IJARSCT

ISSN (Online) 2581-9429

Impact Factor: 6.252

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Volume 2, Issue 1, June 2022

# Integrity Auditing for Multi-Copy in Cloud Storage Based on Red-Black Tree

**Mr. R.Arunachalam[1], Deepika Thennarasu[2], Dhanasri Natarajan[3], Madhumitha Baskar[4]**

Assistant Professor, Department of Computer Science and Engineering[1]

Students, Department of Computer Science and Engineering[2,3,4]

Anjalai Ammal Mahalingam Engineering College, Kovilvenni, Tiruvarur, Tamil Nadu, India

**Abstract:** *With the rapid development of cloud storage, cloud users are willing to store data in the cloud storage system, and at the same time, the requirements for the security, integrity, and availability of data storage are getting higher and higher. Although many cloud audit schemes have been proposed, the data storage overhead is too large and the data cannot be dynamically updated efficiently when most of the schemes are in use. In order to solve these problems, a cloud audit scheme for multi-copy dynamic data integrity based on red-black tree full nodes is proposed. This scheme uses ID-based key authentication, and improves the classic Merkel hash tree MHT to achieve multi-copy storage and dynamic data manipulation, which improves the efficiency of real-time dynamic data update (insertion, deletion, modification). The third-party audit organization replaces users to verify the integrity of data stored on remote cloud servers, which reduces the computing overhead and system communication overhead. The security analysis proves that the security model based on the CDH problem and the DL problem is safe. Judging from the results of the simulation experiment, the scheme is safe and efficient.*

**Keywords:** Cloud storage, data integrity auditing, red-black tree, dynamic data update.

## I. INTRODUCTION

The introduction of cloud computing and cloud storage gives a new scheme for user administration and sharing in the face of growing user management and sharing needs. Users can get enough storage for a lesser price while also benefiting from highly concentrated computing resources [1]–[3]. When consumers utilize cloud storage services, they typically upload their files to the cloud and save them on a remote cloud server. The local copy will be destroyed in order to conserve local storage resources. In this manner, there are two concealed threats. The first is the absence of control over the data's confidentiality and integrity, and the second is the difficulty of recovering the data if the local copy is deleted [4]–[6. The researchers advised that users encrypt data before outsourcing and sending it to a remote cloud server to solve these difficulties [7]–[9]. People also consider storing several copies of the original data to improve data availability and recoverability. If a portion of a user's data is damaged, just one copy of the data is required to restore the data correctly, and the data in the cloud remains unaffected [10]–[12]. The concept of proven data possession was proposed by Ateniese et al. [18]. (PDP). Users can successfully verify the integrity of cloud server data without downloading the entire file, and they presented and validated a safe PDP approach based on homomorphic, linear verification.

It is solely for static data that.it proposes. Later researchers [16], [17] have found that in the situation of rising data, dynamic data operation is also a crucial study point. Wang et al. [19] suggested a Merkle hash tree-based dynamic data method. Luo et al. [13] improved the polynomial-based authentication tag by employing the Shamir secret sharing idea. Barsoum et al. [20] developed a map-based provable multi-copy dynamic data possession (MB-PMDDP) system based on the PDP paradigm. Users can manipulate data in real time and save fewer copies. Although security is ensured, any insert or delete operations will necessitate recalculating the label and the position of the operation block, resulting in substantial calculation costs. Following that, the system [21] was devised The approach of dynamic updating efficiency was substantially enhanced as a result of this. Yang et al. [22] suggested a public cloud audit method for dynamic updating and revocation of user data at the same time, but it did not solve the problem of dynamic revocation at any time. Min et al. [23] introduced a spatiotemporal chaos-based integrity verification scheme that facilitates dynamic data analysis, information blinding, and prevents third parties from exposing user data privacy. [24] Min et al. proposed a binary balanced tree-based data integrity verification scheme Data update efficiency has substantially increased, and it

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, June 2022
Impact Factor: 6.252

has also provided us with a study direction. Curtmola et al. [25] developed an audit technique based on RSA signatures to enable public audit of multi-copy storage, but it does not offer dynamic data analysis. Barsoum and A. Barsoum and Hasan [26] presented a DPDP structure to overcome this problem, which provides dynamic data copy, block change, quick deletion, and fast addition on the cloud server. Shacham and Waters [27] developed a remote public data integrity verification technique and a private data integrity verification scheme based on the pseudo-random function and BLS signature. Shen et al. [28] presented a doubly linked list dynamic structure that may effectively test whether the data saved by the cloud service provider is safe. Most schemes used to generate their public and private keys using the certificate issuing authority PKI, which was a big overhead for certificate issuing authorities To reduce the expense of certificate administration, following methods began to use ID-based signatures [14], [15]. Shen et al. [29] suggested an identity-based remote data integrity audit approach that enables data sharing while also simplifying cumbersome certificate management. Then Zhang et al. [30] introduced a comprehensive dynamic multi-copy session approach (MR-DPDP). They employed the Merkel hash tree's rank characteristics to validate the data in this protocol, but because all of its copies are stored on a cloud storage server, the performance of multiple copies is useless. Li et al. [31] proposed delivering all copies to separate cloud storage servers and auditing the integrity of all copies by homomorphic verifiable homomorphism. tags. Other features of remote data integrity audits have been examined, such as privacy protection [32] and data de duplication [33]. Although many copies of a batch update can be stored, the audit cost has always been expensive and the efficiency has been low As a result, we'll look at how to optimize the storage structure and reduce dynamic operation overhead in order to create a dynamic multi-copy integrity audit scheme that works. Specifically, the main contributions of this paper can be summarized as follows:

1. The red-black tree data structure is designed to store data in a way that is favorable to efficient data storage, enhances data update efficiency, minimizes data storage consumption, and standardizes data administration. TPA and CSP spoofing attacks should be avoided at all costs.
2. A digital signature scheme based on ID is designed, which improves the security of the private key and the secret key pair, and avoids the resource consumption caused by the use of PKI as the basic public and private key generation, and effectively reduces the cost.
3. The user can select the amount of copies to be made, encrypt the copies, and then generate signatures for the encrypted files. If the user wants the original file, he can easily convert the encrypted file to the valid original file. This technique efficiently protects data sharing and storage security while also enabling remote data integrity auditing. The scheme's safety was assessed by broad experiments, and its effectiveness was demonstrated through concrete implementation. The findings revealed that the suggested strategy performed as expected in terms of safety and efficiency..

## II. LITERATURE REVIEW

1. The capacity of the Indonesian healthcare system to respond to COVID-19 utilizing Cloud computing to study in 2021 was written by Mahendradhata, Y., Solikha, D.A., and others, and it has merits such as It was less complicated and had fewer flaws. Only one file was saved.
2. The title of the book, A review of problems and possibilities in machine learning for health applying Machine learning to study in 2020, was written by M. Ghassemi, R. Ranganath, and others, and it has merits like as The level of accuracy grew, while the number of demerits decreased. Makes mistakes from time to time..
3. B. Norgeot, B.S. Glicksberg, and A.J. Butte, 2019. The book's title was A call for deep-learning healthcare, and it was written by name. Deep learning is being used in research in 2019, and it has benefits such as Data collection is secure, but there is a need for more storage..
4. Kumar's book's title is "scalable and secure access control policy for health care system block chain and upgraded bell Lapadula model employing Methodology to research in 2021," and it has benefits such as being more secure and drawbacks such as being difficult to access data.
5. Current status of point-of-care diagnostic devices in the Indian healthcare system with an update on COVID-19 pandemic using Android Application to research in 2020 was written by Konwar, A.N. and Borse, V. The book's title was Current status of point-of-care diagnostic devices in the Indian healthcare system with an update on COVID-19 pandemic using Android Application to research in 2020, and it has merits such as Maintaining the

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Volume 2, Issue 1, June 2022

Impact Factor: 6.252

problem of more memory space and demerits.

## III. SYSTEM MODEL

The system model involves four different entities: cloud server (CSP), users, key pair generator (PKG) and third-party audit agency (TPA),

1. CSP: Cloud servers give consumers access to a large amount of data storage. Users store data on cloud servers, which allows them to conserve local storage space, take advantage of advanced cloud computing features, and share information with others. User: A user is a member of an organization, and a large number of files are stored in the cloud server.
2. PKG: PKG is trusted by other entities. It is responsible for generating system public parameters and signature key pairs.
3. TPA: TPA is a public verifier. It can audit the integrity of the data stored on the cloud server on behalf of the user.

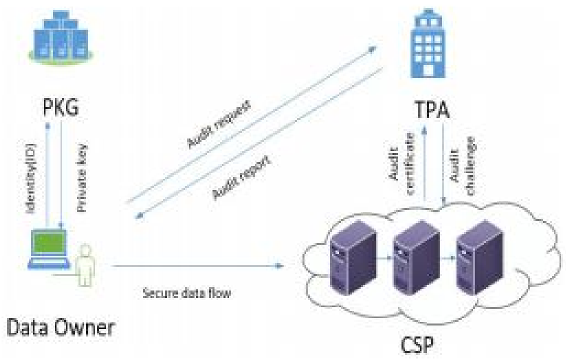

FIGURE 1. The system model of the data integrity auditing.

## IV. SYSTEM COMPOSITION

A multi-copy dynamic data integrity audit protocol includes the following algorithms:

a. KeyGen: This algorithm is executed by the user. It accepts a security parameter x as input, and generates a key pair (sk, pk) and system parameters, which will be used in the following algorithm.
b. ReplicaGen: This algorithm is executed by the user. It accepts a file F as input and outputs Fi = {bi,j} 1≤i≤t,1≤j≤n.
c. TagGen: The algorithm is executed by the user. It takes different copy files and private key sk as input, and outputs aggregate label θj = Qt i=1 σi,j .
d. Store: This algorithm mainly runs on users, CSP and TPA. The TPA verification output result is 1 or 0, and the meaning of 1 or 0 indicates whether you agree to upload data.
e. ChalGen: This algorithm is mainly run by users and TPA to generate a random challenge chal.
f. ProofGen: This algorithm mainly runs on CSP to generate an integrity audit certificate P.

## V. SECURITY MODEL

A secure PDP protocol should be complete, which means that if the prover can generate a valid certificate and pass the verification, the data must be stored. In order to verify the security, we have extended the security model to some copies of the file. We set up a game between the challenger C and the adversary A to show that the adversary A poses a threat to the security of the entire data integrity audit program. In this data model, the data owner is regarded as the challenger C, and the untrusted cloud server is regarded as the adversary A.

This game includes the following stages:

a. Setup phase: The challenger C initializes the system to obtain the public parameters, master key msk and signature key pair.
b. Queries phase: Adversary A and challenger C do three queries next.

2. Hash Queries: The adversary A makes a series of hash queries to the challenger C. The challenger C returns the hash value to the adversary A.
3. Extract Queries: The adversary A queries the private key for the authentication ID. The challenger C runs the Extract algorithm to generate the private key skID and sends it to the adversary A.
4. SigGen Queries: The adversary A queries the signature of file F. The challenger C runs the SigGen algorithm and sends the signature to the adversary A.
5. Challenge phase: In this phase, the adversary A is the audit object, and the challenger C sends a challenge chal = {j, vj}ϒ1≤j≤ϒc to the adversary. At the same time, the adversary A is requested to provide a proof of data possession P based on the challenge chal.
6. Forgery phase: After accepting the request of the challenger C, the adversary A generates a proof of possession P of the data block. If the proof can pass the challenge of the challenger C with a probability that cannot be ignored, it can say that the adversary A has succeeded in the above game.

## VI. DESIGN GOAL:

In order to achieve safe and effective dynamic audit of cloud data, our schema design should achieve the following goals:

1. Public audit: TPA can verify the integrity of the data stored in the cloud server on behalf of the user, and will not cause other problems to the user.
2. Storage accuracy: Only when the CSP can completely save the data stored by the user, the audit certificate chal generated by the CSP can pass the TPA audit successfully.
3. Dynamic operation support: Allow users to perform reasonable dynamic update operations (insert, delete, and modify) the data stored on the CSP, and ensure the efficiency of the entire cloud storage system.
4. Privacy protection: Design a random factor to encrypt the original file, and only calculate the signature of the encrypted file. The CSP and TPA cannot get the valid and correct original file.

## ALGORITHM:

### Dynamic Storage Structure Red-Black Tree

A red-black tree is a self-balancing binary search tree with one extra bit at each node, which is commonly read as the color (red or black). During insertions and deletions, these colors are utilized to keep the tree balanced. The tree's balance isn't ideal, but it's good enough to cut down on searching time and keep it around O(log n), where n is the total number of components in the tree.

$$X(m_{i,j}) = \frac{\sum_{k=1}^{L} A_{i,j,k} \times \frac{1}{256}}{L}$$

$$\begin{cases} K(m_{i,j}) = 1 - aN_L^2 + bN_R \\ T(m_{i,j}) = \dfrac{\sum_{k=1}^{L} A_{i,j,k} \times \frac{1}{256}}{L} \\ X(m_{i,j}) = 1 - aK(m_{i,j})^2 + bT(m_{i,j}) \end{cases}$$

The ASCLL of the k-th character of the j-th block of the i-th copy is represented by Ai,j,k [0, 255](1 I t, 1 j n, 1 k L). A non-leaf node's left and right child nodes are NL and NR. In non-leaf nodes, calculate the value K(mi,j) by NL and NR using the henon map, then calculate the FIGURE 2. The full-node red-black tree T-RB Tree. T (mi,j) using the leaf node calculation method, then use the non mapping to combine the two values.

1. X is the value of a non-leaf node (mi,j). As a result, when a node changes, the change in node value may be enhanced, and the data integrity can be properly appraised. There are usually numerous copies of the data file kept on the cloud server.
2. Checking the binary value of the root node of each red-black tree, especially when the number of copies is huge, will make verification extremely difficult. As a result, the position of the data block of the altered file copy can be calculated by aggregating the root node values of all file copies into the metadata M.

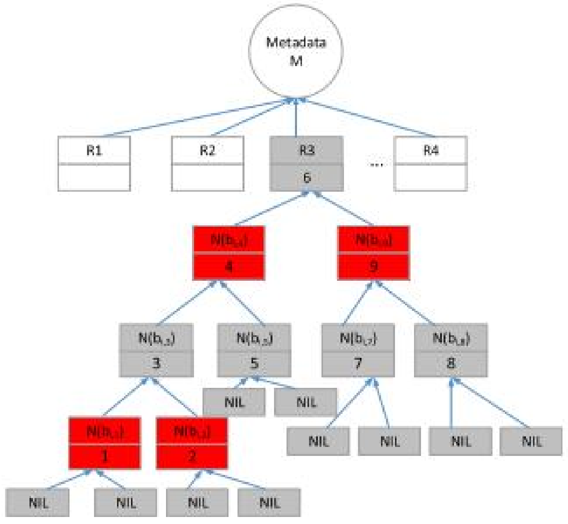

FIGURE 2. The full-node red-black tree T-RBTree.

**Construction of Our Proposal**

The review process is split into two sections in the program the setup phase and the verification phase. The prior phase consisted of some preliminary work, such as KeyGen, ReplicaGen, and TagGen. The user is the major focus of the system settings. The user uses the KeyGen method to establish a public and private key pair, and then uses ReplicaGen and TagGen to conduct further pre-processing operations. Three algorithms are used in the last phase: ChalGen, ProofGen, and Verify Proof. Users, CSPs, and TPAs will all be involved in data verification at this level. The TPA provides verification challenge information to the CSP using the ChalGen algorithm, and the CSP responds using the Proof Gen algorithm to prove the integrity of the stored data. The final Verify Proof algorithm is as follows: TPA audits the proof and sends the audit results to the user by TPA. The following are the specific implementation details:

**1) KeyGen algorithm (1k )**

This algorithm involves the use of a randomly generated number, m, which is used with signing a message along with a private key, k. This number m must be kept privately. The value mis meant to be a nonce, which is a unique value included in many cryptographic protocols.

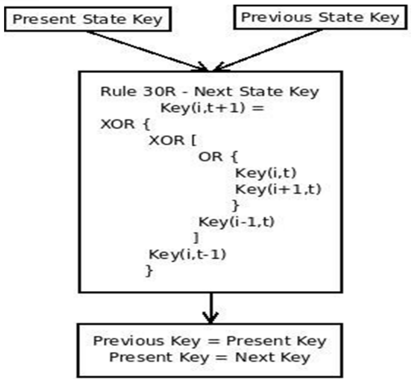

FIGURE 3.KeyGen algorithm Structure

**2) ReplicaGen algorithm (·)**

The dynamic replication algorithm determines when to perform replication, which file should be replicated, and where to place the replica. The main purpose of the dynamic replication algorithm is to increase the data read performance from the perspective of the clients.

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Volume 2, Issue 1, June 2022

Impact Factor: 6.252

1. The user generates multiple copies of the original file by running the ReplicaGen algorithm.
2. The user first selects filename ∈{0,1}∗ for the copies, and divides it into n blocks. Each block has the same size, if the size of the last block is smaller than the other blocks, add 0 to fill it. We select a random number ri,j ∈ Z ∗ p
3. The original file is encrypted, and the encrypted file is F ∗ = {m1, m2, ..., mn} and the data block bi,j = mj + ri,j . If you want to get the original file, you only need to calculate mj = bi,j − ri,j to get the contents of the original file.
4. Finally, t different copy files Fi = {bi,j}1≤i≤t,1≤j≤n can be obtained.

**3) TagGen algorithm (·)**

The owner of the data selects a random element Ui,j , and generates a signature σi,j = sk1 · [H2(U bi,j i,j ) · u bi,j] sk2 ∈ G1 for each random element, where Ui,j = {filename ||ε ||η} is the identification code of data block bi,j , filename is the name of file Fi , ε is the virtual index of data block bi,j , and η is the use of random function f to encrypt filename and ε. The public parameters selected by Ui,j from G1 correspond to the corresponding management cycle. Then an aggregate signature θj = Qt i=1 σi,j is generated. The user initializes the storage into two red-black trees. One red-black tree is used to store the node value and the serial number of the data block, and the other one stores the node value, data block bi,j and the serial number of the data block. The metadata of each copy is calculated by equation 3. Then, the user sends {pk, M, T-RBTree} to the TPA and sends {Fi , θj , C-RBTree} to the CSP. After receiving the data, the CSP needs to verify the consistency between the data block and the signature through the following equation:

e(θj, g) = e(H(ID), mpk) · e( Yt i=1 H2(Ubi,j i,j ) · u bi,j , pk)

**4) ChalGen algorithm (·)**

TPA presents an integrity audit certification challenge to the CSP on behalf of the user using the ChalGen algorithm. On a regular basis, TPA will issue verification challenges to the CSP to ensure that the CSP has all of the copies and that they are complete. The user gives a specific file verification job to the TPA.'

**5) ProofGen algorithm (·)**

To build a relevant audit certificate for a data block, CSP uses the ProofGen algorithm. The CSP locates the storage directory server using the sent command and then transfers j1jc to the relevant storage server after receiving the verification challenge chal. After receiving T (bi,j), i,j1it,1jc, the storage server returns it to the main storage server. The i,j reflects the data block bi verification j's path. The primary storage server will deliver the authentication path i,2 = i,1, i,2, i,3, i,5, i,4, i,9, i,6 of bi,2, where i,1 = T (bi,1), i,4 = T (bi,1)||T (bi,2)||T (bi,3) ||T (bi,5)||T (bi,4), and T (bi,j)1
μ = P γc j=γ1 bi,j ·νj , θ = Q γc j=γ1 θ νj j of the data block bi,j . The CSP returns the audit certificate
P = {μ, θ ,{bi,j, κi,j}1≤i≤t,γ1≤j≤γc } to the TPA.

**6) VerifyProof algorithm (·)**

After receiving the audit certificate P, TPA calculates the node value C(bi,j) based on κi,j , and then calculates M0 based on {C(bi,j), κi,j}1≤i≤t,γ1≤j≤γc . TPA checks M0 = M, if it is not equal, output ''Failure'', otherwise continue to calculate: e(θ , g) = e(H(ID) νj , mpk) · e( Yt i=1 Y γc j=γ1 H2(U bi,j i,j ) νj ·u μ , pk).

## VI. METHODOLOGY

The cloud server (CSP), users, key pair generator (PKG), and third-party audit agency are all part of the system paradigm (TPA),and its processed by using five modules are:

### 6.1 Data Owner

Data owners are either individuals or teams who make decisions such as who has the right to access and edit data and how it's used. Owners may not work with their data every day, but are responsible for overseeing and protecting a data domain.

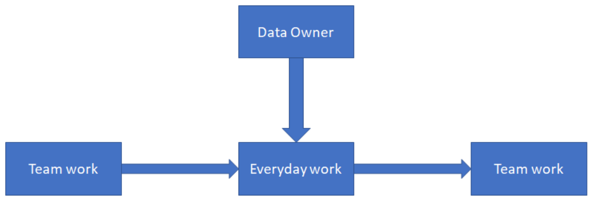

FIGURE 4: process of data owner

**Key Auditing**

Audits provide third-party assurance to various stakeholders that the subject matter is free from material misstatement. The term is most frequently applied to audits of the financial information relating to a legal person. Other commonly audited areas include: secretarial and compliance, internal controls, quality management, project management, water management, and energy conservation. As a result of an audit, stakeholders may evaluate and improve the effectiveness of risk management, control, and governance over the subject matter.

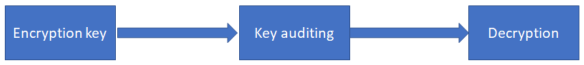

FIGURE 5: process of key auditing

**Third Party Auditing**

In the third party auditor the holistic detection of insecure information flow including implicit and explicit paths is performed and accurate prediction. The lifecycle of cloud applications is also predicted into consideration the inter-component communication between an application's web services.

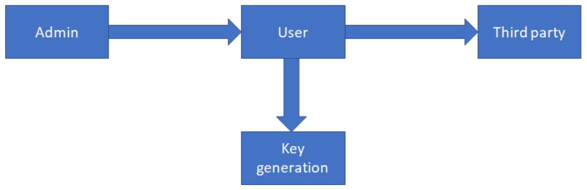

FIGURE 6: process of third party auditing

**Cloud Storage Management**

Cloud data management is a way to manage data across cloud platforms, either with or instead of on-premises storage. The cloud is useful as a data storage tier for disaster recovery, backup and long-term archiving

**Verifications**

Verification in an audit process can be done offsite or onsite. Offsite verification means verification by checking documents, official records, photos and by questioning staff responsible or otherwise trusted to be a reliable source for the facility in verification. Onsite verification means the verifying party is physically visiting the facility, getting introduced into due facts about it on the site where the facility is located and operated.

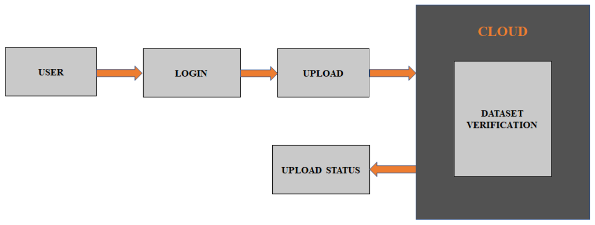

FIGURE 7: process of verification

**Security Analysis**

The proposed scheme is investigated in this part, and the scheme is described by the following theorem.

Theorem 1 (Accuracy): The features of the suggested program are as follows:

(No. 1) (Signing secret key correctness) When the PKG transmits the right signature key to the user, the user can verify the signature key pair.

2) (Correctness of the original file and its associated signatures) CSP can verify the original file and its associated signature.

3) If you're looking for a unique (Authentication correctness) The proof of its generation can pass the TPA verification when the cloud server can completely store the file.

Prove

The user confirms whether the formula (4) is established in the Keygen algorithm by receiving the correct from the PKG. The final derivation of the equation can be proven to be true based on the nature of the bilinear map:

e(ssk, g)

= e(H (ID) x , g)

= e(H (ID), mpk)

The CSP determines the original file Fi and its corresponding tag i,j, which is used to verify the Tag Gen algorithm's formula (5). It can be deduced to achieve the results that prove that it is true based on the nature of bilinear map:

$$\begin{aligned} & e(\theta_j, g) \\ & = e(\prod_{i=1}^{t} \sigma_{i,j}, g) \\ & = e(sk_1, g) \cdot e((H_2(U_{i,j}^{b_{i,j}}) \cdot u^{b_{i,j}}) sk_2, g) \\ & = e(H(ID), g^x) \cdot e(\prod_{i=1}^{t} H_2(U_{i,j}^{b_{i,j}}) \cdot u^{b_{i,j}}, g^{sk_2}) \\ & = e(H(ID), mpk) \cdot e(\prod_{i=1}^{t} H_2(U_{i,j}^{b_{i,j}}) \cdot u^{b_{i,j}}, pk) \end{aligned}$$

After getting a valid audit challenge chal = j, j, CSP was confirmed. The verification equation (6) is derived from the left to the right and then from the right side to the left, depending on the form of the bilinear map, and finally the result is proved correct:

$$\begin{aligned} & e(\theta, g) \\ & = e(\prod_{j=\gamma_1}^{\gamma_c} \theta_j^{v_j}, g) \\ & = e(\prod_{j=\gamma_1}^{\gamma_c} (\prod_{i=1}^{t} \sigma_{i,j})^{v_j}, g) \end{aligned}$$

$$\begin{aligned} & = e(\prod_{j=\gamma_1}^{\gamma_c} (\prod_{i=1}^{t} sk_1 \cdot [H_2(U_{i,j}^{b_{i,j}}) \cdot \mathrm{u}^{b_{i,j}}]^{sk_2})^{v_j}, g) \\ & = e(sk_1^{v_j}, g) \cdot e(\prod_{j=\gamma_1}^{\gamma_c} (\prod_{i=1}^{t} [H_2(U_{i,j}^{b_{i,j}}) \cdot \mathrm{u}^{b_{i,j}}]^{sk_2})^{v_j}, g) \\ & = e(ssk^{v_j}, g) \cdot e(\prod_{j=\gamma_1}^{\gamma_c} (\prod_{i=1}^{t} H_2(U_{i,j}^{b_{i,j}}) \cdot \mathrm{u}^{b_{i,j}})^{v_j}, g^{sk_2}) \\ & = e(H(ID)^{x \cdot v_j}, g) \cdot e(\prod_{j=\gamma_1}^{\gamma_c} \prod_{i=1}^{t} H_2(U_{i,j}^{b_{i,j}})^{v_j} \\ & \quad \cdot \prod_{j=\gamma_1}^{\gamma_c} \prod_{i=1}^{t} \mathrm{u}^{b_{i,j} \cdot v_j}, g^{sk_2}) \\ & = e(H(ID)^{v_j}, \mathrm{mpk}) \cdot e(\prod_{i=1}^{t} \prod_{j=\gamma_1}^{\gamma_c} H(U_{i,j}^{b_{i,j}})^{v_j} \cdot u^{\mu}, pk) \end{aligned}$$

Theorem 2(Audit Security): Assume that the CDH issue in the bilinear group is difficult to solve, and that the file signature generated in the signature method is fundamentally tough.
to hammer If users encounter an issue that is not addressed in Section 4, the strategy outlined in Section 4 will be implemented.
If the CSP is untrustworthy, the user's data may be harmed. omited or even changed At this point, when the CSP is being implemented, It is impossible to calculate a fabricated evidence that can be audited. TPA has confirmed this.
Prove.
To prove the theorem, we'll apply the zero-knowledge theorem. In our suggested approach, a knowledge extractor is built if the CSP can issue a faked audit certificate that can be confirmed by the TPA when the data is inaccurate (such as when the data has been tampered with secretly by the CSP or when the data is destroyed). A complete challenge data block can be obtained by constantly extracting the challenged data block information. A sequence of games will be used to finish our proof.

**Efficiency Analysis**

Our technique is compared to various data integrity verification schemes in this section, such as schemes [20], [23], and [31]. [20] is a dynamic operation method that is based on On MHT, scheme [23] is a multi-copy data integrity verification based on spatiotemporal chaos, while scheme [31] is a multi-copy data integrity verification based on Verification of multi-copy data integrity using an ID signature As a result, contrasting these layouts is a good idea. choose the plan we've proposed.

**Performance Analysis and Comparison**

Assume m is the number of copies, c is the number of data blocks in the audit challenge, n is the total number of data blocks, s is the number of sectors in each block, |F| is the size of the copy file, and n is the total number of data blocks.

**Computational Overhead**

As can be seen, our system has a far lower computing overhead than other schemes. The gap will steadily widen as the number of copies rises. We examined the computational costs of label development, proof generation, and evidence verification at three different stages. These levels use the greatest resources and are the most comparable in the plan.

**Storage Overhead**

The storage cost primarily consists of the cost of storing file copies, the cost of storing associated signatures of file copies, and the cost of storing information, such as verification costs. The storage cost of file copies, as well as the accompanying signature cost of file copies, make for a significant amount of these costs. The Merkle tree is used to store data files in Scheme [20].Each leaf node contains a data block, while each tree keeps a file copy. As a result, in a duplicate, there are as many leaf nodes as there are blocks. The tree's structure will be very large in the case of large data blocks, and its size will be greater than m|F|.

**Communication Overhead**

The communication overhead of the integrity verification scheme generally includes: passing a random challenge chal, responding to a challenge, data integrity certification, and communication overhead in the update phase. For the number c of data blocks users want to challenge, it challenges chal = $\{j, \nu_j\}_{\gamma_1 \le j \le \gamma_c}$ , where j is the log2(c) calculated from the pseudo-random sequence and $\nu_j \in Z*p$ . In the response challenge phase P = $\{\mu, \theta, \{b_{i,j}, \kappa_{i,j}\}_{1\le i\le t, \gamma_1\le j\le\gamma_c}\}$ of our proposed scheme, where $\mu, b_{i,j} \in Z*p$ , $\theta \in G_1$ and $\kappa_{i,j}$ is a cryptographic hash value with a length of log2(n).

## VII. EXPERIMENTAL RESULT

We will use experiments to evaluate our scheme in this part. The majority of the tests are carried out on a Vmware Workstation Pro with 4GB of RAM and 20GB of ROM.
Ubuntu 20.04.1 is the operating system installed on the workstation. Vmware Workstation Pro is installed on a Dell machine with an i5-8300H@2.8GHz processor and 8GB of RAM. The network utilised is the school campus network,

and the code is written in C, with the free Pairing-Based Cryptography (PBC) library [34] and GNU Multiple Precision Arithmetic (GMP) [35] libraries. The basic field size was set to 512 bits in the experiment, and the bit length of the element was set to | p| = 256 bits. The element in G1 has a bit length of |q| = 257 bits.

The computational cost of signature generation will be evaluated. In the experiment, different signatures are generated for different data blocks with an interval of 100 from 0 to 1000, and each block has 5 copies, which are divided into 20 sectors. As shown in Figure 9, the proposed scheme is compared with other schemes, and the signature generation increases with the increase of data blocks, and it grows linearly.

Our scheme also compared with other schemes the integrity audit overhead of CSP for 100 to 1000 challenged data blocks. Figure 10 shows the calculation cost of evidence generation. It can be seen from the figure that our scheme, scheme [20], scheme [23] and scheme [31] have a linear relationship with the number of challenged data blocks in the evidence generation stage, but our scheme overall more efficient.

Then, during the update step, the communication overhead will be assessed. In scheme [20], Barsoum stated that Inserting and removing data are examples of update operations. issues with engineering There are several solutions for various issues. Insertion procedures, such as data insertion and deletion, are technical concerns. Different insertion strategies are employed for different challenges, with the most common balanced binary tree mode being used. Our technique uses the red-black tree data structure, which only requires two rotation operations after the data node is inserted. The schemes [20] and [23] require O(logN) times in the worst case when deleting data nodes, however the red-black tree in our method only requires three times.

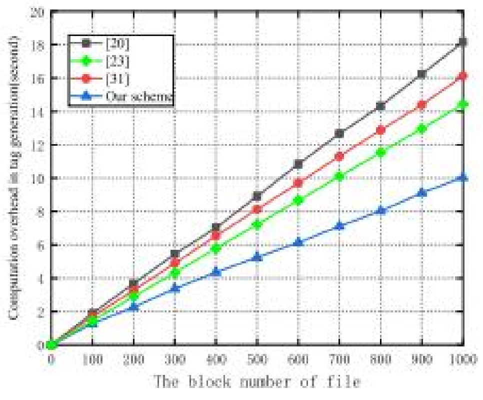

FIGURE 8. The computational overhead of label generation.

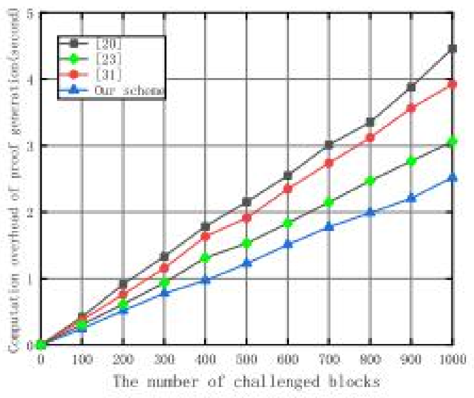

FIGURE 9. The computational overhead of evidence generation.

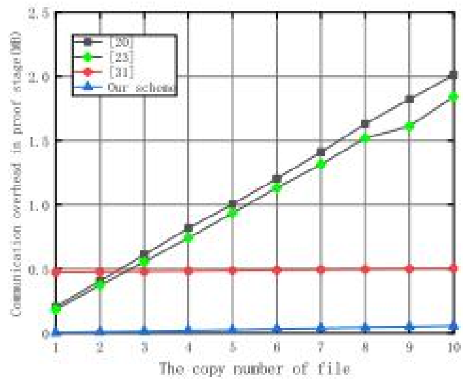

FIGURE 10. The communication overhead of integrity verification.

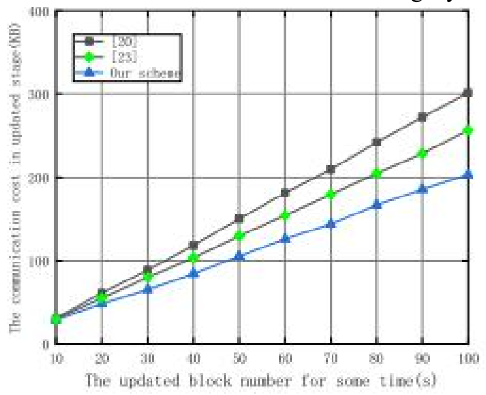

FIGURE 11. The communication overhead of integrity verification.

## VIII. CONCLUSION

This study presents a red-black tree-based multiple copies data integrity verification scheme for cloud storage. It can do dynamic operations on many copies .Boost productivity. The red-black color scheme is ideal for data storage. To efficiently optimize data storage, a tree data structure is used Data storage efficiency and data update activities are simplified. The theoretical study of our approach further demonstrates its security. The results of the experiment also reveal that In comparison to other comparable schemes, our method is superior in Computing, storage, and communication costs are all factors to consider. The results of the experiment show that our hypothesis is correct. system provides the desired level of security and efficiency.

## REFERENCES

**[1].** W. Shen, J. Yu, H. Xia, H. Zhang, X. Lu, and R. Hao, ''Light-weight and privacy-preserving secure cloud auditing scheme for group users via the third party medium,'' J. Netw. Comput. Appl., vol. 82, pp. 56–64, Mar. 2017.

**[2].** M. Alshehri, A. Bhardwaj, M. Kumar, S. Mishra, and J. Gyani, ''Cloud and IoT based smart architecture for desalination water treatment,'' Environ. Res., vol. 195, Apr. 2021, Art. no. 110812, doi: 10.1016/ j.envres.2021.110812.

**[3].** S. Srivastava, S. Saxena, R. Buyya, M. Kumar, A. Shankar, and B. Bhushan, ''CGP: Cluster-based gossip protocol for dynamic resource environment in cloud,'' Simul. Model. Pract. Theory, vol. 108, Apr. 2021, Art. no. 102275, doi: 10.1016/j.simpat.2021.102275.

**[4].** K. He, J. Chen, Q. Yuan, S. Ji, D. He, and R. Du, ''Dynamic group-oriented provable data possession in the cloud,'' IEEE Trans. Dependable Secure Comput., early access, Jul. 2, 2019, doi: 10.1109/TDSC.2019.2925800.

**[5].** M. Kumar, M. Alshehri, R. AlGhamdi, P. Sharma, and V. Deep, ''A DE-ANN inspired skin cancer detection approach using fuzzy C-means clustering,'' Mobile Netw. Appl., vol. 25, no. 4, pp. 1319–1329, Aug. 2020, doi: 10.1007/s11036-020-01550-2.

**[6].** L. Zhou, A. Fu, G. Yang, H. Wang, and Y. Zhang, ''Efficient certificateless multi-copy integrity auditing scheme supporting data dynamics,'' IEEE Trans. Dependable Secure Comput., early access, Aug. 4, 2020, doi: 10. 1109/TDSC.2020.3013927.

**[7].** C. Dhasarathan, M. Kumar, A. K. Srivastava, F. Al-Turjman, A. Shankar, and M. Kumar, ''A bio-inspired privacy-preserving framework for healthcare systems,'' J. Supercomput., early access, Mar. 19, 2021, doi: 10.1007/ s11227-021-03720-9.

**[8].** L. Krithikashree and S. Manisha, ''Audit cloud: Ensuring data integrity for mobile devices in cloud storage,'' IEEE Trans. Depend. Sec. Comput., pp. 1–5, Sep. 2018, doi: 10.1109/ICCCNT.2018.8493963.

**[9].** L. Deng, B. Yang, and X. Wang, ''A lightweight identity-based remote data auditing scheme for cloud storage,'' IEEE Access, vol. 8, pp. 206396–206405, 2020, doi: 10.1109/ACCESS.2020.3037696.

**[10].** S. Peng, F. Zhou, Q. Wang, Z. Xu, and J. Xu, ''Identity-based public multireplica provable data possession,'' IEEE Access, vol. 5, pp. 26990–27001, 2017, doi: 10.1109/ACCESS.2017.2776275.

**[11].** A. Bhardwaj, S. B. H. Shah, A. Shankar, M. Alazab, M. Kumar, and T. R. Gadekallu, ''Penetration testing framework for smart contract blockchain,'' Peer-Peer Netw. Appl., early access, Sep. 5, 2020, doi: 10. 1007/s12083-020-00991-6.

**[12].** P. Shen, C. Li, and Z. Zhang, ''Research on integrity check method of cloud storage multi-copy data based on multi-agent,'' IEEE Transl. Content Mining, vol. 4, no. 8, pp. 17170–17178, 2020, doi: 10.1109/ACCESS. 2020.2966803.

**[13].** Y. Luo, M. Xu, S. Fu, D. Wang, and J. Deng, ''Efficient integrity auditing for shared data in the cloud with secure user revocation,'' in Proc. IEEE Trustcom/BigDataSE/ISPA, Aug. 2015, pp. 434–442, doi: 10.1109/ Trustcom.2015.404.

**[14].** R. Rabaninejad, S. M. Sedaghat, M. Ahmadian Attari, and M. R. Aref, ''An ID-based privacy-preserving integrity verification of shared data over untrusted cloud,'' in Proc. 25th Int. Comput. Conf., Comput. Soc. Iran (CSICC), Jan. 2020, pp. 1–6, doi: 10.1109/CSICC49403.2020.9050098. 75130 VOLUME 9, 2021 Z. Liu et al.: Integrity Auditing for Multi-Copy in Cloud Storage Based on Red-Black Tree

**[15].** C.-T. Hunag, C.-Y. Yang, C.-Y. Weng, Y.-W. Chen, and S.-J. Wang, ''Secure protocol for identity-based provable data possession in cloud storage,'' in Proc. IEEE 4th Int. Conf. Comput. Commun. Syst. (ICCCS), Feb. 2019, pp. 327–331, doi: 10.1109/CCOMS.2019.8821766

**[16].** C. Liu, J. Chen, L. Yang, X. Zhang, and R. Kotagiri, ''Authorized public auditing of dynamic big data storage on cloud with efficient verifiable fine-grained updates,'' IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 9, pp. 2234–2244, Sep. 2014.

**[17].** H. Jin, H. Jiang, and K. Zhou, ''Dynamic and public auditing with fair arbitration for cloud data,'' IEEE Trans. Cloud Comput., vol. 6, no. 3, pp. 680–693, Jul. 2018, doi: 10.1109/TCC.2016.2525998.

**[18].** G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, ''Provable data possession at untrusted stores,'' in Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS), 2007, pp. 598–609.

**[19].** Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, ''Enabling public auditability and data dynamics for storage security in cloud computing,'' IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 5, pp. 847–859, May 2011.

**[20].** A. F. Barsoum and M. A. Hasan, ''Provable multicopy dynamic data possession in cloud computing systems,'' IEEE Trans. Inf. Forensics Security, vol. 10, no. 3, pp. 485–497, Mar. 2015.

**[21].** M. Sookhak, F. R. Yu, and A. Y. Zomaya, ''Auditing big data storage in cloud computing using divide and conquer tables,'' IEEE Trans. Parallel Distrib. Syst., vol. 29, no. 5, pp. 999–1012, May 2018.

**[22].** X. Yang, T. Liu, P. Yang, F. An, M. Yang, and L. Xiao, ''Public auditing scheme for cloud data with user revocation and data dynamics,'' in Proc. IEEE 2nd Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC), Dec. 2017, pp. 813–817, doi: 10.1109/ITNEC.2017.8284847.

**[23].** M. Long, Y. Li, and F. Peng, ''Integrity verification for multiple data copies in cloud storage based on spatiotemporal chaos,'' Int. J. Bifurcation Chaos, vol. 27, no. 4, Apr. 2017, Art. no. 1750054.

**[24].** M. Long, Y. Li, and F. Peng, ''Dynamic provable data possession of multiple copies in cloud storage based on full-node of AVL tree,'' Int. J. Digit. Crime Forensics, vol. 11, no. 1, pp. 126–137, Jan. 2019.

**[25].** R. Curtmola, O. Khan, R. Burns, and G. Ateniese, ''MR-PDP: Multiplereplica provable data possession,'' in Proc. 28th Int. Conf. Distrib. Comput. Syst., Jun. 2008, pp. 411–420.

**[26].** A. Barsoum and M. Hasan, ''On verifying dynamic multiple data copies over cloud servers,'' Dept. Elect. Comput. Eng., Tech. Rep., 2011.

**[27].** H. Shacham and B. Waters, ''Compact proofs of retrievability,'' J. Cryptol., vol. 26, no. 3, pp. 442–483, Jul. 2013.

**[28].** J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, ''An efficient public auditing protocol with novel dynamic structure for cloud data,'' IEEE Trans. Inf. Forensics Security, vol. 12, no. 10, pp. 2402–2415, Oct. 2017.

**[29].** W. Shen, J. Qin, J. Yu, R. Hao, and J. Hu, ''Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage,'' IEEE Trans. Inf. Forensics Security, vol. 14, no. 2, pp. 331–346, Feb. 2019, doi: 10.1109/TIFS.2018.2850312.

**[30].** Y. Zhang, J. Ni, X. Tao, Y. Wang, and Y. Yu, ''Provable multiple replication data possession with full dynamics for secure cloud storage,'' Concurrency Comput., Pract. Exper., vol. 28, no. 4, pp. 1161–1173, Mar. 2016, doi: 10. 1002/cpe.3573.

**[31].** J. Li, H. Yan, and Y. Zhang, ''Efficient identity-based provable multi-copy data possession in multi-cloud storage,'' IEEE Trans. Cloud Comput., early access, Jul. 16, 2019, doi: 10.1109/TCC.2019.2929045.

**[32].** T. Wu, G. Yang, Y. Mu, F. Guo, and R. H. Deng, ''Privacy-preserving proof of storage for the pay-as-you-go business model,'' IEEE Trans. Dependable Secure Comput., vol. 18, no. 2, pp. 563–575, Mar. 2021, doi: 10. 1109/TDSC.2019.2931193. [33] J. Li, J. Li, D. Xie, and

**[33].** Z. Cai, ''Secure auditing and deduplicating data in cloud,'' IEEE Trans. Comput., vol. 65, no. 8, pp. 2386–2396, Aug. 2016.

**[34].** The Pairing-Based Cryptography(PBC) Library. Accessed: 2020. [Online]. Available: https://crpto.stanford.edu/pbc/download.html

**[35].** The GNU Multiple Precision Arithmetic Library (GMP). Accessed: 2020. [Online]. Available: http://gmplib.org