Batch Auditing for Privacy Preserving on Shared Data in the Cloud

K. Abirami
PG Scholar,
Easwari Engineering College
Chennai, India.

M. Kawya
PG Scholar,
Easwari Engineering College
Chennai, India.

P. Senthamarai
Assistant Professor.Sr.Gr
Easwari Engineering College
Chennai, India

***Abstract*- Cloud Computing is an emerging information technology infrastructure designed for rapid delivery of computing resources which provides services over the internet. The main aim of this paper is to ensure cloud storage correctness based on shared data integrity auditing mechanism. Here the identity of the signer on each block in shared data is kept private from public verifiers, who are able to efficiently verify shared data integrity without retrieving the entire file. In addition, this mechanism is able to perform multiple auditing tasks simultaneously instead of verifying them one by one. Privacy is given using RSA and Homomorphic Ring Signature Algorithm. In case if any attacks happen, auditing will be performed using String Matching Algorithm. This results in better effectiveness and efficiency for auditing shared data integrity.**

Keywords: Batch Auditing, Shared data, Privacy-preserving, Cloud Computing.

I. INTRODUCTION

Cloud computing involves deploying groups of remote servers and software networks that allow centralized data storage and online access to computer services or resources. Clouds can be classified as public, private or hybrid. services or resources. Clouds can be classified as public, private or hybrid. With cloud computing and storage, users are able to access and to share resources offered by cloud service providers at a lower marginal cost. It is routine for users to leverage cloud storage services to share data with others in a group, as data sharing becomes standard feature in most cloud storage offerings, including Dropbox, iCloud and Google Drive [1]. With cloud computing and storage, users are able to access and to share resources offered by cloud service providers at a lower marginal cost. It is routine for users to leverage cloud storage services to share data with others in a group, as data sharing becomes standard feature in most cloud storage offerings, including Dropbox, iCloud and Google Drive [1].Cloud service providers offer users efficient and scalable data storage services with a much lower marginal cost than traditional approaches [2].

The integrity of data in cloud storage is subject to skepticism and scrutiny, as data stored in the cloud can easily be lost or corrupted due to the inevitable hardware/software failures and human errors [3], [5]. To make this matter even worse, cloud service providers may be reluctant to inform users about these data errors in order to maintain the reputation of their services and avoid losing profits [6].

The traditional approach for checking data correctness is to retrieve the entire data from the cloud, and then verify data integrity by checking the correctness of signatures (e.g., RSA [8]) or hash values (e.g., MD5 [9]) of the entire data. Certainly, this conventional approachable to successfully check the correctness of cloud data. However, the efficiency of using this traditional approach on cloud data is in doubt [7].

Recently, many mechanisms [7] have been proposed to allow not only a data owner itself but also a public verifier to efficiently perform integrity checking without downloading the entire data from the cloud, which is referred to as public auditing [6]. Recently, many mechanisms [7] have been proposed to allow not only a data owner itself but also a public verifier to efficiently perform integrity checking without downloading the entire data from the cloud, which referred to as public auditing [6].

In these mechanisms, data is divided into many small blocks, where each block is independently signed by the owner; and a random combination of all the blocks instead of the whole data is retrieved during integrity checking [7].

II. RELATIVE WORK

B. Wang, B. Li, and H. Li proposed an "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud". This paper presents three approaches for data integrity. Keys like public and private key is generated by bilinear mapping. Encryption is performed by Hashing and Ring Signature for giving privacy. Block less Verifiable and Non-malleability is verified for data integrity. An iterative hash function breaks up a message into blocks of a fixed size and iterates over them with a compression function [1]. K. Ren, C. Wang, and Q. Wang, proposed a "Security challenges for the Public Cloud". This paper uses Fully Homomorphic Encryption for privacy preserving and access control is given only to authorized users. It does not preserve the malicious cloud users from abusing cloud resources [3].

We proposed an "AFS: Privacy-Preserving Public Auditing with data freshness in the cloud" To ensure freshness, it is necessary to authenticate not just data

blocks, but also their versions. Each block has an associated version counter that is incremented every time the block is modified. This version number is bound to the file-block's MAC: To protect against cloud replay of stale file-blocks (rollback attacks), the counters themselves must be authenticated [4]. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, proposed a "Provable Data Possession at Untrusted Stores". This paper tells how data is divided into small blocks, where each block is signed for privacy preserving from unauthorized user [7].

Paper Name	Issues
Oruta: Privacy-	Block less
Preserving Public	verifiability
Auditing for	
Shared Data in	
the Cloud	
Security	Does not preserve
challenges for the	the malicious
Public Cloud	cloud users
AFS: Privacy-	Does not provide
Preserving Public	traceability
Auditing with	
data freshness in	
the cloud	
Provable Data	Maliciously or
Possession at	Accidentally erase
Untrusted Stores	the data

III. PROPOSED SYSTEM

3.1 SYSTEM MODEL

In this paper, System model includes four parties: Cloud server provides services to the users for storing the data in the cloud. Security is provided by the cloud server for preserving the data stored in the cloud. Cloud owner creates the group in cloud and add users to that group for sharing the data. Owner gives either read or write permission to users for accessing the data. Uploaded data will be divided into blocks and stored dynamically in memory and references are stored in FAT file system.Fig.1.Public Verifier

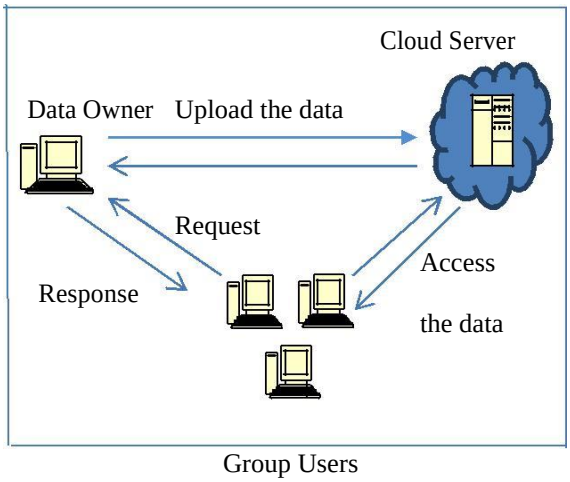

Fig.1 our system model shows privacy preserving for shared data.

Verifier will audit the records for performing data integrity without downloading the file. While downloading the file from cloud computation cost will be more. So, verifier audits the files without downloading. Signature is verified and related blocks are taken from the cloud and given to user without any loss in data. The data taken from the cloud is verified for correctness by auditing the files.

This paper is splitted into three modules,

1. Registration
2. Encryption
3. Signature Verification

3.2 ALGORITHM

3.2.1 RSA ALGORITHM

RSA Algorithm is used for key generation, which includes public and private key. The public key can be known by everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key. Key is generated by the following, Choose two very large random prime integers: p and q. Compute n and

$\varphi(n)$: $n = pq$ and $\varphi(n) = (p-1)(q-1)$. Choose an integer e, $1 < e < \varphi(n)$ such that: $\gcd(e, \varphi(n))=1$(where gcd means greatest common denominator). Compute d, $1 < d < \varphi(n)$ such that: $ed \equiv 1 \pmod{\varphi(n)}$.

Where, The public key is (n, e) and the private key is (n, d). The values of p, q and

$\varphi(n)$ are private. e is the public or encryption exponent. d is the private or decryption exponent. In Encryption, The cypher text C is found by the equation '$C = M^e \bmod n$' where M is the original message. In Decryption, The message M can be found form the cypher text C by the equation '$M = C^d \bmod n$' [10]. In our paper, Key is generated using RSA whereas, plain text is converted into cipher text.

3.2.2 HMAC ALGORITHM

A keyed-hash message authentication code (HMAC) is a specific construction for calculating a message authentication code (MAC) involving a cryptographic hash function in combination with a secret cryptographic key. As with any MAC, it may be used to simultaneously verify both the data integrity and the authentication of a message. Any cryptographic hash function, such as MD5 or SHA-1, may be used in the calculation of an HMAC; the resulting MAC algorithm is termed HMAC-MD5 or HMAC-SHA1 accordingly. The cryptographic strength of the HMAC depends upon the cryptographic strength of the underlying hash function, the size of its hash output, and on the size and quality of the key. An iterative hash function breaks up a message into blocks of a fixed size and iterates over them with a compression function [11]. Using this algorithm, we generate the hash value and use it as ring signature for privacy.

3.2.3 DATA SPLITTING

The encrypted cipher text in our proposed approach will be split up into n number of parts and then stored in cloud storage area. The different parts of file will be stored in different cloud servers as cipher text. The n value for number of storage denotes the number of cloud server exists where each different part of file will be stored. Thus the split files needs accurate information or the legitimate private key for integration of files again into single file. The data split up information will be stored in the local client system for security reasons.

For securing information using data splitting and integration includes access control of data makes communication so secure which protects private data that process inappropriate observations or modification from their intervention.

Split information is stored in FAT File System by having the references of split blocks and it is preserved from public auditor. While group user is downloading the entire file using private key, the entire data is received in correct order and is checked by batch auditing.

3.2.4 DATA INTEGRATION

When data need to be integrated this information will be reclaimed and all the data from each cloud server will be integrated in that particular order. Then by using the digital signatures the encrypted cipher text will be decrypted to readable text using private signature keys.

Number of file split ups according to number of servers will be taken a copy or back up into various database. This data backup does not affect the security of data as the two different parts of a single encrypted file which cannot be integrated with one another will be stored in a one particular server. If any one of the server gets crash down or any loss in related information of data under attack then those data will be recovered using this data backup.

After data retrieval they will be integrated according to the local information stored. Such integrated data will be decrypted then it will convert them into original plain text.

3.2.5 BATCH AUDITING

Auditing is performed for verifying data integrity whether the correct data is retrieved by the group users. Correctness is verified by string matching algorithm. Each character in the string is verified one by one by comparing with the copy it had. If any changes happen by hacker, human or technical errors this problem can be solved by auditing and replacing the incorrect data with correct data. String matching algorithm scans the text with the help of a window which size is generally equal to m. They first align the left ends of the window and the text, and then compare the characters of the window with the characters of the pattern - this specific work is called an attempt - and after a whole match of the pattern or after a mismatch they shift the window to the right. They repeat the same procedure again until the right end of the window goes beyond the right end of the text. This mechanism is usually called the sliding window mechanism. We associate each attempt with the position in the text when the window is positioned on y[j .. j+m-1] [12].

IV. CONCLUSION AND FUTURE WORK

In this public auditing of privacy preserving of shared data in cloud service the signatures for authenticators will ensure and audit shared data integrity without fetching the complete data. From our approach data will be secured by the user itself and not by any other third party auditing. This kind of third party auditing may also be an intruder or attacker of our system or data even in the cloud service. Data can be recovered from the cloud service even if there is any loss in particular server or data set or even when a server is completely crashed as we have n number of back up data in different servers which cannot even be integrated one another without any digital signatures.

There are two interesting problems we will continue to study for our future work. One of them is traceability, which means the ability for the group manager (i.e., the original user) to reveal the identity of the signer. Another problem is data freshness (prove the cloud possesses the latest version of shared data) while still preserving identity privacy.

REFERENCES

[1] B. Wang, B. Li, and H. Li,"Oruta: Privacy - Preserving Public Auditing for Shared Data in the Cloud", IEEE Transactions of Cloud Computing January-March 2014, Vol.2, No.1.

[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A.Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, andM. Zaharia, "A View of Cloud Computing", Comm.ACM,vol. 53, no. 4, pp. 50-58, Apr. 2010.

[3] K. Ren, C. Wang, and Q. Wang, "Security challenges for the Public Cloud", IEEE Internet Computing, vol. 16, no. 1, pp. 69–73, 2012.
[4] P.Maheswari, B.Sindhumathi, "AFS: Privacy-Preserving Public Auditing with data freshness in the cloud", IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p-ISSN: 2278-8727 PP 56-63.
[5] D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud Data Protection for the Masses", IEEE Computer, vol. 45, no. 1, pp. 39–45, 2012.
[6] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing", Proc. IEEE INFOCOM, pp. 525-533, 2010.
[7] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores", in Proceedings of ACM CCS'07, 2007, pp. 598–610.
[8] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", vol. 21, no. 2, pp. 120–126.
[9] The MD5 Message-Digest Algorithm (RFC1321). https://tools.ietf.org/html/rfc1321, 2014.
[10] The RSA Algorithm from en.wikipedia.org/wiki/RSA_ (cryptosystem).
[11] The HMAC Algorithm from en.wikipedia.org/wiki/Hash-based_message_authentication_code.
[12] The String Matching Algorithm from www-igm.univ-mlv.fr/~lecroq/string/node2.html#SECTION00 20.