IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 3, March 2021

DOI 10.17148/IJARCCE.2021.10319

Cloud Data Auditing using Hashing Algorithm

Prof.Biju Balakrishnan[1], Arathi U[2], Harihar S[3], Joshitha M[4], Muhammed Shahinsha T K[5]

Department of Computer Science and Engineering, JCT College of Engineering and Technology

Coimbatore,Tamil Nadu, India[1-5]

Abstract: Cloud computing is an important part of any small or large organization. With cloud storage service users can remotely store their data to the cloud and realize the data sharing with others. Data outsourcing possess the risk of sensitive data getting breached. Remote data integrity auditing is proposed to guarantee the integrity of data stored in the cloud. In some common cloud storage systems such as Electronic Health Records (EHR) systems, the cloud file might contain some sensitive information which must not be altered. This project proposes a novel privacy-preserving mechanism that supports auditing on shared data stored in the cloud. In particular, hashing algorithm are used to compute verification metadata needed to audit the correctness of shared data. A signature is generated for each file by the data owner and then it is uploaded. After a file is uploaded to the database by the data owner, the auditor re-computes it's signature and compares with the local copy signature of the file in data owner side. If suppose an intruder or a user breaches the security and modifies the public cloud data, the Hashing algorithmisperformed, and the intruder or user is terminated from further accessing of the cloud data. This method ensures data integrity in shared cloud data particularly for EHR systems, much more efficient and less computation cost compared to the existing system. As a result, the data in the cloud are reliable and authentic.

Keywords: Cloud storage; Hashing Algorithm Auditing mechanism; Data Integrity; Storage Auditing

I. INTRODUCTION

In Today's Technological world, Cloud storage is an important service of cloud computing [9].Organizations produce a huge volume of sensitive data; the speed of data generation increases and overtakes the storage capacity of the organizations [10]. Particularly, there is an increase in the number of EHR (Electronic Health Record) System files in the hospital sector. Therefore, more hospital sector would like to store their data in the cloud, which reduces the heavy burden of storing them in their unzipped format and further it can be easily accessed by the researcher from remote location. However, there is a chance for the data stored in the cloud to get corrupted or lost due to software bugs, hardware faults, intruders and human errors in the cloud. This paper proposes an efficient RDC (Remote Data Checking) method which audits the integrity of the data in cloud using Hashing algorithm.

II. PROBLEM ANALYSIS

A signature is a string of a few bytes intended to identify uniquely the contents of a data object. Different signature proves the inequality of the contents while same signatures indicate their equality. Objects are signed using 4 bytes instead of 20-bytes standard SHA-1 which leads to more frequent collisions[1]. In this hashing algorithm technique, MD-5 of 32-byte is used which has lesser probability for collision.

Auditing of the documents on the remote side is important. But data originator checks only chunks of the storage server for verification [2]. In this project auditing after update scheme is used to verify all the updated documents in the public cloud storage to ensure data integrity and protection from alternation,

Remote data checking (RDC) schemes allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The data is representation as a file spilt in blocks and is distributed among multiple servers [4,10]. RDC for distributed system becomes more costly since it is harder to ensure the correctness of algorithms, especially operation during failures of part of the system and recovery from failures that does not arise in centralized system such as this project.

III. RELATED WORK

Hashing Algorithm

STEP:1

A) Assign 64 constant values to array constant X. Assign Initial hash value for H. Now convert the string message into 512 blocks using the following equations. $L = \frac{Message\ Length}{4} + 2$

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 3, March 2021

DOI 10.17148/IJARCCE.2021.10319

B) Number of 16-integer (512 bit) block required

$$N = ceil\frac{L}{16} = \lceil \frac{L}{16} \rceil$$

C) Message M is N x 16 array of 32-bit integer

$$Array\ of\ M = M\ [\ N \times 16\ ]$$

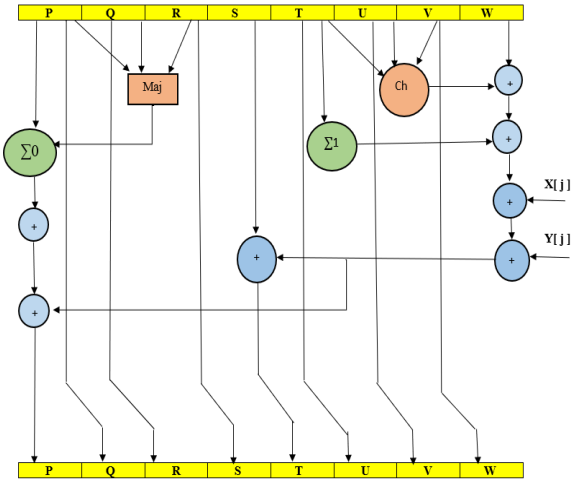


Fig. 1. Single iteration in SHA-256 compression

STEP:2 Encode 4 character per integer (64 per block), big-endian encoding

STEP:3 Hash Computation

STEP:3.1 Prepare message schedule ' Y '

$$Y_0^{\{16\}}(j) = M(i)(j)$$

$$Y_{16}^{\{64\}}(j) = \sigma_0(\,Y[\,j-2\,]) + Y[\,j-7\,] + \sigma_0(\,Y[\,j-15\,]) + Y[\,j-16\,]$$

Where,

$$\sigma_0(c) = ROTR^7\,(c) \oplus ROTR^{18}\,(c) \oplus SHL^3\,(c)$$

$$\sigma_1(c) = ROTR^{17}\,(c) \oplus ROTR^{19}\,(c) \oplus SHL^{10}\,(c)$$

STEP:3.2 Initialize the buffer p, q, r, s, t, u, v, w with previous hash value. The below function is a rotation function

$$\sum_0^{\{256\}}(c) = ROTR^2\,(c) \oplus ROTR^{13}\,(c) \oplus ROTR^{22}\,(c)$$

$$\sum_1^{\{256\}}(c) = ROTR^6\,(c) \oplus ROTR^{11}\,(c) \oplus ROTR^{25}(c)$$

This is the main looping function which loops from 0 < j < 64

$$B1 = W + \sum\nolimits_1(t) + ch(t,u,v) + X[\,j\,] + Y[\,j\,];$$

$$B2 = \sum\nolimits_0(p) + maj(p,q,r);$$

$$w = v;$$

$$v = u;$$

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 3, March 2021

DOI 10.17148/IJARCCE.2021.10319

$$v = t;$$

$$t = (s + b1);$$

$$s = r;$$

$$r = q;$$

$$q = p;$$

$$p = (B1 + B2);$$

STEP :3.3 Compute Intermediate hash value

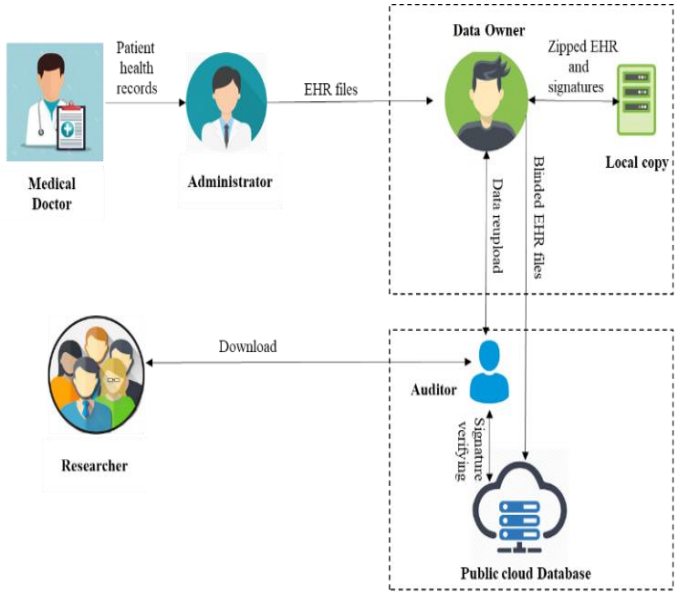
$$H[i\ ] = H[\ i\ ] + g$$

Where g represents the working buffers where $i \rightarrow 0 \leq i \leq 7$

STEP:4 Convert the hash value to hex strings with leading zeroes. Convert hex bytes to string of characters

IV. PROPOSED SYSTEM

The Scheme‘s main algorithm is the Auditing using Hashing signature which is implemented and works between the public cloud and local data



Public Cloud- Public cloud is an on-demand computing services provided by third-party organisation over the public internet. Anyone the place where the Patient's EHR files are stored by encrypting the sensitive data. It provides ease of access to files for the users. They can access anywhere at anytime.

Local Data-This is situated at the Administrator side where all the EHR files of the patient are stored in the local disk. The Data present here are much more safer without any corruption.

EHR Files- Electronic Health Record Files consists of details of the patients and their history of disease which tells diseases from which they have been affected. The most sensitive information includes Patients details like Name, Age, Address, Phone number and etc. Remaining Part of the EHR files consists of Patient’s Blood group, X-ray Scans details, diseases which affected the patient, Symptoms which occurred during the spread of disease, Name and Dosage of medicine given to patient at the advancement of disease.

Hashing Signature- this is our proposed system which uses signature verification to prevent data corruption and to enable data integrity of the EHR files stored in public cloud. Data Owner before uploading the EHR files to the public cloud, the files is hashed using SHA-2 algorithm which uses 256 bits for Hashing. Hashed 256 bits value is stored in the local disk in a table directory.When the sensitive data in the EHR files are encrypted and stored in the same table directory.When the Auditor detects any upload or modification, then hash value of the file is calculated. It is compared

IJARCCE

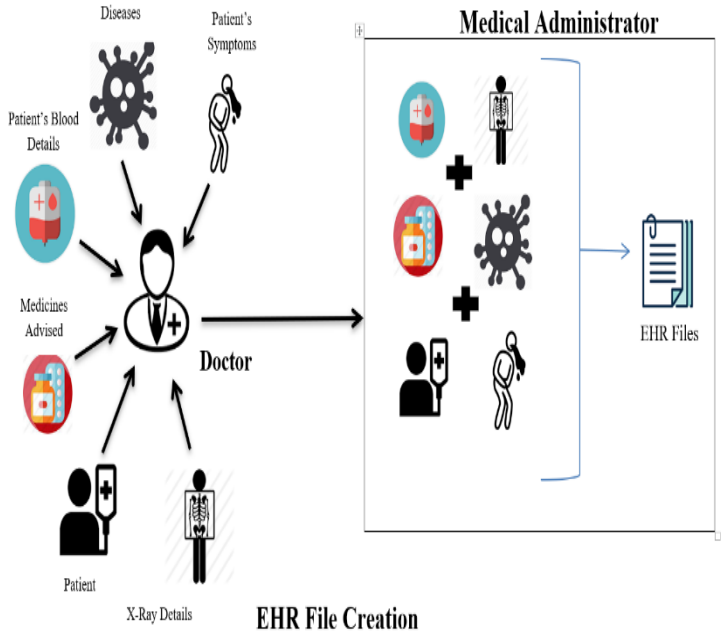
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

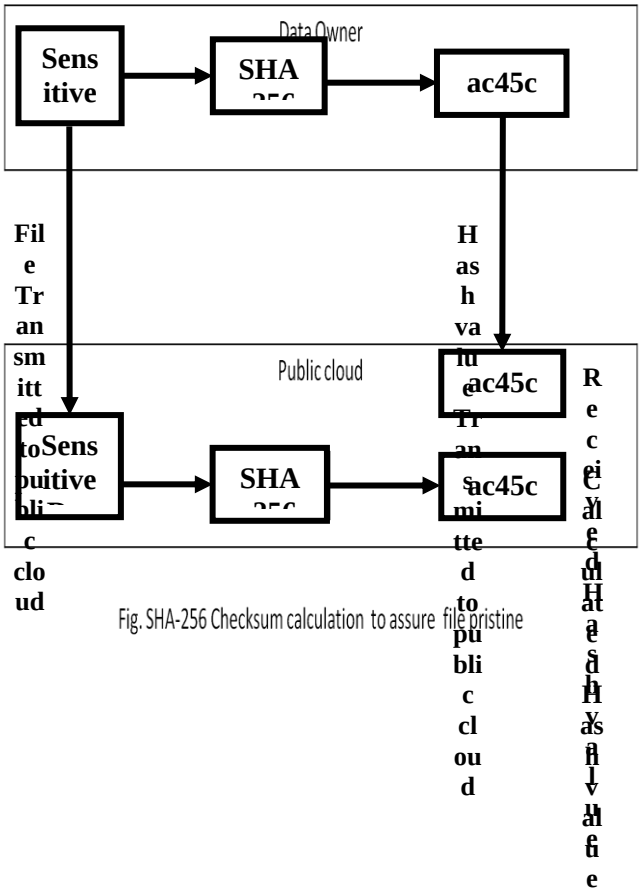
Vol. 10, Issue 3, March 2021

DOI 10.17148/IJARCCE.2021.10319

with the data owner's local disk for its match. If matched, then the file is marked as the original file.If not,local copy of the file is again uploaded to the public cloud.



- Auditor – here auditor is an automated entity



IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 3, March 2021

DOI 10.17148/IJARCCE.2021.10319

V. EXPERIMENTAL EVALUATION

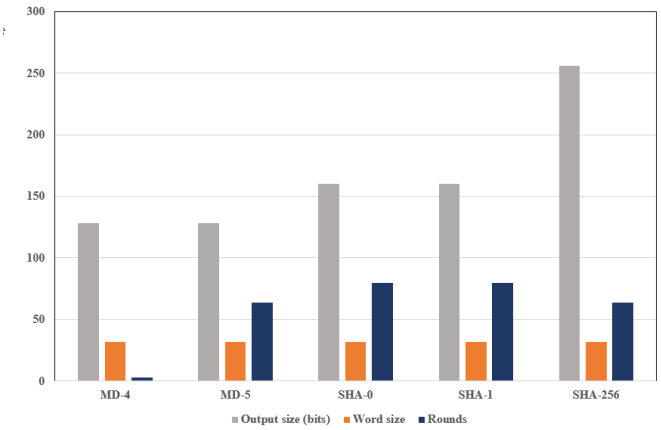


Fig. 2. Parameters comparison

VI. FUTURE ENHANCEMENT

The length extension attack can degrade the security mechanism of SHA-256 which makes it easy for finding the data in the file with the bit length and the hashed value of the output. So, in near future we would like to introduce the SHA-3 Security Mechanism which counter measures the Length Extension Attack.

CONCLUSION

In this paper, we proposed an identity-based data integrity auditing scheme for secure cloud storage, which supports data sharing with sensitive information hiding. In our scheme, the file stored in the cloud can be shared and used by others on the condition that the sensitive information of the file is protected. Besides, the remote data integrity auditing is still able to be efficiently executed. The security proof and the experimental analysis demonstrate that the proposed scheme achieves desirable security and efficiency.

REFERENCES

[1] Fursan Thabit ,Prof Sharaf Alhomdy ,ProfDr Sudhir Jagtap ,Security Analysis and Performance Evaluation of a New Lightweight Cryptographic Algorithm for Cloud Computing Environment Global Transitions Proceedings(2021).

[2] Md. Abbas Ali Khan, Mohammad Hanif Ali, A.K.M Fazlul Haque, Farah Sharmin, Md. Ismail Jabiullah, "IOT-NFC Controlled Remote Access Security and an Exploration through Machine Learning", ICT and Knowledge Engineering (ICT&KE) 2020 18th International Conference on, pp. 1-10, 2020.

[3] Zaw, Than & Thant, Min & Bezzateev, Sergey. (2019). Database Security with AES Encryption, Elliptic Curve Encryption and Signature.(2019) 1-6. 10.1109/WECONF.2019.8840125.(2019) .

[4] #Mehdi Sookhak, F. Richard Yu, and Albert Y. Zomaya, Fellow, IEEE, "Auditing Big Data Storage in Cloud Computing Using Divide and Conquer Tables," IEEE Trans. Parallel Distribution System, vol: 29, no. 5,pp. 999-1012, 2018

[5] Elimination of Redundant Data in Cloud with Secured Access Control April 2017 DOI: 10.1109/ICTACC.2017.44 Conference: 2017 International Conference on Technical Advancements in Computers and Communications (ICTACC) (2017).

[6] R. SHOBANA, K. SHANTHA SHALINI, S. LEELAVATHY and V. SRIDEVI "De-Duplication of Data in Cloud" Int. J. Chem. Sci.: 14(4), 2016

[7]] J. Blasco, R. Di Pietro, A. Orfila, and A. Sorniotti, ,,,,A tunable proof of ownership scheme for deduplication using bloom filters,"" in Proc. IEEE Conf. Commun. Netw. Secure. (CNS), Oct. 2014, pp. 481–489.

[8] K. Yang and X. Jia, "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing," IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 9, pp. 1717–1726, 2013.

[9] C. Liu, J. Chen, L. Yang, X. Zhang, C. Yang, R. Ranjan, and K. Ramamohanarao, "Authorized Public Auditing of Dynamic Big Data Storage on Cloud with Efficient Verifiable Fine-grained Updates," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 9, pp. 2234–2244, Sep 2013.

[10] Q. A. Wang, C. Wang, K. Ren, W. J. Lou, and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE Trans. Parallel Distribution Syst., vol. 22, no. 5, pp.847–859, May 2011.