization.

Figure 3: IBE diagram



IBE key management removes the need to communicate with a key server in an encryption transaction.

IV. Results and Analysis

IBE Uniquely Meets the Six Requirements of an Effective Key Management SystemThese technical innovations translate into tangible benefits when looking at the six requirements for an effective enterprise key management system. As encryption migrates from a server-to-server

and internal-only application to an information encryption utility, IBE becomes a natural choice for managing encryption keys because it is the only architecture that meets all six requirements of an effective key management system.

• Requirement 1:Deliver encryption keys. Using IBE, keys are always available for all recipients. The encryption key is derived mathematically from the receiver's identity. Groups are handled just as easily, because a key can be made from a group name as easily as an individual name.

• Requirement 2:Authenticate users and deliver decryption keys. IBE interfaces with existing authentication infrastructures, so any authentication resources that are already deployed (e.g., directories or web authentication) can be reused. The customer experience for getting a decryption key can be the same as logging into a portal, for example.

• Requirement 3:Jointly manage keys with partners. IBE enables the sender to select a local key server, a partner's key server, or a service to protect the data, depending on its particular requirements.

• Requirement 4:Deliver keys to trusted infrastructure components. Because IBE mathematically generates all keys at the server, theserver can securely regenerate keys for infrastructure components as needed.

• Requirement 5: Recover keys. In an IBE-based system, all keys are generated from a base secret stored at the key server. This base secret is backed up at server generation and, as long as the secret can be retrieved, any key can be securely regenerated.

• Requirement 6: Scale for growth. Without the need for databases that grow over time or requirements for per-transaction connections to the key server, IBE enables additional applications and transactions to be added with very little, if any, additional key management infrastructure.

REQUIREMENT	SYMMETRIC KEY MANAGEMENT	PKI	VOLTAGE IDENTITY-BASED ENCRYPTION (IBE)
1. Encrypt	Yes, online connection required.	Often no, when no recipient certificate is available.	Yes, to anyone including groups.
2. Decrypt	Yes, online connection required.	Yes.	Yes, without pre-enrollment required, including offline support.
3. Manage with partner	Yes, but must perform per encryption connection.	Yes, but must publish a directory externally.	Yes.
4. Integration with infrastructure	Yes, but requires a per-decryption lookup.	Not without complex key escrow and sharing.	Yes. No per message lookup or key escrow required.
5. Key recovery	Must maintain a key database.	Must maintain a key database.	Yes. No database required.
6. Scalability	Limited by per-transaction key server operations.	Limited by operational complexity.	Yes.

IV. CONCLUSION AND FUTURE WORK

In this paper, we design a secure anti-collusion data sharing scheme for dynamic groups in the cloud. In our scheme, the users can securely obtain their private keys from group manager 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. In this paper I can use identity based encryption algorithm but in future more and new secure encryption technique used and revoked users data may be available but they could not get the original data files.

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] 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.

[4] 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.

[5] 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.

[6] 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.

[7] V. 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.

[8] 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.

[9] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in Proc. Int. Conf. Practice Theory Public Key Cryptography Conf. Public Key Cryptography, 2008, pp. 53–70.

[10] X. Liu, Y. Zhang, B. Wang, and J. Yang, "Mona: Secure multi owner data sharing for dynamic groups in the cloud," IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 6, pp. 1182–1191, Jun. 2013.

[11] D. Boneh, X. Boyen, and E. Goh, "Hierarchical identity based encryption with constant size ciphertext," in Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn., 2005, pp. 440–456.

[12] C. Delerablee, P. Paillier, and D. Pointcheval, "Fully collusion secure dynamic broadcast encryption with constant-size Ci-phertexts or decryption keys," in Proc. 1st Int. Conf. Pairing-Based 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.

Inf. Sci. Cloud Comput., Dec. 7, 2013, pp. 185–189.

[14] L. Zhou, V. Varadharajan, and M. Hitchens, "Achieving secure role-based access control 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 the security of public key protocols," IEEE Trans. Inf. Theory, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.

[18] B. Dan and F. Matt, "Identity-based encryption from the weil pairing," in Proc. 21st Annu. Int. Cryptol. Conf. Adv. Cryptol., 2001, vol. 2139, pp. 213–229.

[19] B. Den Boer, "Diffie–Hellman is as strong as discrete log for certain primes," in Proc. Adv. Cryptol., 1988, p. 530.

[20] D. Boneh, X. Boyen, and H. Shacham, "Short group signature," in Proc. Int. Cryptology Conf. Adv. Cryptology, 2004, pp. 41–55.

About the Authors:

Mutyala Ramya Krishna is pursuing M.Tech in Sri Vasavi Engineering College in the branch of Computer Science and Engineering.

Dr. Vankamamidi Srinivasa Naresh is currently working as Associate professor in Sri Vasavi Engineering College. He obtained an M.Sc. in Mathematics from Andhra University, an M.Phil. in Mathematics from Madurai Kamaraj University and an M.Tech & Ph.D in Computer Science and Engineering from J.N.T.UK- Kakinada. He is also a recipient of U.G.C.-C.S.I.R. JUNIOR RESEARCH FELLOSHIP and cleared NET for LECTURERSHIP He completed UGC Minor Research Project with a financial assistance of Rs. 3,70,000.