International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-10, August 2019

Merkle Hellman Knapsack Cryptosystem based Cloud Auditing Protocol

A. Charles

Abstract*: The cloud environment provided the cloud storage services to allow the users for storing, managing and accessing their data through the internet. The user worried about the integrity of data because of their data can be modified by external hackers due to the limited control. The new auditing technology is required for allowing the user to use local storage without worrying about the integrity of the data. An independent entity called Third Party Auditor (TPA) is presented for auditing task rather than putting the burden of task to the user themselves. In our proposed work, the Cloud Auditor (CA) plays the role of monitoring the modifications done to the block by any external hacker if it happens; original data is recovered from its cache log. Every data that transacts between the storage, user and the CA are encrypted with Merkle Hellman Knapsack Cryptosystem based Cloud Auditing Protocol (MHKC-CAP) algorithm. Thus the proposed methodology outperforms the existing algorithms like Message Digest 5 (MD5) algorithm and Advanced Encryption Standard (AES) by means of auditing time around 13 milliseconds and accuracy in terms of False Positive Rate (FPR) and False Negative Rate (FNR) are around 0.67% and 0.5%.

***Index Terms*: Cloud Storage Services, Third Party Auditor, Message Digest 5, Advanced Encryption Standard and Merkle Hellman Knapsack Cryptosystem based Cloud Auditing Protocol..**

I. INTRODUCTION

Cloud Computing (CC) is the integration and development of many types of computing such as parallel computing, distributed computing, grid computing, and CC computed more resources to connect huge quantity of storage over the Internet [1]. Numerous forms of web based services, distributed systems and cluster computing techniques render computing resources to CC. In public institution and private management, CC plays a major role in an economic recession for the decrement of cost used by Information Technology (IT) services and the IT services also offered assured further features [2]. The key determination of the CC technology is to reduce the cost of IT companies and CC allow the standardization to manipulate the Data Centers (DC) of their own. The main features of the CC are providing the self-services build by on-demand process, the pooling of resource which is independent of location, capabilities of accessing the platform over the network, the services are rapid elasticity and the resources providing the transparency for both the provider and the customer [3]. The supply of enough software and hardware resources are handled by the Cloud Service Provider (CSP) and the hardware resources managed the client's databases. The CSP also provide mechanisms like creation, updation, and deletion process of the outsourced data to the clients [4]. A Cloud Provider (CP) offered a guarantees for hosting critical infrastructure services in the cloud and the security and flexibility are assured by the CP. The invalidate certificates, security and resilience are changed due to unpredictable changes and continuous refinements in elastic cloud environment [5]. Though the CSP provide a secure storage and reliable services to the users, the reliability of the data can be corrupted due to human's mistake or hardware/software failure.

Cloud Data (CD) storage is lacking by security of outsourced data and the storage correctness mechanism is specified by the DC. The integrity of the CD is verified by cloud auditing process and these integrity of data are stored on a cloud [6]. The auditing schemes can be broadly categorized into the following schemes such as private auditing scheme and public auditing scheme. The private scheme checks the integrity of the CD by allowing only the data owner whereas the public auditing method allowed any verifier for checking the integrity of the data [7]. The users needed to convince for that the outsourced data of the users are correctly stored in the cloud by the public integrity auditing. The various techniques proposed the number of protocols for ensuring the integrity of data in an untrusty cloud. The public scheme provides a convenient way for users allocating the auditing task to an independent TPA [8]. The TPA scheme includes three parties such as Data Owners (DO), DC and Data Integrity (DI), in which the DI is verified by the auditing scheme in a sampling style. The DO outsources the data and signatures into the center, the DC stores the data and signatures physically and sends the evidences to the TPA [9]. The user can verify the cloud DI with the help of TPA, which is denoted as public cloud storage auditing and the consistency, DI ratings and higher reliability is achieved by means of CSP with public auditing to maintain the reputation [10].

In this paper, the design and implementation of cloud storage framework called MHKC-CAP and the privacy of user's data is more reliable and secure in cloud storage by using the proposed algorithm. The multi-cloud is used for storing the large volume of user's data by which the data availability can be achieved by the MHKC-CAP algorithm. The rest of paper is organized as follows: Section II surveys several recent papers on cloud auditing protocol. In Section III, the detailed description of the proposed method MHKC-CAP is presented. In Section IV, the performance of the MHKC-CAP is evaluated by conducting set of experiments. Finally, the conclusion is made in the Section V.

Manuscript published on 30 August 2019.
***Correspondence Author(s)**
Dr. A.Charles, Assistant Professor Government College of Engineering, Bargur, India

II. LITERATURE REVIEW

Several techniques are suggested by researchers in cloud auditing protocol. In this scenario, a brief evaluation of some important contributions to the existing literatures are presented. Salem Mohaned Zkaria, *et al.,* [11] designed and implemented a storage framework in cloud that contained an effective TPA and the main objective was preserved the privacy of data while providing the auditing services of data by the TPA. The major purposes of the protocol were composed by several functions such as public audibility, storage correctness, improving data availability, preserving data privacy, and effectiveness. The TPA achieved the public audibility and storage correctness by granting the necessary privileges for implementing the auditing process and detecting any corrupted data. The techniques for encryption process were introduced namely MD5 that was used for the efficiency of the protocol. The other process was AES in the auditing protocol while the time required for the encryption was minimized. The performance on a set of files with different sizes in terms of time and accuracy was evaluated by set of experiments and the mechanism was unable to allow the data operations dynamically which was a problem of the protocol.

J. Shen, *et al.,* [12] recommended a dynamic structure made up of a Doubly Linked Info Table (DLIT) and a Location Array (LA) in a public auditing protocol. The mutual trust between the DO and cloud service providers were achieved by implementing the global and sampling verification. The protocol supported the various auditing properties such as public auditing, blockless verification and batch auditing for improving the performance of the protocol. The relationship between the DLIT and LA performed the efficient dynamic support and the reduction of overhead when compared with the state of the art. The basic challenges such as batch auditing, blockless verification and lazy updates were overcome by the scheme. The security of the protocol was indicated by the theoretical proof and the auditing protocol was less computationally expensive when compared to the existing methods.

S. Anbuchelian, et al., [13] presented the algorithm called secure cryptographic hash algorithm which was used to encrypt the data and these encrypted data were splited into number of chunk files. The RSA key generation algorithm was enhanced with the auditing scheme and developed a new modified cryptosystem known as Modified RSA Cryptosystem (MRSAC). The safer and trusted data were given to the cloud user which was retrieved from the Cloud Server (CS) and these trusted data was developed by a key generation process. The efficiency of the method was validated by the experimental results which was considered as an advantage of the method. This auditing scheme was suffered from the high cost for processing the encryption and the trustworthiness was increased by providing an interface to view the file details of users such as upload files, download files, modified files and showed the exact modification.

Zhang, *et al.,* [14] proposed an auditing protocol which was based on ID-depend cryptography for cloud DI and the protocol was secured in the random oracle method. The efficient ID-based auditing protocol was associated with Diffie-Hellman problem for reducing tight security. The method also extended to support batch auditing in the multi-user settings and the security and the results showed that the ID-based auditing protocols were secure and efficient, especially reduced the computation cost of the auditor in the multi-user setting. The ID-based remote checking protocol was more suitable when compared with the existing methods in the large-scale cloud storage system. The simulation results showed the cost of CS was linear with the number of the challenged data blocks and the protocol achieved the error detection probability.

Y. Yu, et al., [15] checked the integrity of data by developing the Identity- Based CDI Checking (ID-CDIC) protocol and eliminating the complex certificate management. The method was concreted the construction from RSA signature which the public auditing and the variable-sized file blocks were supported by ID-CDIC method. The method provided the formal security in random oracle model and the construction of security of the ID-CDIC was proved under the RSA assumption with the large public exponents. The method developed a prototype which was demonstrated by the experimental result to test the efficiency of the ID-CDIC scheme. The method were unable to change the size of the block once the block size was fixed to balance the computation cost of data owner and the verifier.

III. PROPOSED METHODOLOGY

The CC provide the services like storage space to cloud users, accessing the data, reduce the burden for hardware and software maintenance. Though CSP provides several services, this may also affected by challenges in the range of security and data preserving. The CD may exist like white puffy in cloud services and are affected by many temporal errors. The CP allow the Data Proprietor (DP) to audit the integrity of data repeatedly for ensuring the outsourced data is secure and is not intrude. Researchers proposed various Remote Data Auditing (RDA) protocols for auditing the DI of the cloud storage so far. In presented work, analyzing the issues of current protocol for DI auditing in cloud storage and propose an approach based on MHKC for handling insecurity problem. The dynamic operations of data may include inserting the data, modifying the data based on user's need and deletion of data which is supported by the protocol MHKC. The protocol contain less computational cost and supported the public auditing of data. The basic structure of the cloud architecture are as follows:

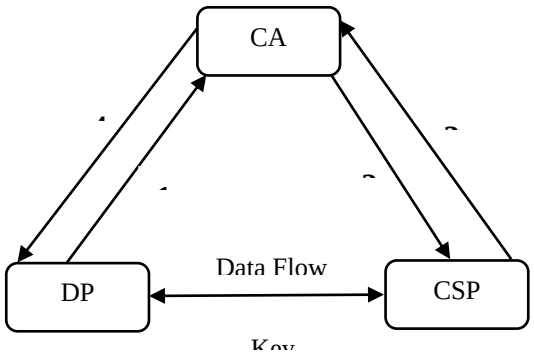

Figure 1 Structure of Cloud architecture

Retrieval Number: J93810881019/19©BEIESP
DOI: 10.35940/ijitee.J9381.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Where,

1. Delegate auditing task
2. Challenges
3. Proof of Possession
4. Result of integrity audit

This auditing model contains three entities are described as follows:

❖ **Data Proprietor (DP):**

The outsourcing is carried out with sufficient amount of data files in DP and these DP may be a firm, a business organization or a person. The outsourced data may be update later by several operations such as insert, modify and delete. Generally, DP is a resource bounded entity and a dependent entity for data maintenance on CP.

❖ **Cloud Service Provider :**

The CSP have unlimited storage space and having sufficient resources which are computationally capable and stored many servers. The CSP is considered as an entity which is used to keep the preserved outsourced data and a CSP is expressed as an untrusted object.

❖ **Cloud Auditor (CA):**

The DP's data are audited by skilled object called CA and both the service provider and DP trusted the cloud auditor. While auditing the data on DP, the CA minimizes the computational data pressure.

Threat Model :

Some threats can be caused to DP's data by both CA and CSP are explained as below:

3.1.1. Threat caused by CA:

CA is independent object which is genuine and reliable and the data integrity is assured by DA that is relied on CA. The DA's data is maybe curious for CA and in a public auditing protocol CA compromised the privacy of data. A privacy maintenance is needed along with the DI preservation from preventing the CA to gain any knowledge from the DA's data.

3.1.2. Threat caused by CSP:

Some threats are caused by CSP to DS's data are given as below:

a) The DS's data may damage permanently due to some processing error which is caused by CSP.
b) The CSP may remove rarely accessed data without notifying DS for saving server space.

3.1.3. Some external threat:

a) The stored data may be harmed by the former administrator which can break into a cloud server at CSP.
b) The outsourced data through an API can be accessed by a valid user which is offered by CSP.
c) The DS's data may attain risk due to weak API and an unauthorized person can use the information of valid user' data that can cause pollute or delete data without being exposed.

The CSP are offered variety of cloud services, prevent external attacks to the data and securing data accessibility is more important in the proposed method.

3.2. MHKC based on Cloud Auditing Protocol Architecture :

Cryptography is a tool which is used for the security purpose of the sensitive data, specifically the issue of confidentiality of the data. The encryption algorithms are classified as Symmetric (secret key) and Asymmetric key (public key). The key is represented as sequence of letters, symbols and numbers and the encryption algorithm strength is directly depends on the key, therefore key chosen is a significant issues. In the proposed MHKC-CAP, the algorithm is used for the cryptography is MHKC and the basic description of the algorithm is presented as below:

Merkle Hellman Knapsack Crypto System:

The MHKC is a one-way process and the two keys are needed for communication (i.e. public key and private key) in MHKC which is an asymmetric-key cryptosystem. The one way process is like for the encryption process, the public key is used and the private key is used for the decryption method. The authentication process is not suitable in MHKC algorithm because of one way method in cryptographic signing. The subset sum problem is used in the MHKC process (i.e. a knapsack special case problem) and the NP-complete problem is maybe called as MHKC algorithm. The set of numbers is greater than the sum of all the numbers in set and these sum of number is lesser than the each element of the set, then the set of numbers is superincreasing. The problem can be solved with a greedy algorithm in simple technique with polynomial time and the MHKC is considered as an easy problem.

Key Generation:

The following equations are able to describe the process of generation of public key and private key,

$$\beta = r\, w_i \ mod\ q \qquad (1)$$

Where, β is the public key, r is a random integer, such that gcd (r,q) = 1 (i.e. r and q are coprime)

$$q > \sum_{i=1}^{n} w_i \qquad (2)$$

For encrypting the n- bit messages, $w = (w_1, w_2, \ldots . w_n)$ whereas n is natural nonzero natural numbers and a random integer q.

Encryption:

The process of encrypting the public key, the following equation are calculated as,

$$c = \sum_{i=1}^{n} \alpha_i \beta_i \qquad (3)$$

An n-bit messages is used to encrypt as follows

$$\alpha = (\alpha_1, \alpha_2, \ldots . \alpha_n) \qquad (4)$$

where α_i is the *i*-th bit of the message, $\alpha_i \in \{0,1\}$, and c is a ciphertext.

Decryption:

The process of decrypt the ciphertext *c* a receiver has to find the original message, the equation should satisfy the following conditions as,

$$c = \sum_{i=1}^{n} \alpha_i \beta_i \qquad (5)$$

An instance of the subset problem is hard because β_i is a random values but the decryption process is easy because of the known value of a private key (w, q, r). The integer s is identified for the key to decrypt the message. The equation computes as

$$c' \equiv cs\ (mod\ q) \qquad (6)$$

Where, c' is a received ciphertext.

Hence

$$c' \equiv cs \equiv \sum_{i=1}^{n} \alpha_i \beta_i s\ (mod\ q) \qquad (7)$$

The $\beta_i = rw_i\ mod\ q$ follows because of $rs\ mod\ q = 1$

$$c' \equiv \sum_{i=1}^{n} a_i w_i \pmod{q} \quad (8)$$

Thus the final equation to solve the subset sum problem for decrypting the ciphertext is as

$$c' = \sum_{i=1}^{n} a_i w_i \quad (9)$$

3.2.2. Schematic Architecture of the workflow of MHKC-CAP:

In this section, the work flow of the proposed method described the generation of hash values and the process of encryption and decryption techniques are as followed:

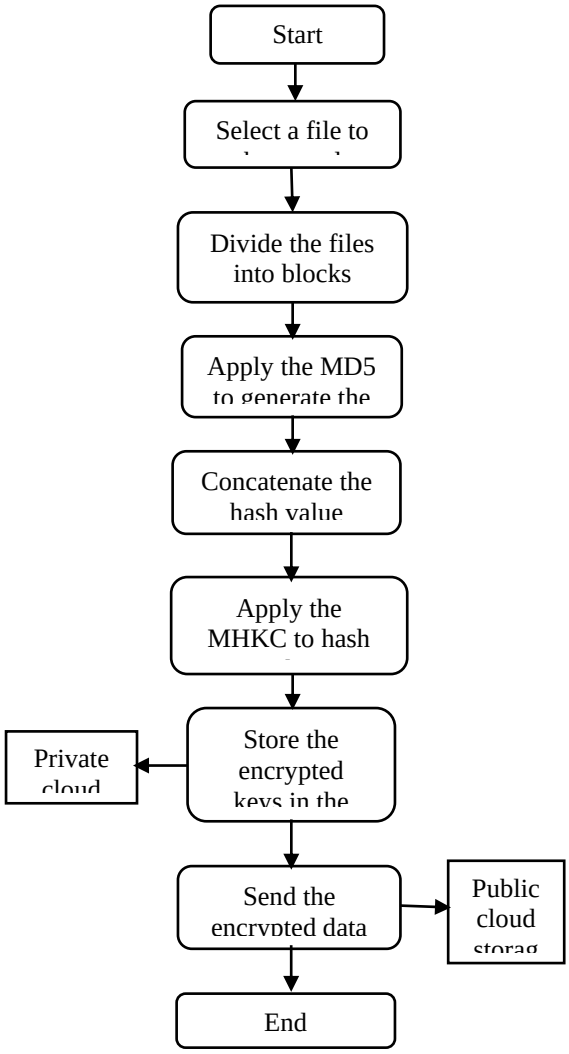

Figure 2 Work Flow Architecture of MHKC-CAP

The client is responsible for selecting the file that needs to be stored remotely on the cloud storage. Then, the file is divided into blocks of certain size. After that, the MD5 is applied on the divided blocks to generate hash value for each block. The basic properties for using MD algorithm is the hash values can J93810881019be easily get from the given message (input) of any length and the algorithm produce the fixed-length value of output for auditing purposes. The MD5 algorithm divides the input data into groups in a unit of 512 bits, and every group is divided into 16 groups of 32 bits, and at last, 4 groups are concatenated into a hash value of 128 bits which is the output result. Then, the hashes are concatenated into one string to be the input for the MHKC algorithm. Also, it is the responsibility of the client to store the keys used in the encryption process locally. In order to store the data in the cloud storage, the scenario occurs when the TPA works in Semi Trusted mode; it is the responsibility of the client to upload the file into the cloud storage directly without TPA intervention. The MHKC Algorithm has been selected as the main encryption algorithm in the proposed auditing mechanism because of its immunity against some of the problems.

The doubly encryption process is adopted to make the task of the intruder harder if he wants to read the content of the data and hence the confidentiality of the data is preserved. After the completion of this doubly encryption process, the client stores the encryption keys locally on a private cloud and transfer the encrypted data to the selected cloud storages by himself. If a Semi-trusted TPA is adopted or only directs the data to the TPA which transfers the encrypted to the selected storage on the behalf of the client if a Fully-trusted TPA is adopted. The MHKC-CAP method stored the secured privacy files by implementing authenticator-evolving mechanism with zero-knowledge and proceed the key updation process. The method can also integrates with homomorphic linear authenticators, proof systems and proxy re-signatures. For updating the key, the user need not regenerate all the authenticators and download all the entire files, instead of that he used to download one single file tag. After the key updation, the user upload the new files together with some information and store the verification instruction to the CS for undertaking the least amount of workload in the updating phase.

IV. EXPERIMENTAL RESULTS

The proposed MHKC-CAP based CA method has been implemented with the help of java jdk 1.8 and NetBeans 8.2 IDE. Here the storage architecture is built with the help of oracle cloud server and accounts for CA, External Hacker and user are created. A User Interface has been built to represent the vision of user, hacker and CA separately. By running the user page one can view his/her account and the documents they have stored and their details. And also they can also be able to manage and access their documents. While in CA side one can see the list of documents stored and number of hashed blocks for each document. And one can view the block hashed encrypted block and challenge whether any external modification is done or not. If done the original data can be recovered in place of the modified data. In the hacker page, one can view the list of documents stored by user and can able to modify any block in the document. Thus the demonstration of the complete CA against an external hacker attack can be visualized with the help of the experimental set up.

For the evaluation of the proposed methodology 150 different files with different types and numbers and sources are given in the table 1 below. The size of each file is different as a total they were comprised of 1.5 Giga Bytes.

Retrieval Number: J93810881019/19©BEIESP
DOI: 10.35940/ijitee.J9381.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Table 1 Set of files used to evaluate CA

S. No.	Type of File	No. of Files	Source
1	Document	37	Microsoft Word
2	Text	25	Notepad
3	Image	35	Internet
4	PDF	14	Internet
5	Video	24	YouTube
6	Audio	3	Internet
7	PowerPoint	12	Microsoft PowerPoint

A consecutive set of experiments have been performed on the proposed system to evaluate its performance based on the set of data described in below table (Table 2). It comprises of 25 different files with 25 different sizes to measure the parameters like upload time and encryption time consecutively.

Table 2 Timing Measures of Proposed MHKC-CAP system

S. No.	Type of File	Size of File (KB)	Upload Time (millisecond)	Key Generate Time (millisecond)	Encryption Time (millisecond)
1	Text	10	154	98	137
2	Text	14	168	105	142
3	Text	16	175	124	156
4	Text	22	183	139	162
5	Text	25	191	150	181
6	Text	23	199	167	187
7	Text	31	203	184	195
8	WORD	37	210	207	204
9	WORD	125	213	215	213
10	WORD	1075	224	228	220
11	Photo	1248	235	235	231
12	Photo	1551	249	247	238
13	Photo	18464	258	258	245
14	Photo	33123	267	262	256
15	Photo	37420	274	269	262
16	Photo	72748	285	283	270
17	Photo	1154490	296	290	275
18	Photo	1155207	340	303	281
19	Mp3	3507764	369	317	286
20	Mp4	5120428	377	325	290
21	Mp3	10325038	385	336	293
22	Mp4	20570492	396	348	297
23	Mp3	21674900	407	353	310
24	Video Mp4	27233473	420	359	334
25	Video Mp4	2733537	436	370	342

The experimental setup is evaluated for computing its performance in means of time like upload time and encryption time. Upload time is the time taken to load a file into CS or it is the time difference between start and end of uploading process of a file. The values obtained for upload time of various file sizes is tabulated in Table 2 and it is plotted in the below graph (Fig 3).

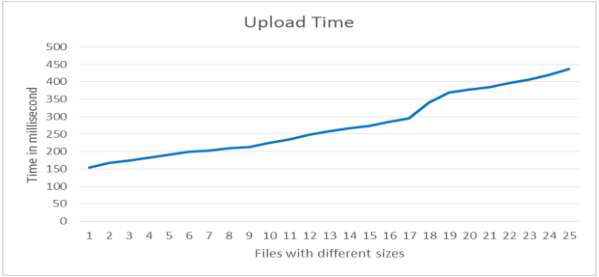

Figure 3 Upload Time for files with different sizes

The generation of the encryption keys for each file is computed from the second experiments and the files is encrypted by using MD5 encryption algorithm. The results of the second experiments from table 2 is shown in Figure 4.

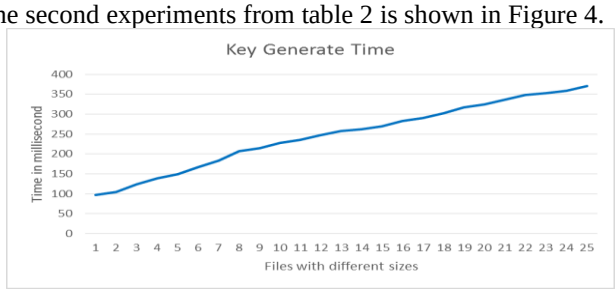

Figure 4 Time Generation for the encryption keys for set of files with different sizes

Encryption time is defined as the time to encrypt all the blocks in a document to be uploaded. In proposed methodology, MHKC-CAP algorithm is used and its encryption time for various size of files is taken and plotted in the below graph (Fig 5).

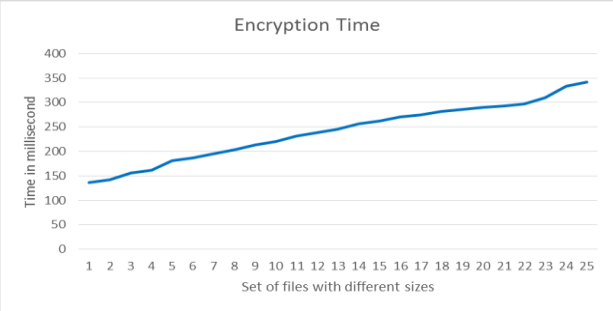

Figure 5 Encryption Time for Files with Different Sizes

The time taken for auditing the file blocks is said to be as the auditing it includes process like challenging, verifying blocks for modification etc., its values are taken and tabulated in the following table (Table 3) and graph (Fig 6) below.

Table 3 Auditing Time for various CA methods

S. No.	Size of File	Time taken for Auditing by RSA digital signature	Time taken for Auditing by MD5 and AES scheme based CA	Time taken for Auditing by MHKC-CAP scheme
1	20	25	19	13
2	50	34	29	22
3	80	41	33	25
4	110	49	40	34
5	140	56	46	40
6	170	62	51	45
7	200	69	58	50
8	230	76	63	57

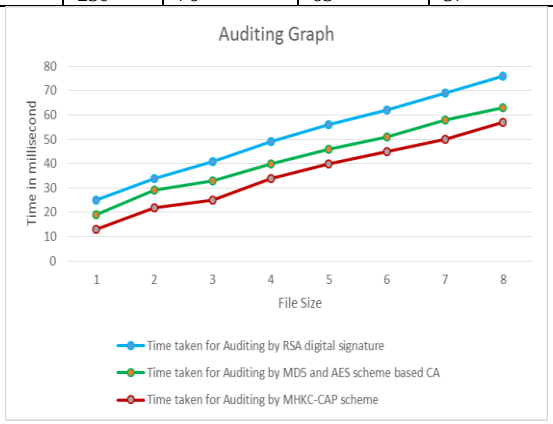

Figure 6 Auditing Time for several CA techniques

Finally, 150 different files with different sizes have been uploaded to the proposed methodology and verified for the accuracy of CA. The half of files has been modified with changes by hacker tool and then CA is said to verify the whole data set and the accuracy of finding the modifications by CA is found. The two parameters namely FPR and FNR evaluated the efficiency of CA. FPR is the means the amount of false notifications raised by the system in case of identifying an unmodified block of data as modified and FNR means the amount of false notifications arose by the system when challenging a modified data as unmodified. The equations for calculating these parameters are described below:

$$FPR = \frac{FP}{TN+FP} \times 100 \quad (10)$$

$$FNR = \frac{FN}{TP+FN} \times 100 \quad (11)$$

Where,

FP represents **F**alse **P**ositive
FN represents **F**alse **N**egative
TP represents **T**rue **P**ositive
TN represents **T**rue **N**egative

By use of the above equations the FPR and FNR values are calculated for the various CA schemes and they are tabulated in the table (Table 4) below and its comparison chart is plotted in the fig 7.

Table 4 FPR and FNR rate for several CA schemes

CA method	FPR	FNR
RSA digital signature scheme	4.3%	3.2%
AES and MD5 based CA scheme	2.2%	1.7%
MHKC-CAP based scheme	0.67%	0.5%

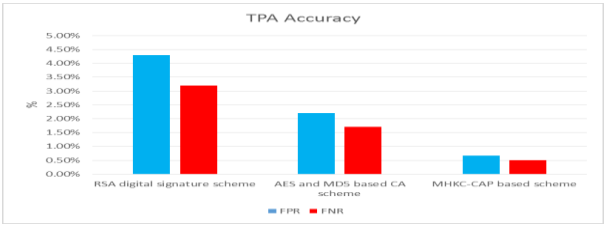

Figure 7 FPR and FNR rate for various CA schemes

From the graph illustrated above, it is inferred that proposed CA scheme has better efficiency in monitoring external hacker modifications in the data block stored in CS. And also from the total evaluation discussed it is stated clearly that proposed MHKC-CAP scheme has capability to monitor the data access in a storage efficiently along with privacy preserving and lost data recovery mechanism.

CONCLUSION

Among the services provided by the cloud storage service which allows the users to store, to manage remotely and to access their data over the internet. But still, there is some of research problems and challenges with the CSP and this to be mitigated to increase the better services for users. Some of the problems are presented as users' anxiety about the integrity, availability, and confidentiality of their data and their feelings that the data can be accessed. Hence, to justify the fear of the users the public auditing mechanism is needed for the stored data in cloud. In this work, a MHKC-CAP protocol is considered with employing a TPA for auditing process on data owners while preserving the data privacy. The design objectives of the proposed protocol are public audibility, storage correctness, improving data availability, preserving data confidentiality, and efficiency. The proposed protocol efficiently supports freshness of data, public auditing of data, data dynamic procedures, which are missing in most of the existing techniques. In encryption process, the redundant data is more for encrypting the plaintext, therefore there is a chance of increasing computation overhead. In future work, the proposed mechanism can be modified to reduce the complexity of computational overhead.

Retrieval Number: J93810881019/19©BEIESP
DOI: 10.35940/ijitee.J9381.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

REFERENCES

1. Yang, Changsong, Xiaofeng Chen, and Yang Xiang. "Blockchain-based publicly verifiable data deletion scheme for cloud storage." Journal of Network and Computer Applications, vol. 103, pp. 185-193, 2018.
2. Alassafi, Alharthi, Walters, and Wills, "A framework for critical security factors that influence the decision of cloud adoption by Saudi government agencies", Telematics and Informatics, vol. 34, no. 7, pp, 996-1010, 2017.
3. El-Booz, Sheren A., Gamal Attiya, and Nawal El-Fishawy. "A secure cloud storage system combining time-based one-time password and automatic blocker protocol." EURASIP Journal on Information Security, no. 1, pp. 13, 2016.
4. T. Xiang, Li, Chen, Yang, and Zhang, "Achieving verifiable, dynamic and efficient auditing for outsourced database in cloud", Journal of Parallel and Distributed Computing, vol. 112, pp. 97-107, 2018.
5. Hudic, Aleksandar, Paul Smith, and Edgar R. Weippl. "Security assurance assessment methodology for hybrid clouds." Computers & Security, vol. 70, pp. 723-743, 2017.
6. Shen, Yu, Xia, Zhang, Lu, and Hao, "Light-weight and privacy-preserving secure cloud auditing scheme for group users via the third party medium", Journal of Network and Computer Applications, vol. 82, pp. 56-64, 2017.
7. Xu, Wu, Khan, Choo, and He, "A secure and efficient public auditing scheme using RSA algorithm for cloud storage", The Journal of Supercomputing, pp. 1-25, 2017.
8. Luo, Xu, Huang, Wang, and S. Fu, "Efficient auditing for shared data in the cloud with secure user revocation and computations outsourcing", Computers & Security, vol. 73, pp. 492-506, 2018.
9. Wan, Zhang, Pei, and Chen, "Efficient privacy-preserving third-party auditing for ambient intelligence systems", Journal of Ambient Intelligence and Humanized Computing, vol. 7, no. 1, pp. 21-27, 2016.
10. Zhang, Jianhong, Hongxin Meng, and Yong Yu. "Achieving public verifiability and data dynamics for cloud data in the standard model." Cluster Computing, vol. 20, no. 3, pp. 2641-2653, 2017.
11. Salem Mohaned Zkaria, Sahar F. Sabbeh, and E. L. Tarek, "An Efficient Privacy Preserving Public Auditing Mechanism for Secure Cloud Storage", International Journal of Applied Engineering Research, vol. 12, no. 6, pp. 1093-1101, 2017.
12. J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data", IEEE Transactions on Information Forensics and Security, vol. 12, no. 10, pp. 2402-2415, 2017.
13. S. Anbuchelian, C. M. Sowmya, and C. Ramesh. "Efficient and secure auditing scheme for privacy preserving data storage in cloud." Cluster Computing, pp. 1-9, 2017.
14. Zhang, Jianhong, and Qiaocui Dong, "Efficient ID-based public auditing for the outsourced data in cloud storage", Information Sciences, vol. 343, pp. 1-14, 2016.
15. Y. Yu, L. Xue, M. H. Au, W. Susilo, J. Ni, Y. Zhang, and J. Shen, "Cloud data integrity checking with an identity-based auditing mechanism from RSA", Future Generation Computer Systems, vol. 62, pp. 85-91, 2016.
16. Wei, Jinqiao, Ying Wang, and Xiaoxue Ma. "Text image authenticating algorithm based on MD5-hash function and Henon map." Ninth International Conference on Digital Image Processing (ICDIP 2017). Vol. 10420. International Society for Optics and Photonics, 2017.