IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 1, Ver. IV (Jan. 2014), PP 102-106
www.iosrjournals.org

Privacy Preserving and Load Balancing For Secure Cloud Storage

S. Sasikala, T. Karthick
M.E. (Computer Science and Engineering), Anand Institute of Higher Technology, Chennai,
Asst. Professor of Computer Science Dept, Anand Institute of Higher Technology, Chennai,

Abstract: *Using Cloud Storage, users can remotely store their data and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in Cloud Computing a formidable task. Unfortunately, sharing data in a multi-owner manner while preserving data and identity privacy from an untrusted cloud is still a challenging issue, due to the frequent change of the membership. In proposed a secure cloud storage for multi-owner data sharing authentication system supporting privacy-preserving by enabling public auditability for cloud storage with critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free. And also by encryption and hashing techniques, such as Advanced Encryption Standard (AES), Merkle Hash Tree, any cloud user can anonymously share data with others. By using these techniques the storage overhead and computation cost is reduced. Also trustworthiness will be increased between the user and the Cloud Service Provider. Load Balancing is also implemented to process the User requested job by allocating to the sub servers which will process the task by evaluating the CPU performance level.*

I. Introduction

Cloud Computing is recognized as an alternative to traditional information technology due to its intrinsic resource-sharing and low-maintenance characteristics. In cloud computing, the cloud service providers (CSPs), such as Amazon, are able to deliver various services to cloud users with the help of powerful datacenters. By migrating the local data management systems into cloud servers, users can enjoy high-quality services and save significant investments on their local infrastructures.

One of the most fundamental services offered by cloud providers is data storage. By utilizing the cloud, the users can be completely released from the troublesome local data storage and maintenance. However, it also poses a significant risk to the confidentiality of those stored files. Specifically, the cloud servers managed by cloud providers are not fully trusted by users while the data files stored in the cloud may be sensitive and confidential, such as business plans. To preserve data privacy, as basic solution is to encrypt data files, and then upload the encrypted data into the cloud. Unfortunately, designing an efficient and secure data sharing scheme for groups in the cloud is not an easy task due to the following challenging issues.

First, identity privacy is one of the most significant obstacles for the wide deployment of cloud computing. Without the guarantee of identity privacy, users may be unwilling to join in cloud computing systems because their real identities could be easily disclosed to cloud providers and attackers.

Second, it is highly recommended that any member in a group should be able to fully enjoy the data storing and sharing services provided by the cloud, which is defined as the multiple-owner manner. Compared with the single-owner manner, where only the group manager can store and modify data in the cloud, the multiple-owner manner is more flexible in practical applications. More concretely, each user in the group is able to not only read data, but also modify their part of data in the entire data file shared by the company.

Last but not least, groups are normally dynamic in practice, e.g., new staff participation and current employee revocation in a company. The changes of membership make secure data sharing extremely difficult. On one hand, the anonymous system challenges new granted users to learn the content of data files stored before their participation, because it is impossible for new granted users to contact with anonymous data owners, and obtain the corresponding decryption keys. On the other hand, an efficient membership revocation mechanism without updating the secret keys of the remaining users is also desired to minimize the complexity of key management.

Several security schemes for data sharing on untrusted servers have been proposed. In these approaches, data owners store the encrypted data files in untrusted storage and distribute the corresponding decryption keys only to authorized users. Thus, unauthorized users as well as storage servers cannot learn the content of the data files because they have no knowledge of the decryption keys.

However, the complexities of user participation and revocation in these schemes are linearly increasing with the number of data owners and the number of revoked users, respectively. By setting a group with a single attribute, Lu et al. proposed a secure provenance scheme based on the cipher text-policy attribute-based encryption technique, which allows any member in a group to share data with others. However, the issue of user revocation is not addressed in their scheme. Yu et al. presented a scalable and fine-grained data access control scheme in cloud computing based on the key policy attribute-based encryption (KP-ABE) technique. Unfortunately, the single owner manner hinders the adoption of their scheme into the case, where any user is granted to store and share data.

CLOUD COMPUTING

Cloud computing is a type of computing that relies on sharing computing resources rather than having local servers or personal devices to handle applications. It involves a large number of computers connected through a real-time communication network such as the Internet. It is a synonym for distributed computing over a network and means the ability to run a program on many connected computers at the same time.

DATA CONFIDENTIALITY

Data confidentiality is whether the information stored on a system is protected against unintended or unauthorized access. Since systems are sometimes used to manage sensitive information, Data confidentiality is often a measure of the ability of the system to protect its data. Accordingly, this is an integral component of security.

ACCESS CONTROL

Access control is a system which enables an authority to control access to areas and resources in a given physical facility or computer-based information system. It is the selective restriction of access to a place or other resource. The act of accessing may mean consuming, entering or using. Permission to access a resource is called authorization.

EFFICIENCY

A level of performance that describes a process that uses the lowest amount of inputs to create the greatest amount of outputs. Efficiency relates to the use of all inputs in producing any given output, including personal time and energy. It simply means reducing the amount of wasted inputs.

LOAD BALANCING

Load balancing is a technique to distribute workload evenly across two or more computers, network links, CPUs, hard drives, or other resources, in order to get optimal resource utilization, maximize throughput, minimize response time, and avoid overload of any one of the resources. Using multiple components with load balancing instead of a single component may increase reliability through redundancy.

The objective is to provide secure data storage, to maintain integrity of the data, to increase the user level of authentication and to improve the performance efficiently by 70-80% of balancing the load.

II. Related Work

In [6], Kallahalla et al. proposed a cryptographic storage system that enables secure file sharing on untrusted servers, named Plutus. By dividing files into file groups and encrypting each file group with a unique file-block key, the data owner can share the file groups with others through delivering the corresponding lockbox key, where the lockbox key is used to encrypt the file-block keys. However, it brings about a heavy key distribution overhead for large-scale file sharing. Additionally, the file-block key needs to be updated and distributed again for a user revocation.

In [5], files stored on the untrusted server include two parts: file metadata and file data. The file metadata implies the access control information including a series of encrypted key blocks, each of which is encrypted under the public key of authorized users. Thus, the size of the file metadata is proportional to the number of authorized users. The user revocation in the scheme is an intractable issue especially for large-scale sharing, since the file metadata needs to be updated. In their extension version, the NNL construction is used for efficient key revocation. However, when a new user joins the group, the private key of each user in an NNL system needs to be recomputed, which may limit the application for dynamic groups. Another concern is that the computation overhead of encryption linearly increases with the sharing scale.

Ateniese et al. [1] leveraged proxy reencryptions to secure distributed storage. Specifically, the data owner encrypts blocks of content with unique and symmetric content keys, which are further encrypted under a master public key. For access control, the server uses proxy cryptography to directly reencrypt the appropriate

content key(s) from the master public key to a granted user's public key. Unfortunately, a collusion attack between the untrusted server and any revoked malicious user can be launched, which enables them to learn the decryption keys
of all the encrypted blocks.

In [10], Yu et al. presented a scalable and fine-grained data access control scheme in cloud computing based on the KPABE technique. The data owner uses a random key to encrypt a file, where the random key is further encrypted with a set of attributes using KP-ABE. Then, the group manager assigns an access structure and the corresponding secret key to authorized users, such that a user can only decrypt a ciphertext if and only if the data file attributes satisfy the access structure. To achieve user revocation, the manager delegates' tasks of data file reencryption and user secret key update to cloud servers. However, the single owner manner may hinder the implementation of applications with the scenario, where any member in a group should be allowed to store and share data files with others.

Lu et al. [11] proposed a secure provenance scheme, which is built upon group signatures and ciphertext-policy attribute-based encryption techniques. Particularly, the system in their scheme is set with a single attribute. Each user obtains two keys after the registration: a group signature key and an attribute key. Thus, any user is able to encrypt a data file using attribute-based encryption and others in the group can decrypt the encrypted data using their attribute keys. Meanwhile, the user signs encrypted data with her group signature key for privacy preserving and traceability. However, user revocation is not supported in their scheme.

III. Proposed Work

To overcome this drawback, we propose secure storage for multi-owner data sharing authentication system in cloud. First, data will be uploaded in Encrypted format by the Data Owner in the Cloud Server. Once uploaded, the Cloud Server generates the Public and Private Keys. Then the data will be given to the Trusted Party Auditor for auditing purpose. The Auditor audits the data using Merkle Hash Tree Algorithm and stores in the Cloud Service Provider. If the user wants to View/Download the data, they have to provide the public key. The Data Owners will check the public key entered by the User. If valid, then the decryption key will be provided to the user to encrypt the data. The Load Balancing Concept is also implemented to process the User requested Job. First, the user request will be passed to the Cloud Service Provider's data center. The request is then queued up under the CSP's data center through communication channel. Then the job will be assigned to the sub server by keeping track of the CPU performance level that has minimum load.

3.1 ADVANTAGES

- By providing the Public and Private key components, only the valid user will be allowed to access the data.
- By allowing the Trusted party Auditor to audit the data, Trustworthiness will be increased between the User and Cloud Service Providers.
- By using Merkle Hash Tree Algorithm the data will be audited via multiple level of batch auditing Process.
- As Business Point of view, the Company's Customers will be increased due to the Security and Auditing Process.

3.2 OVERALL DESIGN

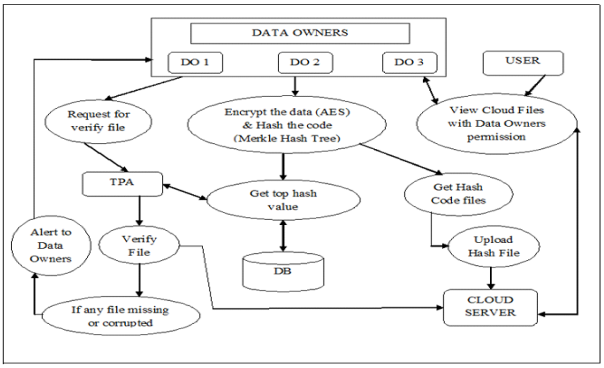

Fig. 3.2.1 Overall Process Design

3.3 ALGORITHM

3.3.1 Advanced Encryption Standard

Step 1: The original data is ciphered using Rijindael's key schedule.

Step 2: Round keys are derived from the cipher key known as Key Expansion. AES requires a separate 128-bit round key block for each round plus one more.
Step 3: Initial Round
Step 3.1: AddRoundKey – each byte of the state is combined with a block of the round key using bitwise XOR.
Step 4: Each round consist of several processing steps.
Step 4.1: SubBytes – a non-linear substitution step where each byte is replaced with another according to a lookup table.
Step 4.2: ShiftRows – a transposition step where each row of the state is shifted cyclically a certain number of steps.
Step 4.3: MixColumns – a mixing operation which operates on the columns of the state, combining the four bytes in each column.
Step 4.4: AddRoundKey
Step 5: Final Round will have only SubBytes, ShiftRows and AddRoundKey. MixColumns step is not performed in this round.

3.3.2 Merkle Hash Tree

Step 1: A file is split up into 'n' number of data blocks.
Step 2: Each data block is hashed and these hashes of data blocks are the leaves in hash tree.
Step 3: Nodes further up in the tree are the hashes of their respective children.
Step 4: Final hash value in a single node becomes a top hash value.

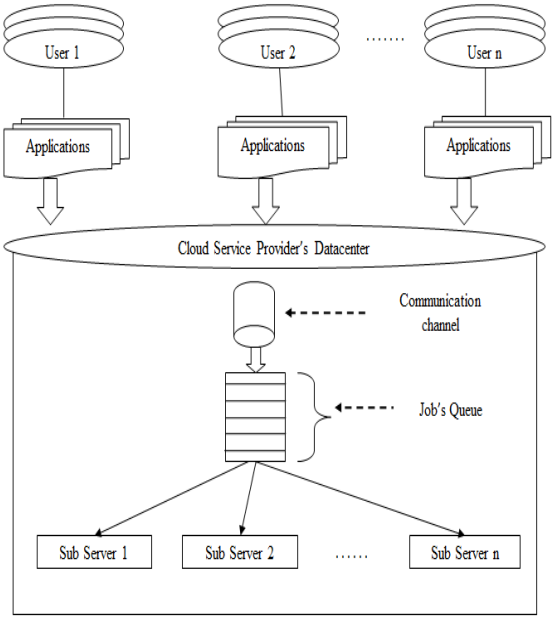

Fig. 3.2.2 Load Balancing Process Design

IV. Implementatiion

4.1 SECURED STORAGE MODULE

Once Data Owners registers in the cloud, private and public keys are generated for that registered owners. By using these keys, data owners can now store and retrieve data from cloud. A data owner encrypts the data using Advanced Encryption Standard (AES) and this encrypted data is then hashed with Merkle Hash Tree algorithm. By using Merkle Hash Tree algorithm the data will be audited via multiple level of batch auditing process. The top hash value is stored in local database and other hash code files are stored in cloud. Thus the original data cannot be retrieved by anyone from cloud, since the top hash value is not in cloud. Even if any part of data gets hacked, it is of no use to the hacker. Thus, the security can be ensured.

4.2 INTEGRITY CHECKER MODULE

To check whether the data is modified or not, that is present in cloud, data owner assigns a third party called Trusted Party Auditor (TPA). Once the data owner sends the request to audit the data, TPA checks the integrity of the data by getting the hash code files from cloud server and top hash value from db and verifies the file using Merkle Hash Tree Algorithm. After each time period, the auditing information will be updated by the

Trusted Party Auditor. If any file is missing or corrupted, email alert will be sent to data owner indicating that the data has been modified. The TPA can verify the file either by random or in manual way. Thus by allowing the Trusted Party Auditor to audit the data, Trustworthiness will be increased between the User and Cloud Service Providers.

4.3 SYSTEM AUTHENTICATION MODULE

In this module, the user is allowed to access the information from the Cloud Server. When a user registers in cloud, private key and public key will be generated for that user by cloud server. If user wants to view his own file, he uses private key. If user wants to view others file, he uses public key. This public key is split up equally for verification by data owners. Each part of the public key is verified by data owners. After verifying the key, if the key is valid, then user is allowed to access the data. If the key is invalid, then the user is rejected to access the data by Cloud Service Provider.

4.4 LOAD BALANCING MODULE

The users submit their diverse applications to the Cloud Service Provider through a communication channel. The requests from the users are queued up under the Cloud Service Provider's Data Center. The sub servers are then checked up for the minimum load with the CPU performance level of the currently executing task. The Cloud Service Provider then allocates the requested job to the sub servers that has minimum load to process the task in a First In First Out (FIFO) manner. Thus the User requested job will be assigned to the available sub server which contains minimum load and it is concerned to process the User requested job.

V. Summary And Conclusion

Data is secured by keeping the top hash value in local database and hash code files in Cloud Server. By enabling TPA to audit the data, integrity is maintained. Authenticating the requested user key by all data owners. Improving the overall performance by 70-80%.

References:

[1] G. Ateniese, K. Fu, M. Green, and S. Hohenberger (2005), "Improved Proxy Re- Encryption Schemes with Applications to Secure Distributed Storage", Proc. Network and Distributed Systems Security Symp. (NDSS), pp. 29-43.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores", proc. 14th ACM Conf. Computer and Comm. Security (CSS '07), pp. 598-609.

[3] K.D. Bowers, A. Juels, and A. Oprea (2009), "HAIL: A High-Availability and Integrity Layer for Cloud Storage", Proc. ACM Conf. Computer and Comm. Security (CCS '09), pp. 187-198.

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

[5] E. Goh, H. Shacham, N. Modadugu, and D. Boneh (2003), "Sirius: Securing Remote Untrusted Storage", Proc. Network and Distributed Systems Security Symp. (NDSS), pp. 131-145.

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

[7] X. Liu, Y. Zhang, B. Wang, and J. Yan (2013), "Mona: Secure Multi-Owner Data Sharing for Dynamic Groups in the Cloud", IEEE Trans. On Parallel and Distributed Systems, pp.1182-1191.

[8] H. Shacham and B. Waters (2008), "Compact Proofs of Retrievability", proc. Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (Asiacrypt), pp. 90-107.

[9] C. Wang, Q. Wang, K. Ren, and W. Lou (2013), "Privacy-Preserving Public Auditing for Secure Cloud Storage", IEEE Trans. on Computers, pp. 362-375.

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

[11] 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.