Modified Dynamic Hash Table with Threshold RSA for Dynamic and Public Auditing on Cloud Data

T. Kishore Babu,.Guruprakash C D

Abstract: ***Cloud storage is one of the major application in the cloud, which can provide the on-demand outsourcing data service for both organizations as well as individuals. The Data Integrity (DI) check in the cloud is applied by the user to ensure the integrity of data. The Third Party Auditing (TPA) technique is later introduced to check the cloud DI. Many research has been carried out in the public auditing to minimize the computation cost of the integrity check. The most existing method involves in lack of security and low computation overhead. In this research, the Modified Dynamic Hash Table with threshold Rivest, Shamir, and Adelman Algorithm (RSA) algorithm (MDHT-RSA) is proposed to improve the security and reduce the computation cost. The threshold RSA cryptography system increase the security by generating the secret key to the user and reduce the computation cost. The Modified Dynamic Hash Table (MDHT) is used to record the data information for dynamic auditing, which is located in the TPA. The MDHT is differed from the Dynamic hash table, that the MDHT doesn't contain the tag block whereas the dynamic hash table has the tag block. The MDHT-RSA is analyzed with the computation cost and compared with existing method. The experimental result proved that the MDHT-RSA method has low computation cost than state-of-art method in public auditing. The verification cost of the MDHT-RSA is 1.3 s while a state-of-art method DHT-PA has the 1.35 s for the 200 blocks of data.***

Index Terms**: *Cloud storage, Modified Dynamic Hash Table algorithm, Public Auditing, Tag Block and Third Party Auditing.***

I. INTRODUCTION

Many people can work on the same resource and easily share their data with each other in a cloud by using sharing and data storage services such as Google Drive. After storing the shared data in the cloud by user, the other people in a group can able to modify, access and share the latest version of data to another group of people [1]. In order to avoid the profit losses or to maintain the reputation among user, cloud owner may do the data error accidents in the worst case [2]. There is still a hesitation and discussion on the usage of cloud, even though proliferation and development of cloud computing are rapid, because data security is the major concern of user in cloud environment [3]. The users are unable to move their valuable data from the cloud, once the user loses their direct control on data, Specially in public cloud with multi-tendency and high consolidation [4]. The ordinary user's rarely accessed data may be neglect to keep or delete by Cloud Service Providers (CSPs) for saving more storage space, which is considered as most severe case [5].

With the concern on DI of cloud storage services, users wish to have a way of auditing the cloud server to ensure that the server stores all their latest data without any corruption [6]. The integrity of their data should be guaranteed by clients in the cloud storage system, but user cannot eliminate the weak cloud servers, which are vulnerable to security threats [7]. The three main objectives of security are integrity, availability and confidentiality, where integrity can be assured by auditing the cloud data i.e., verification of DI from an external party, has been an extensively investigated research problem in recent years [8]. In cloud, the DI is periodically check by introducing the TPA to help the end users for reducing the computational burden [9]. But, most of existing methods face challenges like high computation cost of TPA and still there is a need for improving the security of TPA [10]. To overcome the issues of security and computation cost, this research works designed an MDHT-RSA. The proposed method is analyzed and compared with the existing dynamic hash table method. The threshold RSA is designed because the algorithm is easy to implement, secure and also ensured the fast computation of MDHT with threshold RSA signature. There are no strict pre-conditions in RSA and also applied to almost all circumstances where secret sharing signature is required.

The organization of the paper is in the form of Related Works on recent techniques used to secure the data in section II, the proposed method is explained in section III, Experimental result illustrated in section IV. The conclusion of this research work is made in section V.

II. RELATED WORKS

The process of verifying the DI in the cloud is very difficult due to the increasing number of data in the cloud. The TPA auditing technique is used effectively to guarantee the DI and data encryption system is used in the TPA to secure the data. Recent method involves in the TPA aried [11-15] in this section to analyze its performance.

Anbuchelian, et al. [11] established secure cryptography hashing algorithm in the TPA to improve the security.

Revised Manuscript Received on September 2, 2019.

T. Kishore Babu, Department of CSE, Visvesvaraya Technological University, Gulbarga, India.

Dr. Guruprakash C D, Department of CSE, Sri Siddharatha Institute of Engineering & Technology, Maraluru, Tumakuru, India

After uploading the data, the user was provided with a private and public key for trustworthy retrieval of the data file, where modified RSA algorithm was used to generate these keys. The data files were effectively audit by using a multilevel hash tree algorithm occasionally. This method was tested with attacks and this provided the privacy against the attacks. The complexity is high for developed hash tree function, which needs more time for the auditing process.

Cheng Guo, et al. [12] generated a constant-size key to support the flexible delegation of decryption for cipher texts by key-aggregate authentication cryptosystem. The expense of this scheme was stable and solved the problem of key-leakage to share the data. The message in authentication key was denied to access and unable to copy the data, which was proven by this method. This showed that the method achieved secure data sharing and leakage-resilient in dynamic cloud storage. The developed methodology provides insecure data sharing in the condition of one to one solution.

Yue Zhang, et al. [13] developed a storage auditing scheme for key generation and method for updating the private key to achieve highly-efficient user revocation method. The total number of file blocks were totally independent of this revocation method which was possessed by revoked user in cloud. The private keys of non-revoked user group were just updated by user revocation method than authenticators of revoked user, which was observed by this technique. The security and computation cost of the method are needed to be optimized.

Daeyeong Kim, et al. [14] developed a public auditing scheme to provide the data privacy and integrity for the educational multimedia data. This auditing method supported fully dynamic data as well as protect the data against both the untrusted cloud server and the TPA under the loss and tamper attacks by using the random values and homomorphic hash function. The data security was preserved against TPA and cloud by combining the data block of homomorphic authenticator and data owner with random values. The computation cost between the user and the TPA is high because the protocol is needed to ensure security.

Hui Tian, et al. [15] designed a Dynamic Hash Table (DHT) which was located at TPA for recording the data property information for dynamic public auditing. DHT method reduced the communication overhead and computation cost by migrating the information to TPA from CSP. According to public key, the homomorphic authenticator was combined with random masking which was generated by TPA to support the privacy preservation. The batch auditing was achieved by employing the bilinear maps and Boneh-Lynn-Shacham (BLS) signature. When compared with previous scheme, the DHT method reduces the storage costs, communication and computation cost by achieving secured auditing in clouds. The table size can be reduced to improve the effectiveness and decreasing the computation cost.

A. Problem Statement

The traditional cloud data storage service includes numerous challenging design issues, which have a profound influence on the security and performance of the overall system. The significant issues of cloud auditing process is addressed in the following sections.

- The privacy preserving of cloud user's data is a significant task during the whole auditing process. But, the cloud server is not a fully trusted entity and outsourced data in cloud may revealed and significant information may have leaked.
- In complex methods, the computation overheads of TPA increases, which will automatically increase the computation time of encryption systems.
- Multi-tenancy implies the sharing of the application resources by more than one user. Hence, lack of confidentiality occurs due to multi-tenancy.
- In data auditing process, the traditional methods used the number of Meta data, which is stored in a huge dataset, and consumed more storage space.

III. PROPOSED METHOD

In the cloud, the DI is checked by auditing process, which will reduce the user's computation load. The security of cloud data is improved by verifying the DI in cloud using various existing methods. In this research work, MDHT-RSA algorithm is developed to reduce the computation cost and also to increase the security of outsourced data. The RSA cryptographic system is used to secure the data and MDHT to store the records of the dynamic changes in the cloud.

A. System Model

Figure 1 shows the auditing system used in this research work for cloud storage, which consists of TPA, cloud users and Cloud Storage Service (CSS). The cloud users are the data owners, who have a bulk of data to be stored in CSS. A massive amount of computational resources and storage space are effectively handled by CSP, which is used to manage the CSS. The CSP managed and maintained the user's data which is stored in CSS, where the user can dynamically update or access their remote data whenever they need. Most of user are lacked in auditing the outsourced data due to less expert and capabilities in this process, which are effectively handled by TPA.

Fig. 1. The system architecture of TPA

1) Data Owner:

The data are stored in the cloud by the data owner, which can interact with CSP for managing their stored data on cloud.

Without having the local copy of data, the security of data should be periodically verified by the owner. If they don't have time or resource to verify the security, these jobs can be assigned to trusted TPA by the data owner.

2) Cloud Service Provider:

The distributed cloud storage servers are built with major resources and managed by CSP, which is also offered software or storage services to the end user over the Internet. The major role of the CSP is to response the verifier queries and maintain the data properly.

3) Third Party Auditor:

The vulnerability of user's privacy data is improved by the auditing process and the DI which is stored in the cloud is checked by TPA. The outsourced data can be managed or monitored under the delegation of data owner which is considered as a major responsibility of TPA. Whenever TPA receives the DI verification request from the data owner, immediately TPA send the challenge request to the cloud server. The TPA receives the proper response from cloud auditing task and sends the result back to the user. In proposed MDHT, these three entities performed the following activities such as:

1. The pre-processed files are sent to CSP by the data owner and also sends the metadata to TPA or user can keep locally for verifying the integrity later.
2. Then, files are stored by CSP, where metadata is stored by TPA.
3. The validity of response is checked and generated a challenge by verifier (either TPA or user), then send it to CSP. If the response is valid, it will return 1, otherwise return 0.
4. The response is generated by CSP, then sends to verifier.
5. An update request is generated by data owner and send to CSP for verification.
6. The CSP updates data and generates an update response based on data owner's requests.
7. When the insertion or modification process occurred, the updated metadata should be stored by TPA.

Even though TPA is secure, there are also some challenges faced by these models because of CSP. These threat model are described as below.

B. Threat Model

In this work, consider the CSP as semi-honest, but it is a curious server for securing the end user's data. A passive attacker is a probabilistic polynomial time adversary, which is a traditional honest-but-curious model and follows the designed specification correctly. An ability-limited active attacker adversary is considered as semi-honest-but-curious model, which can able to modify the designed specifications such as insert, update, delete and return only a small portion of search results, instead of retrieving all the query results.

C. Bilinear Mapping

Let $G1, G2, G_T$ be three cyclic multiplicative groups with the prime order q . Let g_1, g_2 be the generators of groups An admissible pairing $e: G1 \times G2 \to GT$, which satisfies the following three properties:

– Bilinear: If $u \in G_1, v \in G_2$ and $a, b \in Z_q^*$, then $e(ua, vb) = e(u, v)ab$;

– Non-degenerate: There exists a $g_1 \in G_1, g_2 \in G_2$ such that $e(g_1, g_2) \neq 1$;

– Computable: If $u \in G_1, v \in G_2$ one can compute $e(u, v) \in G_T$ in polynomial time.

D. Modified Dynamic Hash Table

This is popular to introduce an authenticated data structure to achieve dynamic auditing and this DHT from the research [15] is used in this experiment. During the verification and updation process, the MHT-based auditing scheme and PDP based skip list face some challenges like large communication overhead and heavy computational costs of TPA [16,17]. Thus, the changes in data blocks are recorded and the hash values of every block are generated by using Index Hash Table (IHT), which is proposed by Zhu et al. [18] in verification process. The IHT are inefficient for updation process like insertion and deletion because of sequence structure, this leads to the adjustments in average elements i.e. $N/2$, where N is a total number of blocks.

In addition, the regeneration of block tags is occurred by the modification of block numbers (Bi) of that corresponding blocks during insertion or deletion process. This process will eventually cause high computation costs of user and communication overhead, which leads to inefficient of IHT. Therefore, a new data structure called DHT is developed to overcome the issues of IHT and provides better auditing efficiency. Like IHT, the latest Version of Information (VI) user's data is tracked by employing the TPA in DHT for auditing process. There are two basic elements in DHT such as file and block elements. Each file element consists of the File identifier (ID_i) and the index number (NO_i) of the given file (e.g. F_i). The files are stored in array-like structure by using pointer that indicates the first block element, and every file is arranged by a linked list with the corresponding file element as the header node.

The DHT operations are classified into two types such as file operations and block operations, where search, insert, modify and delete mechanisms are carried out on both the file operations. According to index, the files can be searched for finding the location of elements, and manipulations on both elements for file and blocks are done by other operations. The file elements are inserted into arrays by using insertion and the linked list are constructed that contains corresponding block elements. The files and its elements can be deleted from the linked list using delete operation and modification can be carried on both the elements of files and blocks.

The insertion and deletion of blocks in IHT is significantly improved by using the linked lists and DHT. Further, the hash values of VI records in DHT will not be influenced by blocks of insertion and deletion in IHT. The communication overhead and computational costs of CSP are significantly reduced by MDHT-RSA in the updation process, when compared with IHT scheme. During the verification, the

search operation cost of DHT is more than IHT, because the material impact on whole verification time cannot be able to neglect by DHT. The verification time of MDHT scheme is validated and showed that it is substantially smaller than the IHT.

E. Threshold-Rivest, Shamir, and Adelman Algorithm (Threshold-RSA)

The detailed descriptions of these algorithm (RSA) of three phases are described in the following sections.

1) Setup Phase

Before storing the file in cloud, it should be pre-processed by user for ensuring the availability, DI and confidentiality of file.

- **Encoding:** The file is encoded by user to check the availability of data in cloud.
- **KeyGeneration:** In this algorithm, the user generates private and public key pair for the later processing the file in the propose system.
- **Encryption:** In case, the data owner wants to ensure the data Confidentiality, the users encrypts the data using public key cryptography.
- **MetadataGeneration:** The metadata for each block of file is computed by user for verifying the DI stored in cloud storage system.

2) Verification Phase:

Whenever the user wants to verify the data that is stored in the cloud servers, the verifier (either user himself or his arranged agent TPA) checks the DI without having the local copy of data through Challenge-Response Protocol. The verification phase consists of three methods described below.

- **Challenge:** A random challenge is created and send it to CSP by verifier to check the DI.
- **Response:** Once a challenge request received from the verifier, the integrity proof as response is generated by CSP based on the challenge and forward back to verifier.
- **Check Integrity:** The response received from CSP is compared with previously computed metadata and identified whether the updated proof is valid or not by verifier. To hold the Integrity, the response must be equal with the metadata otherwise it indicates data have corrupted.

3) Secret sharing RSA Threshold Algorithm

A new threshold RSA algorithm is developed by applying the digital signature to secret sharing scheme, that are as follows:

Digital signature scheme is a triple (KeyGen, Sign, Ver) of efficient algorithms.

- 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.

There are four major components presents in a RSA threshold signature scheme, that are described as below:

- Consider n as security parameters, generation of Q_N using k number of elements, singing servers as l,t as threshold parameters and ω is a random string, which are all taken as input for key generation algorithm and produce the outputs as (N,e) is a public key, where n is the size in bits of N, the private keys $d_1,...d_l$ only known by the correct server and for each $u \in [1,k]$ a list $v_u, v_{u,1} = v_u^{d_1},....., v_{u,l} = v_u^{dl} \bmod N$ of verification keys.
- The input of a share signature algorithm is (N,e), an index $1 \le i \le l$, the private key d_i and a message m; this outputs a signature share $s_i = x^{d_i} \bmod N, where\, x = H(m)$ and $H(.)$ is a hash-and-pad function, and a proof of its validity
- $proof_i\left(for\ all\ u \in [1,k], \log_{vu} v_{u,i} = \log_x s_i\right)$
- The public key (N,e), a message m, a list $s_1,...,s_l$ of signature shares, for each $u \in [1,k]$ the list $v_u, v_{u,1},....v_{u,l}$ of verification keys and a list $proof_1,...proof_l$ of validity proofs are considered as input and a signature s may be an output of combining algorithm.
- Consider, (N,e) as a input public key, m as message and s as signature for verification algorithm and it outputs a bit b indicating whether the signature is correct or not.

In this research work, verification process of the proposed dynamic auditing protocol is presented and also design consideration for efficiency and security are introduced which are explained in the following section.

F. Blockless verification

Without retrieving the original data, the public auditing is achieved and the block tags are generated for data blocks. In verification process, block tags are authenticated instead of original data blocks. In this research work, RSA and BLS are employed, but BLS preferred more for shorter length of each block tag.

G. Sampling verification:

Given the huge amount of data outsourced in the cloud, it is inadvisable to challenge all data blocks for checking the integrity. Instead of checking the whole data files, checking a small portion of data files is more suitable for CSP and TPA to achieve the high verification accuracy, is known as sampling verification. The previous studies have demonstrated the rationality and feasibility of this strategy. Consider t is a fraction of given data, which is corrupted, the detection probability of the verification by checking randomly sampled c blocks is $P = 1-(1-t)c$.

H. Privacy preservation:

The data proof A is generated for preventing the privacy leakage, the M sampled blocks with R random masking is provided by TPA and u is considered as a public key. Under DL assumption, the process of obtaining the user′s privacy data is computationally infeasible for TPA, even though it knows the $A, R,$ and u .

I. Data Freshness:

In existing works like DAP [19], data proof is computed by CSP, but the linear combination of sampled blocks is not computed, which is a major issue of CSP. Due to the continuous updation of dynamic data, dishonest behavior is easily identified in dynamic data auditing [20]. i.e. the CSP could not pass the verification without actually storing the latest version of the data in the dynamic audit. However, the archived data is not updated frequently, the CSP may indeed to have the chance to pass the verification. Therefore, in future, different and more proper auditing methods are developed for diverse data.

IV. Results and Discussion

The computation load of user is decreased by checking the DI in the cloud, and this process are carried out by TPA. Many research has been conducted to improve the effectiveness of the integrity check and reduce the computation cost. In this research, RSA encryption system is proposed for key generation and computation cost is reduced by MDHT. The experiments are implemented using JAVA 1.8 Netbeans 8.2 MySQL 8 on a computer with Intel Core i5 CPU 2.2 GHz with 8.00 GB RAM. In this section, the proposed MDHT-RSA method is analyzed with the different computation cost and compared with the existing method. To validate the efficiency of proposed MDHT-RSA method, the experiments are conducted on verification time, processing time, computation cost and searching time when compared with existing methods such as DHT-PA [15], IHT-PA [18] and DAP [19].

A. Computation cost in the setup phase

The computation cost of the setup phase is calculated for the existing and proposed method for the different block number. The values are shown in Table 1. The computation time of existing and proposed method is shown in Fig. (2).

Table I. Setup phase computation cost

Block Number	IHT-PA	DAP	DHT-PA	MDHT-RSA
10	0.33	0.2	0.18	0.16
15	0.46	0.25	0.23	0.21
40	1.02	0.97	0.92	0.87
60	1.54	1.24	1.2	1.07
80	2.14	1.82	1.73	1.63
100	2.46	2.18	2.12	2.11
120	2.62	2.72	2.57	2.49
140	2.82	3.17	3.02	3
160	4.24	3.72	3.51	3.45
180	4.98	4.12	3.74	3.72
200	5.31	4.78	3.67	3.49

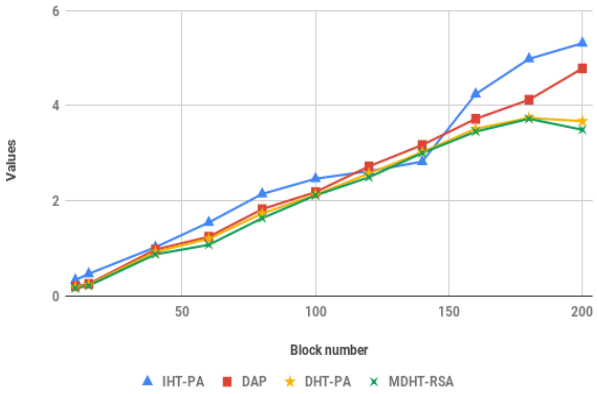

Fig. 2. Setup processing time for different blocks

This shows the computation cost is propositional to the block size. The proposed method has less computation time compared to that other existing method. The proposed MDHT-RSA has the computation time of 3.49 second for the 200 block size, while state-of-art method DHT-PA has the computation time of 3.67 seconds for same block size. When the block size is 10, the proposed MDHT-RSA achieved 0.16 seconds whereas the existing method IHT-PA achieved 0.33 seconds for same block size.

B. Searching Time

The verification of searching time of the existing and proposed method is measured in the different block size which are given in Table 2. Figure. (3) shows that the searching time of the public auditing methods in the different block size.

Table II. Search time for proposed MDHT-RSA

Block Number	IHT - PA	DAP	DHT- PA	MDHT-RSA
10	262	34	32	30
25	187	34	32	30
30	124	34	32	30
50	98	34	32	30
100	81	34	32	30
200	76	34	32	30
300	65	34	32	30
400	60	34	32	30
500	52	34	32	30

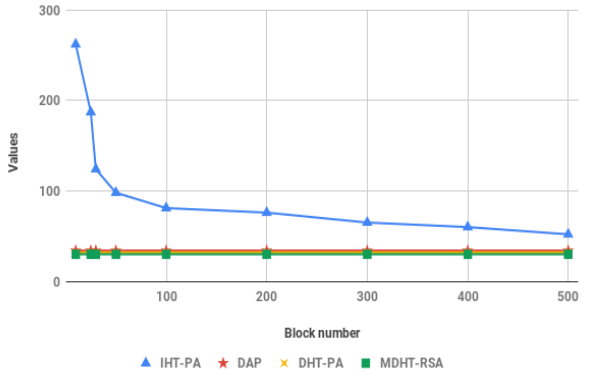

Fig. 3. Searching time of MDHT – RSA

The proposed MDHT-RSA method has a lower computation time compared to the other existing method. The searching time is low for the various block size and block size are increased, which decreases the searching time of the MDHT. The searching time during verification of the proposed MDHT-RSA is 30 ms for the 500 block size in KB, while the existing method has 32 ms for the same block size. In this searching time, all other methods are constant at certain point in block size, but IHT-PA alone varies the searching time depends on block size. For instance, IHT-PA achieved 26 ms for 10 block size, 76 ms for 100 block size and 52 ms for 500 block size.

C. Computation in Verification time

The verification time of the different block size is measured for the existing and proposed method, as shown in the Table 3 and Fig. (4).

Table III. Verification Time of MDHT – RSA

Block Number	IHT - PA	DAP	DHT-PA	MDHT-RSA
10	2.24	1.68	1.5	1.42
20	2.25	1.69	1.47	1.52
30	2.27	1.71	1.46	1.41
50	2.28	1.71	1.45	1.39
70	2.28	1.71	1.38	1.41
100	2.32	1.71	1.36	1.34
150	2.34	1.72	1.35	1.2
200	2.37	1.73	1.35	1.2

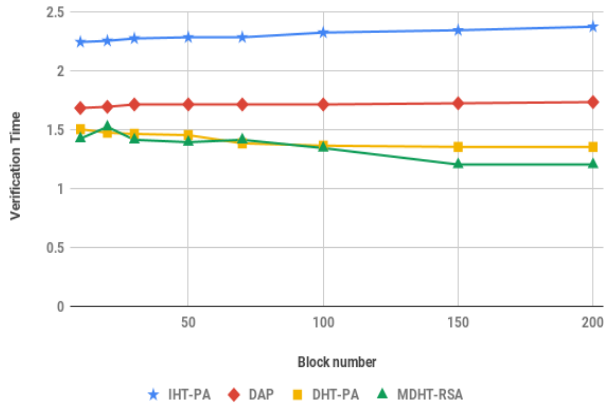

Fig. 4. Verification of MDHT-RSA for different block size

The proposed MDHT-RSA has the lower verification time compared to the other existing method. The verification time of the DHT-PA and MDHT-RSA methods has the much lower computational time than IHT-PA due to the significantly outweigh the disadvantage include by searching operation. When compared with other existing techniques, the proposed method achieved 1.2 sec for block number 200.

D. Operation time for Block Insertion

The computation cost of the block insertion for the existing and proposed method is calculated in this experimental setup. The computation cost for the different file size is measured varying 1 to 10 GB of data is measured and shown in Table 4 and Fig. (5).

Table IV. Block Insertion of MDHT-RSA

Block Number	IHT - PA	DAP	DHT-PA	MDHT-RSA
1	0.02	0.02	0.02	0.02
2	0.1	0.1	0.05	0.05
3	0.18	0.17	0.06	0.05
4	0.21	0.21	0.09	0.08
5	0.28	0.26	0.1	0.1
6	0.32	0.31	0.12	0.12
7	0.41	0.41	0.15	0.147
8	0.46	0.45	0.18	0.173
9	0.52	0.51	0.19	0.182
10	0.57	0.55	0.21	0.205

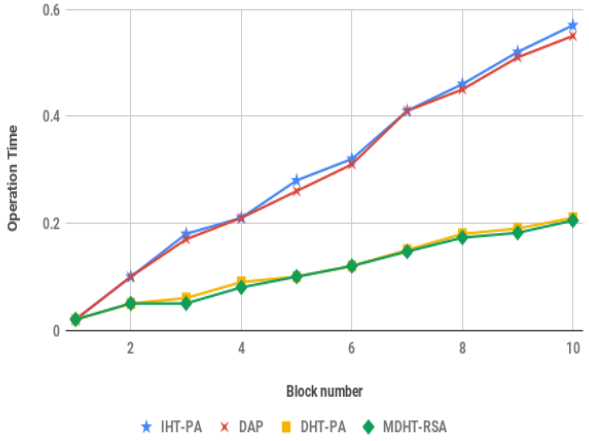

Fig. 5. Operation time for Block Insertion

The proposed MDHT-RSA has lower computation time compared to the other existing method. The MDHT-RSA has the computation time of 0.20 s compared to the state-of-art method which has 0.21 s for the file size 10 GB. When the size of the file is low, all the methods have the same computation time of 0.02 second. But, as the size of file increases, the computation time of existing methods are also increased.

E. Operation time for Block Updation

The computation time for the updating operation is measured for the existing and proposed method. The computation time of updating process for the different file size is measured and shown in Table 5 and Fig. (6).

Table V. Computation time for Block Updation

Block Number	IHT - PA	DAP	DHT- PA	MDHT-RSA
1	1.58	1.58	1.08	1.04
2	1.59	1.59	1.1	0.9
3	1.6	1.6	1.12	1.11
4	1.62	1.62	1.21	1.18
5	1.64	1.64	1.24	1.22
6	1.66	1.66	1.26	1.22
7	1.72	1.72	1.27	1.25
8	1.73	1.73	1.28	1.24
9	1.75	1.75	1.32	1.3
10	1.76	1.76	1.41	1.38

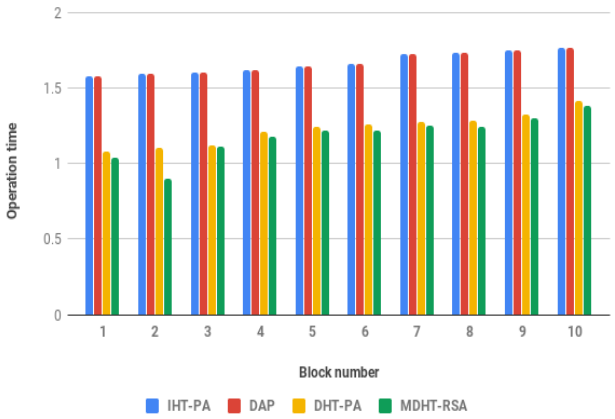

Fig. 6. Operation Time for Block Updation

This shows that the proposed MDHT-RSA has the lower computation time compared to the other existing method. The MDHT-RSA has the computation time of updating process is 1.38 s while existing method has computation time of 1.41 s for 10 GB file. The computation time for block updation are constant for both existing methods such as IHT-PA and DAP in all block size. But, the existing technique DHT-PA slightly varies from MDHT-RSA computation time for block updation.

F. Operation time for Block Deletion

The computation time of the block deletion is calculated for the different file size, which are analyzed for the existing and proposed method. The computation time is compared for the different methods in public auditing and shown in Table 6 and Fig. (7).

Table VI. Computation time for Block Deletion

Block Number	IHT - PA	DAP	DHT-PA	MDHT-RSA
1	0.02	0.02	0.02	0.02
2	0.1	0.1	0.05	0.05
3	0.18	0.17	0.06	0.05
4	0.21	0.21	0.09	0.07
5	0.28	0.26	0.1	0.1
6	0.32	0.31	0.12	0.115
7	0.41	0.41	0.15	0.142
8	0.46	0.45	0.18	0.171
9	0.52	0.51	0.19	0.196
10	0.57	0.55	0.21	0.208

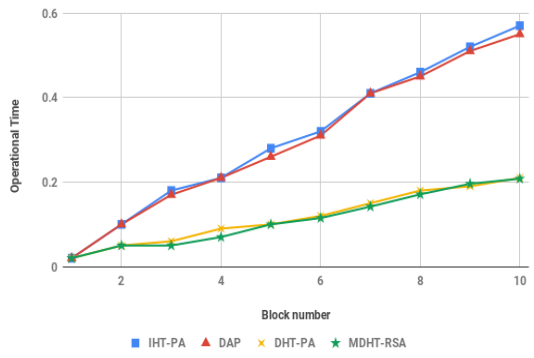

Fig. 7. Operational Time for Block Deletion of MDHT-RSA

The MDHT-RSA has the lower computation time compared to the other existing methods. The MDHT-RSA method has the computation time of 0.2 sec compared to the state-of-art method 0.21 sec. The deletion time of MDHT-RSA achieved 0.115 sec for block size 6, whereas the existing method IHT-PA achieved 0.32 sec for same block size. The MDHT-RSA didn't attain great performance in block deletion time for different block size. The method needs further improvements for achieving better performance in computation time for different block sizes.

G. Auditing Time for MDHT-RSA

The experiments also evaluate the performance of MDHT-RSA in the batch auditing scenario and compare it with IHT-PA and DAP. The experimental results are as shown in Table 7 and Fig. 8.

Table VII. Auditing Time of MDHT-RSA

Block Number	IHT - PA	DAP	DHT-PA	MDHT-RSA
1	1.95	1.81	1.58	1.04
2	1.92	1.79	1.59	0.98
3	1.86	1.76	1.6	1.09
4	1.83	1.75	1.62	1.15
5	1.79	1.7	1.67	1.21
6	1.76	1.69	1.65	1.23
7	1.72	1.66	1.75	1.22
8	1.73	1.64	1.74	1.26
9	1.75	1.67	1.79	1.31
10	1.76	1.69	1.76	1.39

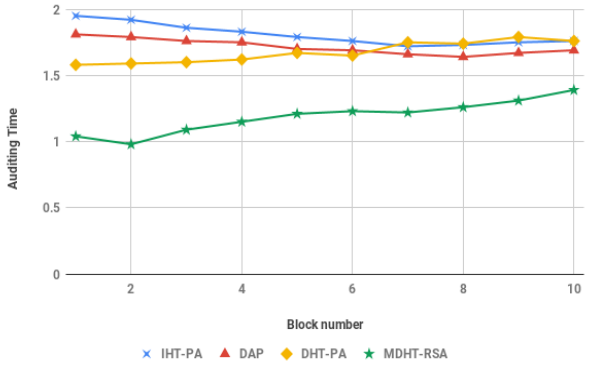

Fig. 8. Auditing Time for Proposed MDHT-RSA

The experimental results suggested that the batch auditing can't handle only verifications from multiple-users simultaneously, and while performing the individual auditing for multiple times, this MDHT-RSA reduces the computational costs of TPA. Also, the results proved that the batch auditing protocol in MDHT-RSA is more efficient than that in DAP, DHT-PA and IHT-PA. The MDHT-RSA has lower computation time compared to the other existing method. The security of the method is increased by using the secure key generation method. Hence, the proposed method can be applicable to practical use in the cloud auditing system.

V. Conclusion

Nowadays, the cloud storage system becomes popular for the different kinds of services storing, managing and processing the large amount of data. The data security is an important to concern in the cloud storage due to the development of different attacks on the cloud. The computation load of user is minimized by checking the DI of data using TPA in the cloud.

The various methods are developed for the TPA to check the DI without increasing the computation cost of the system. In this research work, the security is increased by RSA encryption algorithm and dynamic changes are recorded by MDHT which is used to reduce the computation cost of TPA in cloud system. MDHT is two-dimensional data structure is used instead of the DHT. The tag block is removed in the MDHT and this increase the computation cost of the system. The insertion and deletion are carried out by the identification block instead of tag block that helps to increase the efficiency. The cost of the MDHT is measured for the different file block and compared with existing method. The result shows that MDHT has lower computation cost compared to the other existing method in cloud auditing. The verification time of the MDHT-RSA is 1.2 sec while other existing methods such as DHT-PA and IHT-PA has the verification time of 1.35 sec and 2.37 sec. The results of MDHT-RSA for data insertion, updation, deletion and auditing process achieved 0.205 sec, 1.35 sec, 0.208 sec and 1.39 sec for block size 10. In future work, the proposed MDHT-RSA algorithm will be improved to audit for all types of cloud data by implementing different audit strategies.

REFERENCES

1. B. Wang, B. Li, and H. Li. (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. A. Fu, S. Yu, Y. Zhang, H. Wang, and C. Huang. (2017). NPP: a new privacy-aware public auditing scheme for cloud data sharing with group users. *IEEE Transactions on Big Data*.
3. C. Liu, J. Chen, L. T. Yang, X. Zhang, C. Yang, R. Ranjan, and R. Kotagiri. (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. C. Liu, R. Ranjan, C. Yang, X. Zhang, L. Wang, and J. Chen. (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. M. Thangavel and P. Varalakshmi. (2018). Enhanced DNA and ElGamal cryptosystem for secure data storage and retrieval in cloud. *Cluster Computing*, *21(2)*, pp. 1411-1437.
6. J. Yuan, and S. Yu. (2015). Public integrity auditing for dynamic data sharing with multiuser modification. *IEEE Transactions on Information Forensics and Security*, *10(8)*, pp. 1717-1726.
7. T. Y. Youn, K. Y. Chang, K. H. Rhee, and S. U. Shin. (2018). Efficient client-side deduplication of encrypted data with public auditing in cloud storage. *IEEE Access*, *6*, pp. 26578-26587.
8. H. Tian, F. Nan, H. Jiang, C. C. Chang, J. Ning, and Y. Huang. (2019). Public auditing for shared cloud data with efficient and secure group management. *Information Sciences*, *472*, pp. 107-125.
9. J. Yu, K. Ren, C. Wang, and V. Varadharajan. (2015). Enabling cloud storage auditing with key-exposure resistance. *IEEE Transactions on Information forensics and security*, *10(6)*, pp. 1167-1179.
10. J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo. (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. S. Anbuchelian, C. M. Sowmya, and C. Ramesh. (2017). Efficient and secure auditing scheme for privacy preserving data storage in cloud. *Cluster Computing*, pp. 1-9.
12. C. Guo, N. Luo, M. Z. A. Bhuiyan, Y. Jie, Y. Chen, B. Feng, and M. Alam. (2018). Key-aggregate authentication cryptosystem for data sharing in dynamic cloud storage. *Future Generation Computer Systems*, *84*, pp.190-199.
13. Y. Zhang, J. Yu, R. Hao, C. Wang, and K. Ren. (2018). Enabling efficient user revocation in identity-based cloud storage auditing for shared big data. *IEEE Transactions on Dependable and Secure Computing.*
14. D. Kim, H. Kwon, C. Hahn, and J. Hur. (2016). Privacy-preserving public auditing for educational multimedia data in cloud computing. *Multimedia Tools and Applications*, *75(21)*, pp.13077-13091.
15. H. Tian, Y. Chen, C. C. Chang, H. Jiang, Y. Huang, Y. Chen, and J. Liu. (2017). Dynamic-hash-table based public auditing for secure cloud storage. *IEEE Transactions on Services Computing*, *10(5)*, pp.701-714.
16. C. C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, (2015). Dynamic provable data possession. *ACM Transactions on Information and System Security (TISSEC)*, *17(4)*, pp. 15.
17. Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li. (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.
18. Y. Zhu, G. J. Ahn, H. Hu, S. S. Yau, H. G. An, and C. J. Hu. (2013). Dynamic audit services for outsourced storages in clouds. *IEEE Transactions on Services Computing*, *6(2)*, pp.227-238.
19. K. Yang, and X. Jia. (2013). An efficient and secure dynamic auditing protocol for data storage in cloud computing. *IEEE transactions on parallel and distributed systems*, *24(9)*, pp. 1717-1726.
20. H. Jin, H. Jiang, and K. Zhou. (2016). Dynamic and public auditing with fair arbitration for cloud data. *IEEE transactions on cloud computing*, *6(3)*, pp. 680-693.

AUTHORS PROFILE

Mr. Tunuguntla Kishore Babu is a research scholar from Visvesvaraya Technological University (VTU), Belagavi, karnataka in computer science and engineering and currently working as Assistant professor in Dept. of CSE in Andhra Loyola institute of engineering and technology, Vijayawada and has experience of 7 years in teaching and 3 years in research published various papers in international conferences/ journals. Area of interest Cloud Computing, Data mining and information security.

Dr. Guruprakash C D is currently working as professor in department of computer science and engineering SSIT, Tumkur Karnataka and has experience of 17 years in teaching, published various papers in international conferences/ journals. Area of interest Networking.