International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 05 | May -2017 www.irjet.net p-ISSN: 2395-0072

THIRD PARTY VERIFICATION SCHEME FOR DATA INTEGRITY IN CLOUD

Shwetha Joshi[1], Asst. Prof. Veena A. Patil[2]

[1]M.Tech student, Computer Science and Engineering
BLDEA'S V.P. Dr. P.G. Halakatti College of Engineering and Technology, Vijayapur, Karnataka, India
[2]Assistant Professor in Computer Science and Engineering Department
BLDEA'S V.P. Dr. P.G. Halakatti College of Engineering and Technology, Vijayapur, Karnataka, India

---

Abstract – *Cloud storage services allow the users to outsource their data into cloud server to save local storage costs. Despite of advantages given by cloud server there exist security concern in data integrity. To check for data integrity client delegates this task to third party auditor who has expertise in computation. Our schema uses double hashing methodology to detect external adversaries' attacks namely birthday attack , length extension attack etc and random masking technique where file content will not be available with the auditor. Another difficulty in this approach is to check for trustworthiness of auditor which we have overcome with another mechanism. To do so we have calculated threshold time, if auditor doesn't send report within threshold time then he is a malicious one. Main objectives of checking data integrity are achieved by our proposed scheme.*

Key Words: **Data integrity, Birthday attack, Length extension attack, Botnet , DDoS attack, Random masking technique etc.**

1. INTRODUCTION

Cloud Computing is a delivery model in which a pool of resources are available to clients and they can access them via internet. Cloud computing and storage provided to users and enterprises having features to store and process their data in either privately owned, or third-party data centers that may be located far from the user are cost effective. The biggest advantage of cloud computing is cost effectiveness. It works on pay as you use format where in client associated with the cloud service provider will be charged based on services chosen and per hour cost as set by cloud service provider. This provides many services where in the most important one is storage i.e cloud provides facility for clients to keep their data into the cloud storage. Cloud storage is important because it reduces the burden on the client by maintaining the data intact and secured. Storage service given by cloud helps its users to manage their data efficiently and in flexible way without keeping a copy of data in their local system. Outsourcing of data to cloud storage servers is developing as a trend among many firms and users owing to its economic advantages [1]. Users today regularly access files without knowing or needing to know on what machines or in what geographical locations their files reside. Specifically, users can process their data on their PCs, outsource the processed data to cloud servers, and use the data on other devices (for example, mobile phones). The great convenience provided by such services is leading to a growing number of cloud storage providers [2]. Owing to the advantages of cloud storage, there exist difficulties in providing such a service to the user.

Difficulty which a client face is data resided in the cloud is whether intact and is it possible for an external advisory to breach the security of the cloud. This security concern is called data integrity. Cloud service provider may hide the data loss or may discard the data which is rarely used by the user. To overcome this difficulty many schemes were proposed. First scheme which was proposed is checking the data after very interval of time by the user for data consistency. The biggest disadvantage of this scheme turned to be the burden on the user to check for data integrity even if it the data is intact which results in computation overhead and in increase of communication costs. To reduce the verification burden another scheme was proposed where in an external and independent authority will periodically verifies data integrity on users' behalf.

2. RELATED WORK

Public verification techniques allow the users to outsource their data into the cloud and consistency Of the data is checked by a trusted third party called auditor. Objective of the public verification scheme is to avoid external adversary attacks on the data outsourced by the owner.

In [3], author proposed a scheme to remove burden on the user for checking the data integrity by assigning this task to a third party called auditor. It is assumed that auditor is trust worthy and honest in his task. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities toward user data privacy, and introduce no additional online burden to use. Here Homomorphic linear authenticator and random masking techniques are combined to achieve privacy preserving auditing scheme. In proposed protocol, the linear combination of sampled blocks in the server's response is masked with randomness generated by the server. With random masking, the TPA no longer has all the necessary information to build up a correct group of linear equations and therefore cannot derive the user's data content, no matter how many linear combinations of the same set of file blocks can be collected. The main disadvantage of this scheme is TPA is assumed to be trusted;

there may be chances of colluding TPA with the external adversary and falsely report the user that data is intact without running verification process.

In [4], author has discussed about the DDoS attack on the data dumped into the cloud and a prominent approach to prevent DDoS attack. DDoS attack occurs due to resource competition between defenders and attackers. Cloud resources are under control of provider and dynamic allocation of resources is done based on number of users requested. Individual cloud hosted server is still vulnerable to DDoS attack if they still run in traditional way. Hence author proposed a scheme where resources are allocated dynamically to counter DDoS attack against individual cloud customer. Author designed a strategy where cloud customers who are experiencing DDoS attacks are dynamically allocated with idle or reserved cloud resources and at the same time guaranteeing the quality of service for benign users so that DDoS can be defeated. Author proposes a queuing model in the form of M/M/m model to allocate the resources to users. Future works needed for this paper are to improve the queuing model from M/M/m to M/G/m and if a cloud data center runs out of resources during a battle.

In [5], author illustrates the work on modeling malicious activities from various perspectives. Botnets as malicious networks are discussed; botnet is a group of compromised computers on the Internet, controlled by botmasters through control and command centers. The random graph based modeling is not appropriate for cyber space which is one reason why it is difficult to win against cyber criminals. This paper presents the activities of malicious networks and algorithms to detect these activities.

In [6], author proposed a public privacy-preserving audit scheme which uses BLS signature with random sampling to verify data integrity in cloud.BLS signature involves two phases .One is setup phase and another one is audit phase. In setup phase, firstly key generation step will be conducted and then file to be uploaded into the cloud is divided into blocks. Hash value of each block is calculated using the keys generated in the previous step. Then file tag is computed by applying hash function on file name using a random number. Signature of each block is calculated and then file with the signature is sent to cloud. In audit phase, TPA sends challenge message to server for signature of file to check integrity of data.TPA having signature as sent by the user verifies with the signature as sent by the cloud server. Hash value computed once cannot be decoded that easily and another advantage is just by changing a single bit in the file a new hash value will be generated. Hence the scheme proves to be secured in checking data integrity. But the biggest disadvantage of proposed work is auditor's malicious activities will not be detected.

3. PROPOSED SYSTEM

Proposed system includes three entities namely Cloud Server, Client and Auditor. Client outsources his data to cloud and later access data when needed. Thereafter client delegates data integrity verification to the auditor, a trusted third party who has expertise in computation and verification. As shown in Fig-1 client uploads file to the cloud and sends request to auditor for verification.

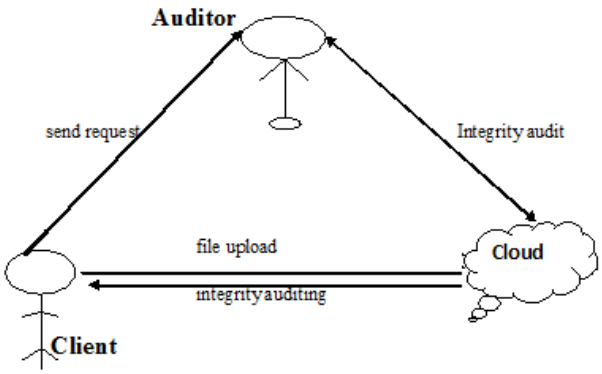

Fig-1: System Architecture of Proposed work

To check for existence of file in cloud during upload, a hash value of file is calculated by the client using MD5 hashing algorithm. MD5 algorithm processes variable length file and outputs a fixed sized message digest of 128 bits. The file to be uploaded is break down into 512 bit blocks and made sure that the size of file is dividable by 512. If not, padding procedure must be followed to make it dividable by 512. Padding procedure works as follows: firstly a single bit 1 is appended at the end of message and then many zeros are appended to bring it 64 bits less than multiple of 512 bits. The remaining 64 bits are filled by the length of the original message, modulo 2^{64}. The Algorithm operates in 128 bit chunks of four 32 bit words denoted by A, B, C, and D. These are initialized to some fixed constants. Processing of 512 bits block of file to be uploaded on cloud involves four similar rounds for each blocks of file. In each round 16 similar operations are preformed based on non linear function F. There are four possible functions for F; a different one is used in each round. Functions are as follows [7]

$$F(B,C,D) = (B \wedge C) \vee (\neg B \wedge D)$$
$$G(B,C,D) = (B \wedge D) \vee (C \wedge \neg D)$$
$$H(B,C,D) = B \oplus C \oplus D$$
$$I(B,C,D) = C \oplus (B \vee \neg D)$$

$\oplus, \wedge, \vee, \neg$ denotes XOR, AND, OR and NOT operations respectively.

Our main objective is to detect attacks of external adversaries and detecting malicious auditor activities .To do so we will be using double hashing strategy to overcome external adversaries attacks namely birthday attack, length extension attack etc. Another hashing algorithm SHA-256 is

used to give more security to the data. It is applied to the message digest previously determined by the MD5 algorithm.

SHA-256 algorithm is hashing technique it works as follows. Firstly it converts file to be dividable by 512 bit block using padding procedure same as MD5[8]. Boolean operations AND, XOR and OR, denoted by $\wedge$, $\oplus$ and $\vee$, respectively. Bitwise complement, denoted by $\neg$. Integer addition modulo 2^{32}, denoted by A + B. Each of them operates on 32-bit words. For the last operation, binary words are interpreted as integers written in base 2. RotR(A, n) denotes the circular right shift of n bits of the binary word A. ShR(A, n) denotes the right shift of n bits of the binary word A. A||B denotes the concatenation of the binary words A and B.The algorithm uses the functions:

$Ch(X, Y, Z) = (X \wedge Y) \oplus (X \wedge Z)$
$Maj(X, Y, Z) = (X \wedge Y) \oplus (X \wedge Z) \oplus (Y \wedge Z)$
$\Sigma_0(X) = RotR(X, 2) \oplus RotR(X, 13) \oplus RotR(X, 22)$
$\Sigma_1(X) = RotR(X, 6) \oplus RotR(X, 11) \oplus RotR(X, 25)$
$\sigma_0(X) = RotR(X, 7) \oplus RotR(X, 18) \oplus ShR(X, 3)$
$\sigma_1(X) = RotR(X, 17) \oplus RotR(X, 19) \oplus ShR(X, 10)$
and the 64 binary words K_i given by the 32 first bits of the fractional parts of the cube roots of the first 64 prime numbers.

For each block $M \in \{0, 1\}$ 512, 64 words of 32 bits each are constructed as follows: The first 16 are obtained by splitting M in 32-bit blocks $M = W_1||W_2|| \dots ||W_{15}||W_{16}$ .The remaining 48 are obtained with the formula: $W_i = \sigma_1(W_{i-2}) + W_{i-7} + \sigma_0(W_{i-15}) + W_{i-16}$ $17 \le i \le 64$.

First, eight variables are set to their initial values, given by the first 32 bits of the fractional part of the square roots of the first 8 prime numbers. Next, the blocks $M^{(1)}$, $M^{(2)}, \dots, M^{(N)}$ are processed one at a time
$(a, b, c, d, e, f, g, h) = (H_1^{(t-1)}, H_2^{(t-1)}, H_3^{(t-1)}, H_4^{(t-1)}, H_5^{(t-1)}, H_6^{(t-1)}, H_7^{(t-1)}, H_8^{(t-1)})$

Do 64 rounds consisting of:
$T_1 = h + \Sigma_1(e) + Ch(e, f, g) + K_i + W_i$
$T_2 = \Sigma_0(a) + Maj(a, b, c)$
$h = g$
$g = f$
$f = e$
$e = d + T_1$
$d = c$
$c = b$
$b = a$
$a = T_1 + T_2$
After this compute the new value of $H_j^{(t)}$
$H_1^{(t)} = H_1^{(t-1)} + a$
$H_2^{(t)} = H_2^{(t-1)} + b$
$H_3^{(t)} = H_3^{(t-1)} + c$
$H_4^{(t)} = H_4^{(t-1)} + d$
$H_5^{(t)} = H_5^{(t-1)} + e$
$H_6^{(t)} = H_6^{(t-1)} + f$
$H_7^{(t)} = H_7^{(t-1)} + g$
$H_8^{(t)} = H_8^{(t-1)} + h$

The hash of the message is the concatenation of the variables $H_i^{(N)}$after the last block has been processed $H = H_1^{(N)} ||H_2^{(N)}||H_3^{(N)} ||H_4^{(N)} ||H_5^{(N)} ||H_6^{(N)} ||H_7^{(N)} ||H_8^{(N)}$ .After applying double hashing methodology, the original file is encrypted using RSA and sent to cloud. RSA is an asymmetric encryption algorithm where public key is kept for encryption and private key for decryption [9].

Choose two prime numbers, p and q. From these numbers we can calculate the modulus, n=pq. Select a third number, e, that is relatively prime to (i.e. it does not divide evenly into) the product (p-1)(q-1) , the number e is the public exponent. Calculate an integer d from the quotient(ed-1)/(p-1)(q-1) The number d is the private exponent. The public key is the number pair (n,e).Although these values are publicly known, it is computationally infeasible to determine d from n and e if p and q are large enough. To encrypt a message, M, with the public key, creates the cipher-text, C, using the equation: C=M^e Mod n. The receiver then decrypts the cipher-text with the private key using the equation: M=C^d Mod n. Auditor uses random masking to check for data integrity where auditor will be given with hash code of file not with the file content.

After uploading of file if the client wants to check consistency of file whether the file has been modified or not, at this point of time client sends a integrity checking request to auditor. Auditor getting a request from client prepares a challenge message which consists of signature as sent by client and id to cloud server. Cloud server checks the authentication of auditor by checking signature with one which is already stored in cloud. If auditor passes in authentication then hash value of the file requested is sent to auditor. Auditor now checks it's integrity by comparing hash value sent by client with cloud server's hash value. Then if there is change in data that will be notified to client with report containing the details like bucket name i.e name of the storage provided by the cloud, user name ,file name ,accessed date time, operation performed and client IP address. This makes external adversaries' actions detectable but in real scenario auditor may collude with cloud server or with any attacker to change confidential data. So, to avoid this we proposed a technique where in we have set a threshold value within that if auditor sends report of verification then the auditor will be considered as not a malicious auditor or else he is a malicious auditor and report will not be generated at client side.

4. RESULTS AND ANALYSIS

Client first creates a storage space on cloud with a name and uploads file within that space. After that when client wants to check integrity just sends audit call to auditor as shown in Fig.

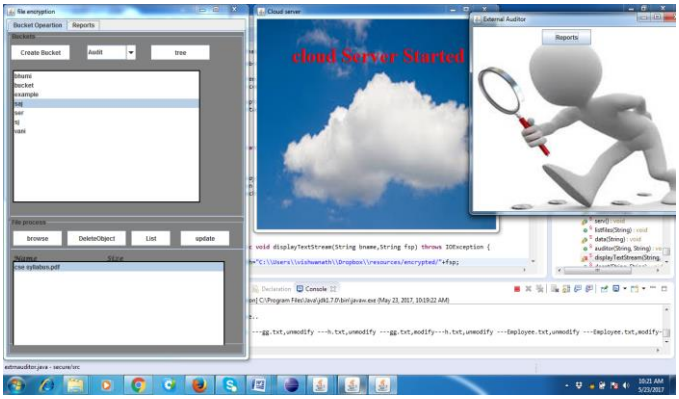

Fig-2: Uploading and auditing of file

If the file is been modified by the attacker then a report will be sent back to the client as shown in Fig- 3.

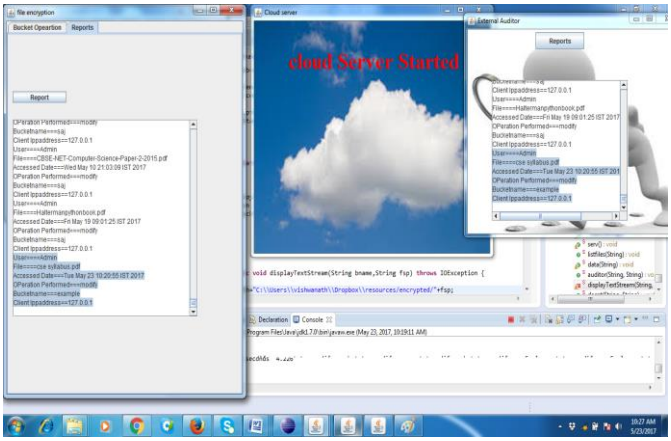

Fig-3: Report sent to client having file modified notification

If auditor gives the report after the scheduled threshold value then the report will not be generated at client side with a pop up message report is not verified by trusted auditor as shown in Fig-4.

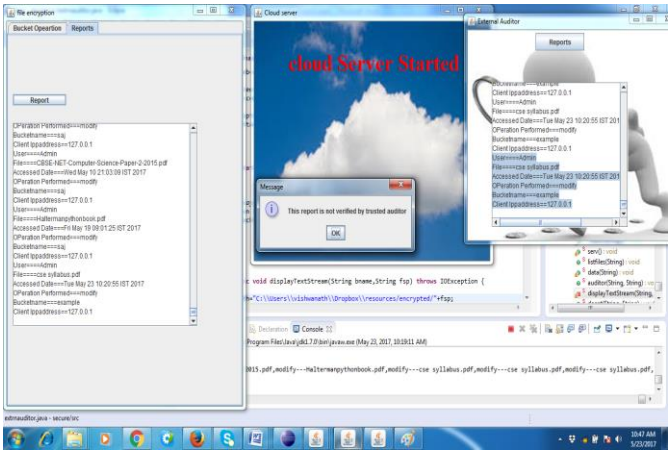

Fig-4: Malicious auditor's activity

5. CONCLUSION

Proposed third party verification scheme delivers the main objectives of the data integrity check. Double hashing methodology and random masking technique discussed above gives opportunity to detect external adversaries' actions without revealing real content of the file to the auditor. Procedure is established to check for malicious auditor activities. Hence proposed scheme strengthens the storage service provided by the cloud server.

REFERENCES

[1] Arsalan Iqbal , Hina Saham ,"Data Integrity Issues in Cloud Servers" IJCSI International Journal of Computer Science Issues, Vol. 11, Issue 3, No 1, 118-121 May 2014

[2] Yuan Zhang, Chunxiang Xu, and Hongwei Li , "Cryptographic Public Verification of Data Integrity for Cloud Storage Systems" IEEE Cloud Computing published by the IEEE computer society IEEE September /October 44-52 2016

[3] Cong Wang, Sherman S.M. Chow, Qian Wang, Kui Ren and Wenjing Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage" IEEE Transactions On Computers, VOL. 62, NO. 2, February 362-375 2013

[4] Shui Yu, Yonghong Tian, Song Guo and Dapeng Oliver Wu, "Can We Beat DDoS Attacks in Clouds?" IEEE Transactions on Parallel And Distributed Systems, VOL. 25, NO. 9, September 2245-2254 2014

[5] S. Yu, G. Wang, and W. Zhou, "Modeling Malicious Activities in Cyber Space," *IEEE Network*, vol. 29, no. 6, pp. 83–87 2015.

[6] Hong-Chun Jiang, Chao-Sheng Feng , Ding Yuan "Enabling Public And Privacy-Preserving Auditability For Cloud Storage" Proceedings of the 2016 International Conference on Machine Learning and Cybernetics, Jeju, South Korea, 10-13 July, 24-28 2016

[7] https://en.wikipedia.org/wiki/MD5

[8] https://www.researchgate.net/file.

[9] Nentawe Y. Goshwe, "Data Encryption and Decryption Using RSA Algorithm in a Network Environment", IJCSNS International Journal of Computer Science and Network Security, VOL.13 No.7, July 9-13 2013