International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Cloud Dynamic and Public Auditing Scheme for Secure Data by using RSA with Modified Dynamic Hash Table

[1]A. Ahadha Parveen, [2]P. S. S. Akilashri

Submitted: 20/12/2023 **Revised**: 31/01/2024 **Accepted**: 10/02/2024

Abstract: Users are able to host and access a broad variety of internet-based services thanks to a paradigm for distributed computing that is known as "cloud computing." It sees widespread deployment in corporate applications like as data storage and internet-based software programmes. Users of cloud storage services are able to access their personal information whenever they want and from any place they want since they are not required to keep the data locally. The cloud service provider, on the other hand, does not put their whole faith in cloud storage since the integrity of the data is reviewed on unstable cloud servers. A significant amount of research has been conducted in public auditing with the goal of lowering the amount of computation time required for the integrity check. The method that is used the most has low costs associated with the processing but offers no security. In the work that we have suggested, a component known as the Cloud Auditor (CA) is responsible for carrying out the task of monitoring any changes that may have been made to the block by an external hacker; original data is obtained from its cache record. The TPA is where the data information required for dynamic auditing is stored, and it is recorded in the TPA's modified dynamic hash table (MDHT). The information is encrypted via the MRSA algorithm, which is a modified version of the RSA algorithm. The MDHT is not the same as a dynamic hash table since the former does not include a tag block while the later does. The cost of calculating the MDHT is analysed, then compared to other approaches that are currently being used. The results of the experiments showed that the MDHT method has a reduced computation cost compared to the most current public auditing methodologies.

Keywords: *Cloud Computing, Cloud Auditing, Modified Dynamic Hash Table, Data Integrity, Computation Cost.*

1. Introduction

When customers use sharing and data storage services such as Google Drive, it is simple for them to collaborate on the same resource while also sharing their data with one another in the cloud. When a user keeps data in the cloud that is shared with other users, that most current version of the data may be accessed, edited, and shared with other user communities [1]. In the worst case scenario, a cloud owner can intentionally generate data error occurrences in order to avoid monetary losses or to keep their good image among consumers [2]. In spite of the fact that cloud computing is constantly expanding and evolving, there is still confusion and dispute regarding its usage. This is due to the fact that users' primary concern in a cloud environment is the safety of their data [3]. Customers are unable to withdraw their data from the cloud once they no longer have direct control over it [4]. This is particularly true for public clouds that have considerable degrees of both multi-tenancy and consolidation. Cloud Service Providers (CSPs), who are responsible for managing customers' data stored in the cloud, may decide not to keep or delete data that is accessed seldom by typical customers [5]. This decision is made in order to free up extra storage space. Because users are concerned about data integrity when using cloud storage services, they want a process that will allow them to audit the cloud server and ensure that it will maintain all of their most current data without causing any corruption [6]. Clients are responsible for ensuring the data integrity of their files while they are stored in the cloud, but they are unable to remove the subpar cloud servers that leave their data vulnerable to security breaches [7]. Integrity, availability, and confidentiality are the three basic aims of security, with integrity being assured by auditing cloud data or the external verification of DI, which has been a popular issue in recent years [8]. The cloud does frequent checks on the DI by adding the TPA [9], which helps end users relieve some of the computational burden that they are under. On the other hand, the bulk of the techniques that are currently being used suffer from issues such as excessive TPA calculation costs and an insufficient level of TPA security [10]. In order to solve the concerns about both the cost of computing and the level of security, this research constructed an MDHT. The strategy that is currently being used, which uses a dynamic hash table, is analysed and compared with the one that was recommended. The research is broken down into four sections: a literature

[1]*Department of Computer Science, National College (Affiliated to Bharathidasan University), Trichy,*
Tamil Nadu, India,620020
Email: ahadha.parveen@gmail.com
[2]*Department of Computer Science, National College (Affiliated to Bharathidasan University), Trichy,*
Tamil Nadu, India,620020
Email: akilas27@gmail.com

assessment of existing approaches for data security in part II, an explanation of the recommended strategy in section III, and an illustration of the experimental findings in part IV of the study. In the next section (V), the results of this investigation are presented.

2. Literature Review

The research methods for cloud data storage authentication are reviewed in this part. A quick assessment of some significant contributions to the literatures now in use is provided.

D. Chattaraj, M. Sarma, and A.K. Das, [11] proposed an effective method of authentication that alleviates concerns about potential security vulnerabilities. The suggested authentication protocol combines a unique key exchange method with active password-based two-server authentication in order to provide a solid user privacy feature. This was done in order to protect the user's privacy. A combination of authorised and unauthorised security workers are providing the protocol with defence against any attacks that could be launched against it. When compared to the authentication techniques that are presently being used, the outcomes of the research reveal that the suggested approach will have overheads that are comparable to and adequate for those that are already in use. These include processing time, communication overheads, and cost. The standard method depended on passwords, which are insecure and may be broken into by attacks employing stolen verifiers and key rollover problems.

J. Brogan, I. Baskaran, and N. Ramachandran, [12] the primary emphasis is on uploading data from various health activities to a distributed ledger. This data is obtained by various wearable and embedded devices. The encrypting, authenticating, and broadcasting of the activity data was handled by a protocol called Masked Authenticated Messaging (MAM), which was built from the application layer of the IOTA stack. This new paradigm may expand the capacities for exchanging data throughout the digital healthcare ecosystem while also restoring patient autonomy over the data pertaining to their own health activities. Nevertheless, when a one-to-one solution is implemented, the technique that was provided allows for unsecure data sharing.

A. Razaque, and S.S. Rizvi, [13] developed the triangular data privacy preservation protocol is a ground-breaking innovation in the realm of privacy protection that will enhance the safety of three important stakeholders. The approach that has been presented begins by putting in place a unique authentication procedure involving all three parties. After that, the method that has been built carries out three essential procedures, which are as follows: cloud service provider, dangerous detecting role for TPA, and dual authentication for the use of cloud services. The outcomes of the experiment illustrate the effectiveness and efficiency of the approach that was recommended, which comprises conducting audits on all of the important stakeholders. As a consequence of the involvement of other parties, the tried and true method creates not one but two important problems: data loss and inadequate data backups.

H. Zhang, and T. Tu, [14] presented to protect against collision and dishonest entities, as well as to permit dynamic modifications that could be independently validated, an outsourced dynamic auditing system was developed. For the purpose of batch-verifying the many leaf nodes and the various indices associated with them, a Batch-Leaves-Authenticated Merkle Hash Tree was used (BLA-MHT).The expenses of setting up the audit for both the user and the TPA were lower as compared to the conventional method of conducting public audits. However, there are problems with the sophisticated BLA-MHT infrastructures as well as the outside platforms in the networks.

G. Sharma, and S. Kalra, [15] proposed a strategy for the construction of cloud servers that included key agreement methods with quantum identity-based authentication. On the other hand, the quantum is inherited with the fundamental rule of quantum physics and the theory of quantum information. The protocol security analysis illustrates how resistant the suggested solution was to any and all security breaches. The results of the experiment show that it is possible to generate shifted keys effectively across a distance of one hundred kilometres of optical fibre while maintaining a key rate and an error rate. Since the architecture of cloud computing was open, there was a significant increase in the number of efforts at social engineering and phishing.

3. Proposed Methodology

Customers of the CC may take use of benefits such as increased storage capacity, easier access to data, and reduced hardware and software upkeep costs. Despite the fact that CSP provides a broad array of services, such services might be negatively affected if there are problems with data preservation or security. It's possible that many timing errors will have an effect on the CD, making it seem like a white fluffy in cloud services. To ensure that the data that has been outsourced is secure and has not been altered in any way, the Controlling Party (CP) grants permission to the Data Proprietor (DP) to do regular audits of the data's integrity. Researchers have created a variety of Remote Data Auditing (RDA) strategies up to this point in order to audit the DI of cloud storage services. The research that is now being presented examines the issues that are present with the process that is currently in place for cloud storage DI audits, and it proposes a solution that is based on MHKC for the purpose of fixing the security problem. Dynamic actions on data are supported by the MHKC protocol, and some examples of these operations include the insertion of data, the alteration of data in response to user requirements, and the deletion of data. The protocol enabled public data audits and reduced the amount of computer resources required. The basic structure of the cloud architecture are as follows:

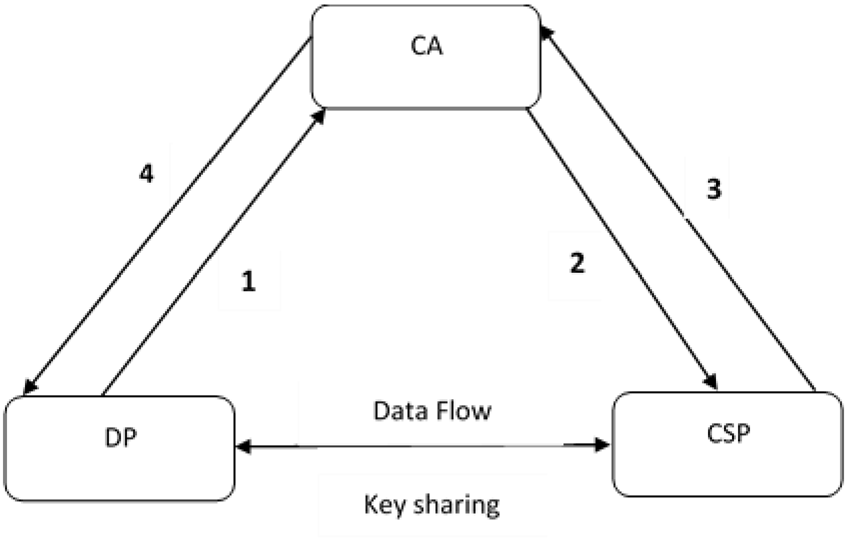

Fig 1 Structure of Cloud architecture

Where,

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

This model of auditing is composed of three different entities, and they are as follows:

❖ **Data Proprietor (DP):**

The outsourcing process is completed after there are sufficient data files in the DP, which might be a business, an organisation within a firm, or an individual. The data that was outsourced may at a later time be updated using a variety of processes, such as adding new data, editing existing data, or deleting old data. In general, DP is an entity that has limited resources and is reliant on CP for the maintenance of its data.

❖ **Cloud Service Provider :**

The capacity of the CSP's storage is almost unbounded, and it also has sufficient resources to manage computations and house a number of servers. An untrusted object, also known as a CSP, is an entity that is used to keep saved data from being outsourced and is considered to be a part of the cryptographic system.

❖ **Cloud Auditor (CA):**

The cloud auditor, who had been tasked with verifying the validity of the DP's data on behalf of an informed organisation known as CA, enjoyed the confidence of both the service provider and the DP. The cloud auditor had been tasked with confirming the legitimacy of the DP's data. During the process of auditing the data on DP, the CA helps to alleviate some of the strain placed on the computational data.

3.1. Threat Model :

The following is an explanation of the potential dangers posed to DP's data by CA as well as CSP:

3.1.1. CA-related threat:

DA is responsible for ensuring the data's integrity, and it does so by relying on CA, an independent object that is both reliable and genuine. It's possible that CA finds the DA's data interesting, but by using a public auditing protocol, CA violated the DA's right to data privacy. In order to stop the CA from discovering anything from the DA's data, the DI has to be preserved, and at the same time, a privacy maintenance has to be performed.

3.1.2. CSP-related threat:

The following is a list of the many risks that CSP presents to DS's data:

a) The DS's data may get irreversibly damaged as a result of a CSP-caused processing mistake.

b) For the sake of reducing server space, the CSP may delete infrequently used data without informing the DS.

3.1.3. Some external threat:

a) It is possible that the former administrator hacked into a cloud server at CSP and corrupted the data that was stored there.

b) Through the use of an application programming interface (API), a CSP-approved user may have access to the data that has been outsourced.

c) The DS's data may be at jeopardy as a result of the bad API, and an unauthorized person may be able to use genuine user information to delete or taint data without being discovered.

Within the framework of the proposed approach, the protection of data accessibility and the prevention of attacks from the outside are of greater importance. The CSP has access to a variety of cloud services to choose from. The explanations that follow demonstrate how this method use the ”Modified Dynamic Hash Table (MDHT)” to audit the CSP and protect the data from unauthorised users,

3.2. Description of Proposed Methodology

The next parts will give in-depth descriptions of this three-step process (MRSA).

3.5.1. Setup Phase:

Prior to storing the file in the cloud, the user should pre-process the file in order to guarantee that the information can be accessed, that its integrity is preserved, and that its confidentiality is maintained.

- **Encoding:** Encrypting the file is a need for the user in order to validate the accessibility of the data that was stored in the cloud.
- **KeyGeneration:** Using this method, the user generates a pair of keys, one of which is private and the other of which is public. These keys are needed for the most advanced kind of file processing that the suggested system would do.
- **Encryption:** Users encrypt the data using a method called public key cryptography when the owner of the data wishes to ensure that the data remains private.
- **MetadataGeneration:** Users must first compute the information contained inside each file block before they can access the DI that has been saved in the cloud storage system.

3.5.2. Verification Phase:

When a user wants to verify data that is stored on cloud servers, the user or his chosen agent TPA utilizes the Challenge-Response Protocol to check the DI rather than having a local copy of the data. This is necessary since the user does not have access to the data. The verification process is comprised of the three steps that are listed below.

- **Challenge:** In order for the DI to be validated, the verifier generates a random challenge and then transmits it to the CSP.
- **Response:** When a challenge request is sent by the verifier, the CSP will create an integrity proof as a response and then transmit it back to the verifier.
- **Check Integrity:** After receiving a response from CSP, the data from that answer is compared to the metadata that was computed in the past in order to establish whether or not the updated proof is valid. If the response does not correspond to the metadata, then the integrity of the data cannot be guaranteed; this will indicate that the data has been corrupted.

3.5.3. Secret sharing MRSA Algorithm

A novel MRSA algorithm is produced whenever the digital signature is used in conjunction with a method for the distribution of confidential information, that are as follows:

A digital signature technique uses three effective algorithms: KeyGen, Sign, and Ver.

➢ ”KeyGen is the key generation algorithm. This outputs a key pair (P,S). P is the public key and S is the secret or private key”.

➢ “Sign is the signing algorithm. Given a message μ and the secret key S, it outputs a digital signature σ”.

➢ “Ver is the verification algorithm. Given a message μ, the corresponding signature and the public key P, it succeeds if σ is a valid signature of the message μ”.

A MRSA signature system consists of the following four key elements, which are detailed below:

❖ Consider n as the security parameters, Q_N being generated with k elements, singing servers being l, t as the threshold parameters, ω being a random string. These factors are all used as inputs for the key generation algorithm, which produces the outputs as (N,e) is a public key, where n is the size in bits of N, the private keys $d_1, \dots d_l$ only known by the correct server and for each $u \in [1, k]$ a list v_u, $v_{u,1} = v_u^{d_1}, \dots v_{u,l} = v_u^{d_l} mod\ N$ of verification keys.

❖ The input of a share signature algorithm is (N, e), an index $1 \leq i \leq l$, the private key d_i and a message m; this outputs a signature share $s_i = x^{d_i}\ mod\ N, where\ x = H(m)$ and $H(.)$ is a hash-and-pad function, and a proof of its validity $proof_i\ (for\ all\ u \in [1, k], log_{v_u} v_{u,i} = log_x s_i)$.

- The public key (N, e), a message m, a list $s_1, \dots, s_l$ of signature shares, for each $u \in [1, k]$ the list $v_u, v_{u,1}, \dots, v_{u,l}$ of verification keys and a list of "proofs" of validity ($proof_1, \dots, proof_l$ are seen as input, and the combining procedure may provide a signature as a result.

- Assume that the verification process takes (N,e) as an input public key, m as the message, and s as the signature and outputs a bit b indicating if the signature is accurate or not.

Then, in order to conduct dynamic auditing, the auditing technique is carried out by adding an authenticated data structure; In this experiment, the MDHT from the study [16] is used. Both the MDHT-based auditing system and the PDP-based skip list encounter a variety of difficulties throughout the verification and updating process, such as a considerable communication overhead and high TPA computation costs [17,18]. These challenges may be broken down into many categories. Therefore, by utilising the Index Hash Table (IHT), which is a method for verifying data blocks that was proposed by Zhu et al. [19], the modifications to data blocks are tracked, and the hash values of each block are created. This is done so that the Index Hash Table (IHT) can be utilised. IHT are useless for updating procedures like as insertion anddeletion because of the nature of the sequence, this causes the average elements to be adjusted, resulting in N/2, where N is the total number of blocks.

In addition, the block numbers (Bi) of the appropriate blocks are altered whenever an insertion or deletion operation is performed, which causes a regeneration of block tags. IHT will, in the end, become inefficient as a consequence of the high user processing expenses and communication overhead associated with this procedure. As a direct consequence of this, a brand new data format known as MDHT has been developed as a solution to the issues caused by IHT and to enhance the efficiency of auditing. In a manner similar to that of IHT, the TPA in MDHT is used to keep track of the data that users of the most recent Version of Information (VI) provide during the auditing process. The MDHT is comprised of a number of essential components, two of which are the file and block elements. Each file element consists of the File identifier, which is denoted by (IDi), and the index number, which is denoted by (NO_i) of the file that has been supplied (for (e.g. F_i). The files are retained in an array-like structure using pointers that identify the first block element. The files are structured by a linked list, with the element of the matching file acting as the header node.

File operations and block operations are the two categories that make up MDHT's operations. Both of these file operations include activities such as searching, adding, editing, and deleting files. According to the index, elements may be found in files by searching through them, and other operations can alter both elements and blocks for usage in files and other objects. Elements can also be located in files by searching through them. A linked list containing the pertinent block elements is created once the file items are inserted into the arrays. Both of these processes take place inside the file. It is possible to modify both the files and the block elements themselves, and the delete operation may be used to remove files and the components included inside them from a linked list.

The usage of linked lists with DHT greatly facilitates the insertion and deletion of blocks in IHT. The hash values of VI entries in DHT will also be unaffected by blocks of insertion and deletion in IHT. This is because IHT and DHT use different hashing algorithms. When opposed to IHT, MDHT helps CSP significantly cut down on its communication overhead as well as its computation expenditures throughout the updating process. The search operation cost is greater for DHT than it is for IHT during the whole of the verification process. This is due to the fact that DHT is unable to disregard the material effect throughout the entirety of the verification time. The verification time required by the MDHT system has been tested and found to be much shorter than that required by the IHT.

4. Experimental Analysis

By handling this task, TPA reduces the amount of computational effort required from the user by validating the DI in the cloud. Numerous studies have examined how successful the integrity check is, and attempts have been made to reduce computing costs. According to this analysis, MDHT lowers computational expenses. The tests are created on a personal computer with an Intel Core i5 CPU running at 2.2 GHz with 8.00 GB of random access memory using Java 1.8, Netbeans 8.2, and MySQL 8. (RAM). This section examines the proposed method for treating MRSA with MDHT, considering a range of possible computational costs, and contrasts it with the methodology that is currently being used. In the next section, table 1, you will find a listing of 150 separate files for the evaluation of the recommended method. These files range in kind, count, and source. There is a difference in the dimensions of each file, and the sum of all of them is one and a half gigabytes.

Table 1 A group of documents used to assess 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

In order to evaluate the usefulness of the proposed method in light of the data shown in the table below, a number of tests have been carried out (Table 2). It includes 25 unique files that range in size from 25 different extremes so that the features of upload speed and encryption time may be tested in a sequential manner.

Table 2 Measures for the proposed MDHT system's timing

S. No.	**Type of File**	**Size of File (KB)**	**Upload Time (millisecond)**	**Key Generate Time (millisecond)**	**Encryption Time (millisecond)**
1	“Text”	10	155	98	137
2	“Text”	14	169	105	142
3	“Text”	16	171	124	154
4	“Text”	22	186	134	162
5	“Text”	25	193	150	180
6	“Text”	23	198	168	187
7	“Text”	31	204	184	194
8	“WORD”	37	211	207	205
9	“WORD”	125	213	219	213
10	“WORD”	1075	225	229	222
11	“Photo”	1248	236	234	233
12	“Photo”	1551	248	247	239
13	“Photo”	18464	259	258	245
14	“Photo”	33123	264	262	256
15	“Photo”	37420	275	266	261
16	“Photo”	72748	289	287	270
17	“Photo”	1154490	297	291	275
18	“Photo”	1155207	341	302	281
19	“Mp3”	3507764	362	318	286
20	“Mp4”	5120428	377	326	290

21	“Mp3”	10325038	385	334	293
22	“Mp4”	20570492	396	349	297
23	“Mp3”	21674900	407	352	310
24	“Video Mp4”	27233473	420	359	334
25	“Video Mp4”	2733537	436	370	342

The experimental setup is evaluated in order to establish how effectively it functions by using various time metrics, such as the upload time and the encryption time. The amount of time necessary to load a file into CS is referred to as the upload time. This time period is often recognised as the interval between the beginning of the process of uploading a file and its completion. The results of the comparison of upload times for files of varying sizes are summarised in Table 2, and the corresponding graph can be seen below (Fig 3).

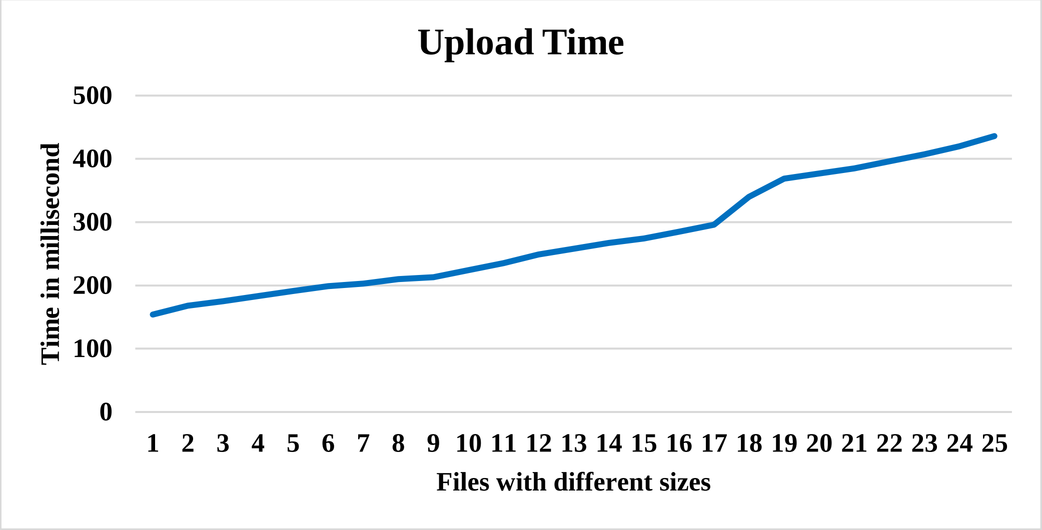

Fig 3 Uploading time for files of various sizes

Calculating the encryption keys for each file, which are subsequently produced and encrypted using the MD5 method, is the purpose of the second set of trials. The findings of the second set of investigations may be found in Table 2 and shown in Figure 4.

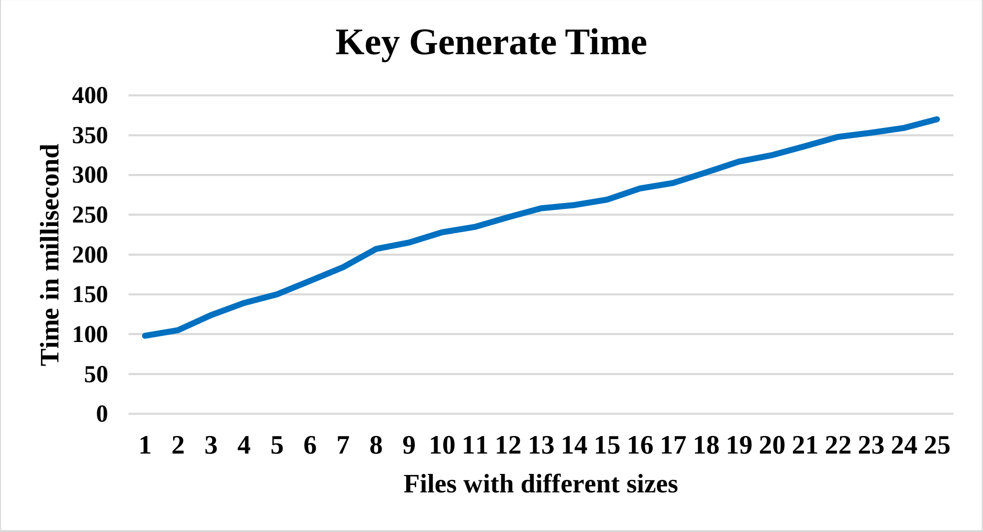

Fig 4 Time Required for Key Generation for a Set of Files with Various Sizes

The length of time required to encrypt each block included in a document before the document can be sent out for review is referred to as the encryption time. The following graph demonstrates how long it takes for the MDHT algorithm to encrypt files of varying sizes when they are processed using the recommended method (Fig 5).

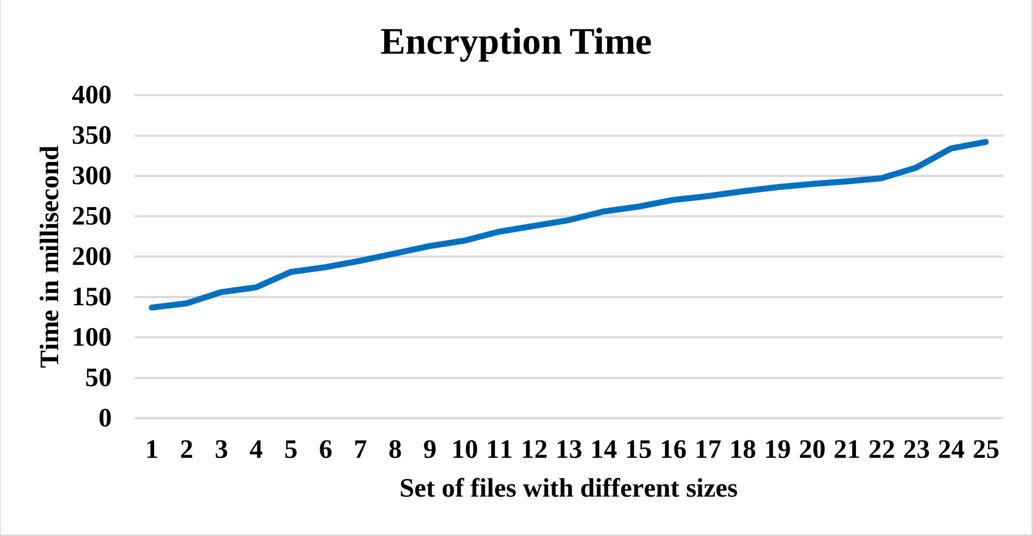

Fig 5 Encryption Time for Files with Different Sizes

The data that were taken and tabulated during the auditing process for the file blocks are included in the table that follows (Table 3) and the graph that follows (Fig 6) below. This process comprises activities such as challenging, inspecting blocks for modification, and other similar activities.

Table 3 Auditing Periods for Different 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 MDHT 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

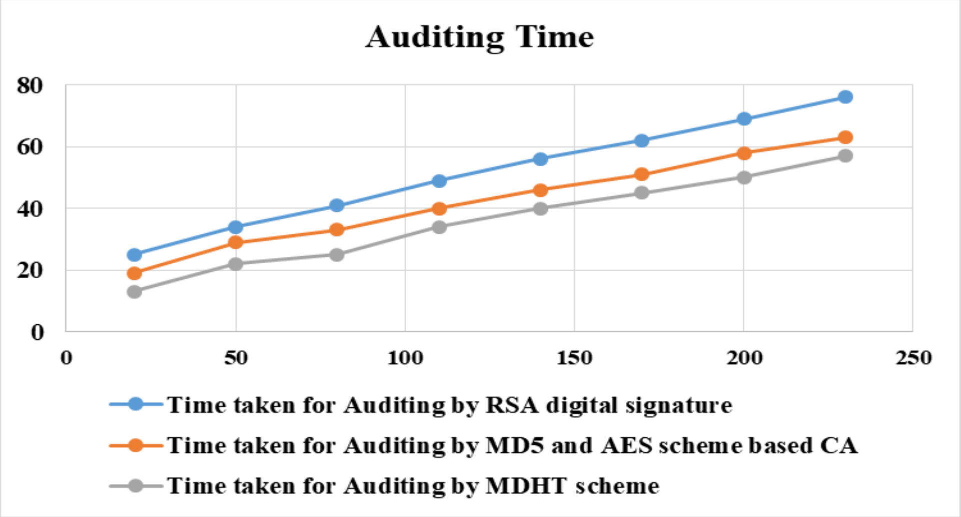

Fig 6 Auditing Time for various CA methods

Last but not least, a total of 150 unique files of varying sizes were evaluated for CA accuracy and then submitted to the specified procedures for evaluation. The vast majority of files have been corrupted as a result of the use of a hacking tool, and CA asserts that they checked the integrity of the whole batch of data before properly recognising the changes.

5. Conclusion

In recent years, the paradigm of cloud computing has emerged as the standard in the realm of computer services. This may be attributed to the cloud's adaptable processing capabilities as well as its large storage capacity. Customers have the option to store their data in the cloud, manage it from a distant location, and retrieve it from the cloud, which is one of the benefits of the cloud storage service. However, in order to make improvements to the services provided to consumers, it is necessary to tackle a number of research challenges and obstacles that are associated with the CSP. The users' concerns about the accuracy, availability, and confidentiality of personal data, as well as their perceptions that the data is accessible, are characteristics of some of the issues. The consumers' impressions of the data's accessibility are indicative of other issues. A publicly available auditing mechanism is required to assuage consumers' concerns over the security of their data stored in the cloud. This research study presents an innovative method of authentication that may be used to better strengthen the safety of cloud-based data storage. The MDHT authentication mechanism is used in this research to ensure the safety of cloud data. In the current investigation, a TPA is used in conjunction with an MDHT to conduct an auditing operation on data owners. This is done while ensuring that the data's confidentiality is preserved. The objectives of the proposed protocol's design are for it to be audible to the general public, to store information accurately, to increase data availability, to keep data private, and to be as efficient as possible. The information is protected using MRSA encryption. The bulk of the existing approaches do not provide enough support for data dynamic processes, public data auditing, and data freshness as the protocol that has been provided does. In further work, the MDHT algorithm that was proposed will be improved in order to audit for medical cloud data and to deliver additional security by making use of a number of different auditing approaches.

Reference

[1] Wang, B., Li, B. and Li, H., 2015. Panda: Public auditing for shared data with efficient user revocation in the cloud. IEEE Transactions on services computing, 8(1), pp.92-106.

[2] Fu, A., Yu, S., Zhang, Y., Wang, H. and Huang, C., 2017. NPP: a new privacy-aware public auditing scheme for cloud data sharing with group users. IEEE Transactions on Big Data.

[3] Liu, C., Chen, J., Yang, L.T., Zhang, X., Yang, C., Ranjan, R. and Kotagiri, R., 2014. Authorized public auditing of dynamic big data storage on cloud with efficient verifiable fine-grained updates. IEEE Transactions on Parallel and Distributed Systems, 25(9), pp.2234-2244.

[4] Liu, C., Ranjan, R., Yang, C., Zhang, X., Wang, L. and Chen, J., 2015. MuR-DPA: Top-down levelled multi-replica merkle hash tree based secure public auditing for dynamic big data storage on cloud. IEEE Transactions on Computers, 64(9), pp.2609-2622.

[5] Thangavel, M. and Varalakshmi, P., 2018. Enhanced DNA and ElGamal cryptosystem for secure data storage and retrieval in cloud. Cluster Computing, 21(2), pp.1411-1437.

[6] Yuan, J. and Yu, S., 2015. Public integrity auditing for dynamic data sharing with multiuser modification. IEEE Transactions on Information Forensics and Security, 10(8), pp.1717-1726.

[7] Youn, T.Y., Chang, K.Y., Rhee, K.H. and Shin, S.U., 2018. Efficient client-side deduplication of encrypted data with public auditing in cloud storage. IEEE Access, 6, pp.26578-26587.

[8] Tian, H., Nan, F., Jiang, H., Chang, C.C., Ning, J. and Huang, Y., 2019. Public auditing for shared cloud data with efficient and secure group management. Information Sciences, 472, pp.107-125.

[9] Yu, J., Ren, K., Wang, C. and Varadharajan, V., 2015. Enabling cloud storage auditing with key-exposure resistance. IEEE Transactions on Information forensics and security, 10(6), pp.1167-1179.

[10] Shen, J., Shen, J., Chen, X., Huang, X. and Susilo, W., 2017. An efficient public auditing protocol with novel dynamic structure for cloud data. IEEE Transactions on Information Forensics and Security, 12(10), pp.2402-2415.

[11] D. Chattaraj, M. Sarma, and A.K. Das, “A new two-server authentication and key agreement protocol for accessing secure cloud services”, *Computer Networks*, vol.131, pp.144-164, 2018.

[12] J. Brogan, I. Baskaran, and N. Ramachandran, (2018). “Authenticating health activity data using distributed ledger technologies”. *Computational and Structural Biotechnology Journal*, *16*, 257-266.

[13] Razaque, and S.S. Rizvi, “Triangular data privacy-preserving model for authenticating all key stakeholders in a cloud environment”, Computers & Security, vol.62, pp.328-347, 2016.

[14] H. Zhang, and T. Tu "Dynamic Outsourced Auditing Services for Cloud Storage Based on Batch-Leaves-Authenticated Merkle Hash Tree." *IEEE Transactions on Services Computing*, 2017.

[15] G. Sharma, and S. Kalra, “Identity based secure authentication scheme based on quantum key distribution for cloud computing”, *Peer-to-Peer Networking and Applications*, vol.11, no.2, pp.220-234, 2018.

[16] Tian, H., Chen, Y., Chang, C.C., Jiang, H., Huang, Y., Chen, Y. and Liu, J., 2017. Dynamic-hash-table based public auditing for secure cloud storage. IEEE Transactions on Services Computing, 10(5), pp.701-714.

[17] Erway, C.C., Küpçü, A., Papamanthou, C. and Tamassia, R., 2015. Dynamic provable data possession. ACM Transactions on Information and System Security (TISSEC), 17(4), p.15.

[18] Wang, Q., Wang, C., Ren, K., Lou, W. and Li, J., 2011. Enabling public auditability and data dynamics for storage security in cloud computing. IEEE transactions on parallel and distributed systems, 22(5), pp.847-859.

[19] Zhu, Y., Ahn, G.J., Hu, H., Yau, S.S., An, H.G. and Hu, C.J., 2013. Dynamic audit services for outsourced storages in clouds. IEEE Transactions on Services Computing, 6(2), pp.227-238.