ISSN No:-2456-2165

Dynamic Outsourced Auditing Service for Cloud Storage

Kovidh Vashishth, Shiva D Nair

Abstract:- Data storage has today emerged as a new challenge with newer modes being introduced. Cloud computing is the most prevalent method by which large amount of data can be outsourced. The concept of Third Party Auditor (TPA) was introduced in order to rid the user from the tedious task of auditing their data. The TPA checks on behalf of the data owner so that he is assured of the veracity and authenticity of his data. But this Third Party Auditor scheme has its own set of loopholes as the basic assumption is that the TPA can be trusted and is honest, but where the TPA is dishonest and any two of the three involved entities (i.e. user, TPA and Cloud Service Provider) might be in an unholy alliance then the whole system collapses. The paper proposes a solution to protect the data stored in cloud from the dishonesty of TPA. The approach batch-verifies leaf nodes and their indexes together. The proposed scheme is also suitable for dynamic outsourced auditing system. It minimizes the costs of initialization for both user and TPA, and also incurs a lower rate for dynamism at user side.

Keywords:- *Cloud Computing, Third Party Auditor(TPA), Authencity.*

I. INTRODUCTION

Cloud Computing is the long dreamed vision of computing as a utility, where users can remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources. Simply put, cloud computing provides computing services such as servers, storage, databases, networking, software, analytics and more, over the Internet. Companies offering these computing services are called cloud providers and typically charge for cloud computing services based on usage. The most important feature of cloud computing is that it stores and manages data and files thus relieving the user from the hectic job of managing their own data. Once data is outsourced to the cloud storage by the data owner, autonomy over the outsourced data will be lost. Due to this data auditing becomes an important challenge for potential cloud users. It is impractical for data owners to keep verifying the integrity of the outsourced data continuously. To rid the user from frequent integrity verifications, auditing schemes are proposed. Third Party Auditor (TPA) is introduced in the public auditing scheme to perform verifications on behalf of user for data integrity assurance. But existing public auditing schemes rely on the assumption that TPA is trusted and will perform the auditing task honestly. But in reality that is not the case. That is because the three entities involved namely: the user, the TPA or the Cloud Service Provider (CSP), may be dishonest. More generally, TPA might even collude with CSP. Now if the TPA is dishonest or irresponsible, all current public auditing schemes will be broken, since these schemes cannot defend against the misconduct of the TPA. The paper proposes a scheme that provides solutions to the existing loopholes posed by the Third Party Auditor. To defend against the collusion of dishonest entities, and support dynamic updates simultaneously, dynamic outsourced auditing scheme will enable the user to authenticate their data and also support dynamic data storage.

II. RELATED WORK

The Cloud Security Alliance (CSA), in 2013, highlighted the nine major threats to cloud computing. These threats make auditing tasks very crucial in cloud computing. Many security measures were proposed. In 2010 a scheme was proposed to uniquely combine the public key based ho- momorphic authenticator with random masking to achieve the privacy-preserving public cloud data auditing system. The technique of index-hash table for auditing untrusted and outsourced storage was proposed a year later. Also, the MIST and the MALACHI algorithms were proposed in 2016 to strengthen the cloud against the looming threats. Also in that year, a scheme was adopted in which third party auditor will preprocess data on behalf of the cloud users before uploading them to cloud service providers and then verify data integrity. In 2017, a scheme named strong key exposure resilient auditing for secure cloud storage was proposed, in which the security of cloud storage auditing not only earlier than but also later than the key exposure can be preserved.

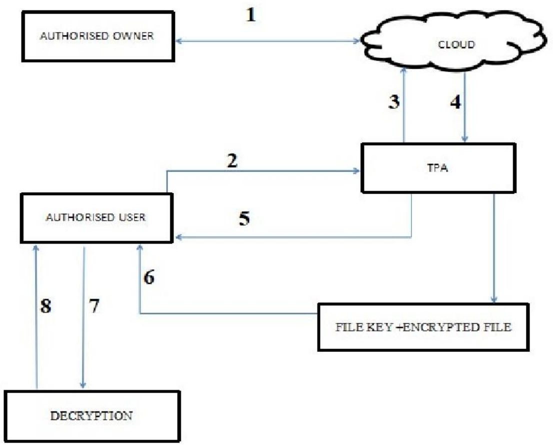
III. CHALLENGES INVOLVED

Once the data owners outsource their data to the cloud storage, the autonomy over the data will be lost. Hence the TPA perform verifications on behalf of user for data integrity assurance. But this was under the assumption that TPA can be trusted and will perform the auditing task honestly. But in reality the three entities involved namely: the user, the TPA or the Cloud Service Provider (CSP), may be dishonest. More generally, TPA might even collude with CSP.

The dynamic outsourced auditing schemes provides a solution to the drawbacks of the TPA. It simultaneously supports dynamic data storage. A new approach based on the idea of Merkle Hash Tree can be implemented. This new approach batch-verifies blocks and their own indexes all together as well as supports dynamic data storage. It also minimizes the costs of initialization for both user and TPA, and also will incur a lower price of dynamism at user side.

IV. ARCHITECTURE DIAGRAM OF PROPOSED SCHEME

Figure 1 represents the architecture diagram of the proposed model. The cloud or Cloud Service Provider (CSP) is an entity that provides various cloud services to the users that choose to avail them. Once users register in the cloud they will receive a unique CSP key for themselves. When they opt to store their personal data in cloud, their data are stored in the cloud server. The cloud servers maintain and manage these data. If another cloud user wants to access that data then they would have to send a request for that particular file to the Third Party Auditor (TPA). Now, the TPA was a model proposed in order to keep the authenticity and privacy of the data being stored. It basically gives a feedback to the cloud user about the security of data they have stored. It also accepts requests from other cloud users for accessing these files on behalf of the data owner. The data owner can accept or reject these requests when they access the cloud. The TPA puts the data owner's mind at ease by verifying the data stored. They also rid the user off the herculean task of constantly accessing the cloud server by acting on their behalf and performing various functions delegated to them by the owner. Once the TPA gets the request it searches for that particular file in the cloud as per request. The data stored in the cloud will be encrypted and will have a unique file key. The TPA searches and sends the encrypted file and the file key and asks for the CSP key from the user requesting the file. The user will enter the unique CSP key. With the help of the CSP key and the file key the file can be decrypted and the user can access it. The user can only view this data and not access it completely thus preventing a scenario where the stored data can be modified. This keeps the authenticity of the stored data intact.



- The first step involves storing the file in the cloud by the authorized file owner. The file will be encrypted using the AES algorithm and a unique file key will be generated for the particular file stored.
- This step involves the part where a cloud user will request a particular file stored in the cloud.
- The TPA will then search for the requested file in the cloud server.
- In this step the TPA receives the encrypted file and the unique file key for that particular file from cloud server.
- The TPA then requests for the CSP key(generated while registering to the cloud server) from the cloud user.
- The cloud user receives the encrypted file and the file key from the TPA.
- The encrypted file is decrypted using the file key and the CSP key.
- The plaintext generated can viewed by the cloud user.

V. ALGORITHM USED

A. *AES Algorithm*

AES algorithm is type of block cipher, it encrypts data in the form of blocks that are of a fix size. As all other encryption algorithms AES also uses key for encrypting the text, the size of these keys can be of 128,192 and 256 bits. For better encryption speed we can use smaller key size but if we need greater key size we have to use larger key size. We use this key for both encryption as well as decryption. The number of rounds in AES are not constant it depends on the key size. For a 128 bit key AES uses 10 rounds for 192, 12 rounds and for 256, 14 rounds. In encryption every round consists of four steps these are- byte substitution, shift rows, mix columns, and add round key. In the end of these rounds we get the ciphertext. In order to obtain the plain text from the ciphertext we have to follow these steps in every order but now in reverse order- add round key, mix columns, shift rows and byte substitution..

B. *Merkle Tree*

The algorithm below constructs a Merkle Tree from files in a given directory. The Merkle Tree in cryptography is a tree in which all leaf nodes are labelled with the hash of a data bock and every non-leaf node is labelled with the cryptographic hash of the labels of its child nodes. They allow sufficient and efficient verification of the contents of the large data structures. The concept of hash trees was patented by Ralph Merkle in 1979.

- *Algorithm*

Step 1. This class constructs a Merkle tree from files within a given directory.
Step 2. Add a method to flip the endian-ness of the hashes (to conform to Bitcoin).
Step 3. Add the trees being loaded from files.
Step 4. Figure out a proper indexing scheme for last two levels so tree doesn't have to be allocated as if it were complete.

Step 5. Add a file or directory to the list of tracked files in this tree if the file is not already.
Step 6. If file is a directory, specify whether or not the directory should be recursively searched.
Step 7. Make the merkle tree from the files in the specified directory and return the root hash after construction.
Step 8. Get the total number of tracked discrete files and directories.
Step 9. The list of directories or files being tracked may be empty.
Step 10. Check whether directory is being recursively tracked.
Step 11. Return true if directory is recursively tracked, false if not a directory or not recursively tracked.
Step 12. Remove file/directory from tracking.

VI. CONCLUSION

In the context of cloud storage and remote data auditing, how to defend against a dishonest TPA is an important issue raised by recent research. Compared to traditional public auditing schemes, outsourced auditing scheme under a stronger security model aims to protect against any dis- honest entity and collusion. The paper proposes a solution that will not only protect the data against dishonest TPA and collusion of entities but also support dynamic updates. The approach batch-verifies leaf nodes and their indexes. It is applicable for dynamic outsourced auditing system. It also minimizes the costs of initialization for both the cloud user and TPA, and also will incur a lower price of dynamism at user side. Future concept provide proxy based data auditing thus when the file in the cloud have been corrupted the proxy itself enhance protocol to change the corrupted file with original file present in the cloud.

REFERENCES

[1]. R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina, Controlling Data in the Cloud: Outsourcing Computation without Outsourcing Control, Proc. 2009 ACM Workshop on Cloud Computing Security (CCSW ’09), pp. 85-90, 2009.

[2]. Cloud Security Alliance (CSA), The Notorious Nine Cloud Computing Top Threats in 2013, https://cloudsecurityalliance. org/download/the-notorious-nine-cloud- computing-top -threats- in-2013, Feb. 2013.

[3]. G. Ateniese, R.C. Burns, R. Curtmola, J. Herring, L. Kissner, Z.N.J Peterson, and D.X. Song, Provable Data Possession at Untrusted Stores, Proc. 14th ACM Conf. Computer and Comm. Security (CCS 07), pp. 598-609, 2007.

[4]. A. Juels and B.S. Kaliski Jr, PORs: Proofs of Retrievability for Large Files, Proc. 14th ACM Conf. Computer and Comm. Security (CCS 07), pp. 584-597, 2007.

[5]. H. Shacham and B. Waters, Compact Proofs of Retrievability, Proc. 14th Intl Conf. Theory and Application of Cryptology and Informa- tion Security: Advances in Cryptology (ASIACRYPT 08), pp. 90-107, 2008.

[6]. Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing, IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 5, pp. 847-859, May 2011.

[7]. C.C. Erway, A. Kp, C. Papamanthou, and R. Tamassia, Dynamic Provable Data Possession, Proc. 16th ACM Conf. Computer and Comm. Security (CCS 09), pp. 213-222, 2009.

[8]. D. Cash, A. Kp, and D. Wichs, Dynamic Proofs of Retrievability via Oblivious Ram, Proc. 32nd Intl Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT 13), pp. 279-295, 2013.

[9]. C. Wang, S.S.M. Chow, Q. Wang, K. Ren, and W. Lou, Privacy- Preserving Public Auditing for Secure Cloud Storage, IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.

[10]. Y. Zhu, G.J. Ahn, H. Hu, S.S. Yau, H.G. An, and C.J. Hu, Dynamic Audit Services for Outsourced Storages in Clouds, IEEE Trans. Services Computing, vol. 6, no. 2, pp. 227-238, April-June 2013.