Security and Privacy of Cloud Data Auditing Protocols: A Review, State-of-the-art, Open Issues, and Future Research Directions

Muhammad Farooq[1], Mohd Rushdi Idrus[2]*, Adi Affandi Ahmad[3], Ahmad Hanis Mohd Shabli[4], Osman Ghazali[5]
School of Computing, Universiti Utara Malaysia, Kedah, Malaysia[1, 2, 3, 4, 5]
Institute for Advanced and Smart Digital Opportunities (IASDO), Universiti Utara Malaysia, Kedah, Malaysia[2]

***Abstract*—Cloud service providers offer a trustworthy and resistant-based storage environment for on-demand cloud services to outsource clients' data. Several researchers and business entities currently adopt cloud services to store their data in remote cloud storage servers for cost-saving purposes. Cloud storage offers numerous advantages to users like scalability, low capital expenses, and data available from any place, anytime, regardless of location and device. However, as the users lose physical access and control over data, the storage service raises security and privacy issues, such as confidentiality, integrity, and availability of outsourced data. Data integrity is a primary concern for cloud users to confirm whether data integrity is intact or not. This paper presents a comprehensive review of cloud data auditing schemes and a comparative analysis of the desirable features. Furthermore, it provides advantages and disadvantages of the state-of-the-art techniques and a performance comparison regarding the communicational and computational costs of involved entities. It also highlights desirable features of different techniques, open issues, and future research trends of cloud data auditing protocols.**

***Keywords*—Cloud computing; proof of possession; data integrity auditing; proof of retrievability; public auditing; proof of ownership**

I. Introduction

Cloud computing is a new, rapidly emerging technology paradigm [1, 2], which can resolve large-scale service issues in multiple industries, such as engineering, sciences, and e-commerce [3]. Currently, several Cloud Service Providers (CSPs), for example, Linode [4], Amazon EC2 [5], Google Cloud Platform [6], and Microsoft Azure [7], manage and distribute shared resources for cloud users [8, 9]. Cloud computing offers manageable shared resources like services, networks, servers, applications, and storage that can be accessed over the Internet and managed with minimal effort without the interaction of cloud service providers [10, 11]. Generally, the distributed shared resources of the CSPs are available to their clients based on the pay-as-you-go. The CSPs mainly offer services like Platform-as-a-Service, Software-as-a-Service, and Infrastructure-as-a-Service [12, 13], as shown in Fig. 1. The web applications we used years ago, for instance, YouTube, Facebook, Instagram, and Twitter, are examples of cloud computing services. Furthermore, Dropbox, Amazon web services, and Google applications are generally utilized for personal and business purposes to store and share information anywhere and anytime via the Internet.

The critical part of this business model is to outsource and share data for distributed computing. Cloud storage is an elastic on-demand service model that attains significant benefits; for example, it decreases storage administration burden and reduces infrastructure and software costs. Moreover, it helps to access data from various geographic locations, focusing on ease of maintenance, efficient computation [14], data storage, data archival disaster recovery, etc. [15]. Statista reported that the cloud computing business model persistently develops and is estimated to reach 379 billion US dollars by 2022, showing a tremendous increase in this service sector [16].

Even though the adoption and development of cloud computing businesses are growing every year, most corporations are not satisfied with using this business model due to some obstacles, like pricing, interoperability, vendor lock-in, reliability, and security [17]. Security is the most critical concern [17] because it can be compromised while transferring the data from one location to another by cloud administrators, dishonest CSPs, malware, or other malevolent users who can mutilate the data [18].

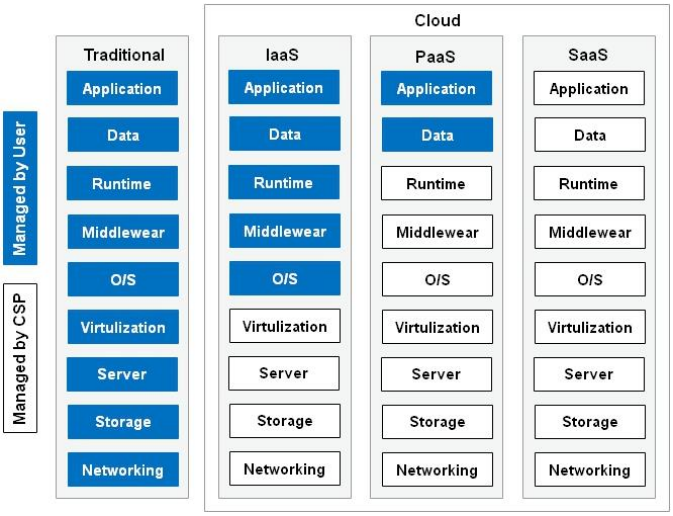

Fig. 1. Service-oriented architecture of cloud computing.

*Corresponding Author.

An insecure storage server can lead to data or privacy leakage if unauthorized users get access to data. For example, the infamous data breach incident of Microsoft's business cloud suit in which unauthorized users gained control of data [19, 20]. Statista shows that the total spending on IaaS is increasing every year [21], and with the increase in cloud demand [21], many security incidents were reported in the top CSPs [22]. The association of information audit systems defines data security as a combination of three fundamental components, i.e., confidentiality, integrity, and availability, also known as the CIA triad [23].

As contribution this paper highlight that cloud computing saves time and monitoring costs for any organization and turns technological solutions for large-scale systems into server-to-service frameworks.

A. Background

Cloud data storage related to IaaS is one of the critical business services offered by CSPs [24, 25]. It provides storage space for users or business organizations to host their personal or business data, as illustrated in Fig. 2. It is an ongoing trend for clients to host their data on remote storage servers. Besides the benefits, the CSPs also incorporate the critical apprehensions for the CIA security fundamentals of their client's data. The main goal is to maintain the integrity of hosted data in the Cloud Storage Servers (CSS) [26, 27].

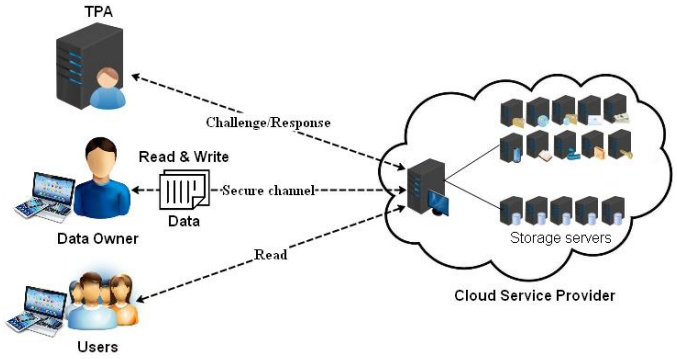

Fig. 2. The architecture of data integrity auditing protocol.

The security of remote data storage is essential because the user loses physical control of their data. One of the solutions to ensure data integrity is to utilize the fundamental cryptographic strategies based on signature and data hashing methods. However, these types of techniques need to store a local copy. Besides, it is unreasonable for the clients to retrieve all the hosted data to confirm its integrity, which incurs high communication overhead over the networks and clients and increases the communication cost. Hence, clients need auditing services for remote data storage to authenticate data integrity periodically.

Currently, researchers are focusing more on verifying the data integrity of the cloud storage servers. However, the cloud server is labelled as an adversary, and the Third-party Auditor (TPA) might see the data contents during the data auditing phase. To overcome these issues, several researchers have proposed different data security techniques. These security techniques can be mainly classified into three categories, namely Proof-of-Data Possession (PDP), Proof-of-Retrievability (PoR), and Proof-of-Ownership (PoW).

1) PDP: These techniques allow the storage server to confirm to its users that the CSPs control the hosted data with probabilistic-based assurance. However, it cannot support the recovery of data exploitation, and the data damage is irrecoverable. This causes serious concerns among cloud users, such as data loss, trust loss, financial damage, etc.

2) PoR: These techniques ensure data integrity and address the limitation of PDP techniques by supporting data exploitation recovery with Error Correction-code (ECC).

3) PoW: In these schemes, the storage servers focus on ownership of data, know which is owned by which user, and prevent downloading remote data by malicious or illegal users.

B. Contributions

In this article, we comprehensively studied several data integrity auditing techniques. The main contributions of this work are mentioned as follows:

1) Briefly discuss system models, basic notations, and security preliminaries used in RDA schemes.

2) Presents a comprehensive review and basic requirements of several cloud data auditing schemes.

3) Comparative analysis with desirable security features of efficient and secure data auditing protocols has been presented.

4) Provides advantages and disadvantages of the state-of-the-art data auditing schemes.

5) Evaluate the performance in terms of communication and computational cost for different entities and the data structures involved in the RDA schemes.

6) Identifies different challenges and future research trends of RDA schemes.

C. Organizations

Section II presents the fundamentals of Remote Data Auditing (RDA) techniques, including system models, notations, and preliminaries used to design RDA protocols. Section III presents state-of-the-art data auditing schemes with basic security requirements. The comparative analysis along desirable security parameters to ensure data integrity and overcome the privacy leakage of RDA protocols is presented in Section IV. Later, Section V presents a comparison and evaluates the performance of different protocols and data structures involved in schemes. Section VI highlights open research issues to design efficient data auditing protocols, and Section VII describes possible future research trends. Finally, we conclude the paper in Section VIII.

II. Fundamentals of RDA Techniques

This section briefly describes the system models, necessary notations, and security preliminaries used for data integrity auditing protocols.

A. System Models

Cloud data storage related to IaaS is one of the critical business services offered by CSPs [24, 25]. It provides storage space for users or business organizations to host their personal or business data, as illustrated in Fig. 2. It is an ongoing trend for clients to host their data on remote storage servers. Besides the benefits, the CSPs also incorporate the critical apprehensions for the CIA security fundamentals of their client's data. The main goal is to maintain the integrity of hosted data in the Cloud Storage Servers (CSS) [26, 27].

The outsourced data integrity auditing services generally include four entities, particularly in public data auditing (see Fig. 3); for instance, (i) Data Owner (DO): is an entity that pre-processed and outsources data to cloud storage servers; it is also capable of performing dynamic data updates through the insert, delete, and update procedures, (ii) the CSP: the entity that offers on-demand shared cloud services and responsible for storing the user data to release the storage burden of its registered users, and it is needed to react the challenges request by the auditor, (iii) Third-party Auditor (TPA): provides audit services, without downloading entire data, with high computational and communicational capabilities to the registered user who delegates his audit task, and (iv) the users: any other user or enterprise who is registered on behalf of the DO and allowed to read the outsourced data.

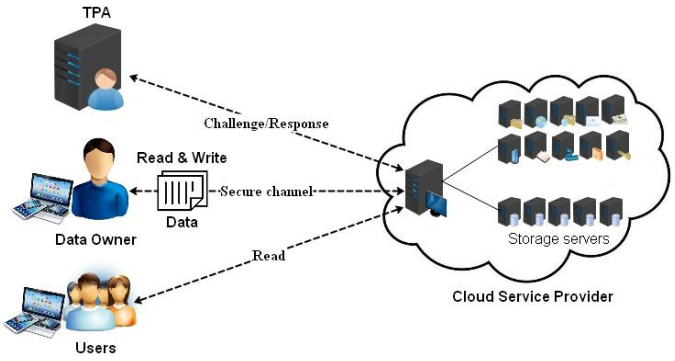

Fig. 3. Public data auditing

On the other hand, in private data auditing, a third-party auditor is not included (see Fig. 4). The outsourced data auditing technique is a challenge-response mechanism, and its comprehensive process is as follows. (i) The users initially pre-compute their data files, utilizing cryptographic, coding, or data block, and afterwards produce some metadata for all data. At that point, they transfer the data files and related metadata to the cloud storage servers. While verifying the data integrity of the hosted data, they send a challenge request to the auditor and wait for the notice of the outcome. (ii) The auditor produces an arbitrary challenge after getting the user's request and sending it to the storage servers. We regularly expect that the user approves the auditor before any activity with cloud storage servers. (iii) after getting the challenge request, the storage server produces corresponding evidence identified with the challenge and responds to the auditor. (iv) to verify the data integrity, the auditor confirms the receiving evidence. If the authentication fails, the auditor responds with a "rejected" notification to the user. Otherwise, the auditor responds "success" notice to the user, which means the outsourced data is secure.

B. Notations and Preliminaries

Homomorphic Verifiable Tag (HVT) is the foundation of current data integrity verification techniques. It can aggregate different data blocks into one value and save substantial communication costs. For enhancing audit phase proficiency, the signature mechanism is integrated with HVT to create a homomorphic signature method to verify the integrity of big data outsourced in existing data auditing techniques. As per the review of current RDA techniques, most of the data storage auditing protocols are constructed using Message Authentication Code (MAC) [28], RSA, BLS, Homomorphic Linear-authenticator (HLA) [18, 29], Identity (ID), and Certificate-less (CL) homomorphic methods. Cryptographic systems utilize these algorithms to build security primitives and cryptographic groups. Cryptographic operations are the security primitives to create the audit process, including key, tag, challenge, and proof generation phases. Nevertheless, the TPA can learn the user's data during the audit phase, which puts the data at risk [18]. To protect the privacy and integrity of data, user-initiated key, tag, or signature generation algorithms. In contrast, the cloud server and TPA execute the challenged and proof generation algorithms using public and private key security parameters.

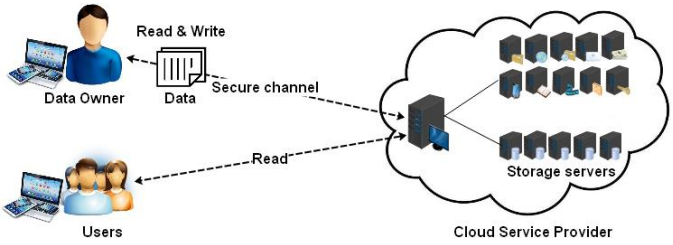

Fig. 4. Private data auditing.

A commonly utilized security parameter in a cryptosystem, denoted as n, describes the length of public and secret keys. The security parameters must be computed viably, and the implementation of the cryptographic system must be polynomial in time. When the user increases the key size, the encryption and decryption time will also be increased. It is harder for an adversary to break the system in polynomial time, generally addressed as 1n. The cryptographic procedures and keys generation for the audit reliability of the cloud storage services, and 80, 128, or 160 bits are the commonly used problem's input size security parameters [30]. The remotely stored file F is addressed as an arrangement of finite sets of memory blocks like m1, m2, m3…mn. It is essential to verify that the security parameter is not lesser than the data blocks because the user needs to encrypt the data with a relevant key [31].

The pairing function is another cryptographic method for security systems, such as let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups belonging to similar prime order $\mathbb{q}$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are additive and multiplicative groups, respectively. A bilinear map $e: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{GT}$ has the following properties:

- Non-degeneracy: suppose $\mathbb{P}$ is a creator of $\mathbb{G}_1$, so $e(\mathcal{P},\mathcal{P})$ is a creator of $\mathbb{G}_2$. Therefore, $e(\mathcal{P},\mathcal{P}) \neq 1$; and e proficiently computable;
- Bilinearity: $e(a\mathcal{P}, \mathcal{b}\mathcal{Q}) = e(\mathcal{P},\mathcal{Q})^{a\mathcal{b}}$, for all $\mathcal{P} \in \mathbb{G}_1$, $\mathcal{Q} \in \mathbb{G}_2$ and $a, \mathcal{b} \in \mathbb{Z}_{\mathbb{q}}$, where $\mathbb{Z}_{\mathbb{q}}$ is a prime order.
- Computability: The map 'e' is efficiently computable.

1) Message Authentication Codes (MAC): The MAC ensures non-repudiation of the origin, validates the owner's identity, and preserves data integrity. The hash functions need to generate the MAC codes for validation, including hash value and data block. The receiver utilizes the shared secret key known by both parties to decode and get the original message. It is easy to use MAC in the data audit phase; the user generates data blocks and their respective metadata (MACs) of data file F, uploads it to storage servers and sends the security keys to TPA. Nonetheless, this methodology needs to release the data blocks to the auditor [18]. So, cloud users can perform the integrity verification process (private audit) to prevent the TPA from validating their data. Even though the MAC authenticates the data integrity, data privacy is lost. Moreover, data integrity is at risk because attackers can modify or share the message with other users. The subsequent algorithms can be utilized in the cloud data audit phase [18]:

- generateKey($\mathcal{K}$): $\mathcal{K} \xleftarrow{s} \mathcal{K}$
- generateCode: $tag \xleftarrow{s} MAC_K(M)$
- verifier: $\mathcal{D} \leftarrow VF_K(M, tag)$ where $\mathcal{D} \in \{0,1\}$

The MAC uses a deterministic and state-less key generation algorithm, so it is not required a tag authentication algorithm because the receiver can create a tag by executing $MAC_K(M)$. The receiver can verify the message if the generated tag is matched to the received tag; otherwise, it fails to authenticate. Instead, using the MAC in data auditing can cause significant security issues, including the following [32]:

- It provides static data only and is unable to support dynamic data operations.
- If the user needs to update data, it requires regenerating the security keys and uploaded to the storage servers and auditor.
- The auditor should keep MAC keys because cloud users can modify their data from any geographical location.

2) Homomorphic Authentication (HA): The study in [30] was the first to introduce homomorphic linear authenticators. The HA allows the user to generate the data tag α by using the data blocks $\{M_i\}$ along with secrete keys, where $i = \{1,2,3 \ldots n\}$ and store it on the cloud servers. After that, the CSP generates the data blocks $\{M_i\}$ with relative data tag by using publically accessible functions. So, the HA permits anyone to validate the output derived from the verified datasets using complex computational procedures with data tag α. Moreover, this approach also allows users to combine data stream bits of several files without revealing data contents to others.

Consider the case of supply chain management, the transactions of each department carried out for production, sales, and retail, without revealing information to other departments. The HA can be categorized as (1) partially homomorphic cryptography, which can be additive or multiplicative, and (2) fully homomorphic encryption that supports both additive and multiplicative operations [33]. The procedure to perform the storage authentication can be briefly described as follows:

- The data file φ is represented as an N vector.
- Then generated the tags τ of every data block φ.
- The user randomly generates a challenge request c and sends it to the cloud storage servers.
- The server responds as data verification proof by using:

$$\mu = \sum_j c_j, \varphi_j$$

The homomorphic provable tag has been utilized in data audit procedures. They have flexibility and Blockless authentication properties. Blockless authentication permits the cloud servers to authenticate the integrity of data deprived of computing the data and metadata of the data block. Metadata (tags) and the distinctive index are created and stored as inclusive counters for every data block. Later, the storage servers generate a proof and allow users to authenticate data integrity with linear summation of tag values [34]. The HA provides cumulative signatures, where ɳ signatures related to ɳ messages for ɳ number of users [35]; support homomorphic signature [36] and batch authentication [37].

The following four algorithms are included in the homomorphic authentication [30]:

$(\rho\kappa, \mathcal{s}\kappa) \leftarrow (1^k)$: The data owner initiates this algorithm for the setup phase to verify data storage. It requires security parameters to compute public $\rho\kappa$ and secret $\mathcal{s}\kappa$ keys.

$(\vec{t}, st) \leftarrow Tag_{sk}(\vec{f})$: It is also a probabilistic algorithm executed by the user to tag a file. It uses a private key $\mathcal{s}\kappa$ and a file $\vec{f} \in [\mathbb{B}]^n$ as an input security parameter and computes the tags' vector and state information st.

$\tau := Auth_{pk}(\vec{f}, \vec{t}, \vec{c})$: The cloud storage servers executed this deterministic method by computing a tag. It uses public-key $\rho\kappa$, a data block $\vec{f} \in [\mathbb{B}]^n$, a tag $\vec{t}$, and a challenge request $\vec{c} \in \mathbb{Z}_p^n$ as an input security parameter and computes a proof τ.

$b := Verify_{\rho\kappa}(st, \mu, \vec{c}, \tau)$: It is also a deterministic algorithm that is run by the verifier; it uses security parameters public-key $\rho\kappa$, state info st, an element $\mu \in \mathbb{N}$, challenge request $\vec{c} \in \mathbb{Z}_p^n$, a proof τ as an input, and generates a bit '1' or '0', accept or reject, respectively.

We determine that the private key is not required during authentication in the algorithms mentioned above. Moreover, linear data block combinations expose sufficient information to

the auditor for downloading the complete file f [32, 37]. The state info belongs to $\{0,1\}^k$ which is just a security parameter resulting from tag or encode a file f algorithm. The researchers Ateniese et al. treated data file f to be n vectors. Every tagged file f could be recognized by using the state information computed in encoding algorithm [30]. The HA technique can enhance using random-masking [38] and ring-based signatures [18] schemes, particularly for public data auditing. The homomorphic authenticator-based ring signatures strategies use to share data among multiple cloud users and propose supporting data privacy and Blockless authentication. Furthermore, random masking provides privacy-preserving data auditing process.

3) RSA-based homomorphic methods: The researchers in [34] introduced a sample-based publicly audit data possession technique that integrates the RSA approach with the HVT method. The succeeding RSA-based methods mostly improved their scheme. In their scheme, some essential elements are produced similar to the RSA signature, where the public and private keys are generated as $\{N, g\}$ and $\{e, d, v\}$. The client split the data file F into n number of data blocks $F = \{x_1, x_2, \dots, x_n\}$, where $x_i \in \mathbb{Z}_q^*$. For every data block, the client creates a block tag $\sigma_i = (h.u^{x_i})(y_i)^d modulo\ N$, wherein $h: \{0,1\}^* \to R_N$ is a hash method that homogeneously maps to R_N, u is a creator of R_N, x_i denotes the i-th data block, and the value of y_i is computed by concatenating the index i with the secret value.

The value of y_i is distinct and unidentifiable for every tag. Then, the cloud storage servers store data files and their corresponding tags. Afterward, the user can confirm that the cloud server holds the data by creating a requested message for arbitrarily chosen data blocks. The storage server creates the proofs of data possession based on requested data blocks and related tags. Consequently, the user could ensure data integrity without retrieving entire data blocks. The replica storage, Curtmola et al. [39] also created a tag for data blocks similarly $\sigma_{ij} = (h.u^{x_{ij}})\left(y_{ij}\right)^d modulo\ N$, wherein x_{ij} denotes the j-th data block of i-th copy. Currently several solutions that adapted RSA method just change the hash function, such as $h(filename)$ [40] and $h(filename \parallel n \parallel i \parallel j \parallel g)$ [41].

4) BLS-based homomorphic method: This method generates a shorter signature than the RSA-based homomorphic method at the same security level; thus, most existing data auditing schemes use the BLS-based technique. The first publicly data integrity auditing technique with data dynamics features based on this method was proposed in [42]. The bilinear pairing utilized to construct the BLS technique for confirmation and signature is computed by elliptic-curve cryptography. It is an undisputable technique that verifies parties that the signature is reliable.

Moreover, the BLS could integrate with any other approach in group Diffie-Hellman assumption (GDH) $\mathbb{G}$ [43]. This procedure needs a hash-based method resulting in data space on $\mathbb{G}$. This technique could also be used in data audit schemes. Suppose $\mathbb{G} = \langle g \rangle$ prime number $\mathcal{P}$ set of the GDH, by using hash method $\mathbb{H}: \{0,1\}^* \longrightarrow \mathbb{G}$ be measured in the random oracle model. Each data block can encrypt by utilizing subsequent algorithms [43]:

- Key creation algorithm run by the cloud user by selecting an arbitrary variable, $x \xleftarrow{R} \mathbb{Z}_{\mathbb{p}}$ and generates $v \leftarrow g^x$. The public $\rho\kappa$ and secrete $\mathcal{s}\kappa$ keys are $v \in \mathbb{G}$ and $x \in \mathbb{Z}_{\mathbb{p}}$ correspondingly.
- The tag generation algorithm $\sigma_i = (h.u^{x_i})(m_i)^x$ for each data block. The exclusion of index *i* supports the technique to provide data dynamics features.
- The signing algorithm utilizes a secret key and message $\mathbb{M} \in \{0,1\}^*$ computed has with $h \leftarrow H(M)$, wherein $h \in \mathbb{G}$ and $\sigma \leftarrow h^x$.
- Verify function generates $h \leftarrow H(M)$ by using a signature σ, public key $\rho\kappa$, and a data block. Therefore $(\mathcal{g}, v, h, \sigma)$ is confirmed as a legal tuple.

These techniques can further enhance supporting public data auditing and data update operations. The Merkle-binary hash-tree (MHT) is a technique to meet these objectives. Qian Wang et al. described that the MHT-based scheme hashes leave to authenticate data blocks [42]. Furthermore, to support dynamic data audits, their work extends the techniques [34, 44] to measure signatures with relative file indexes. Consequently, the previously stored data file needs to be re-computed in the data file modification process. Hence to minimize the file indexes storage overhead, the study [42] discarded the file indexes and generated tags for every data block to support dynamic data operations.

Several BLS-base techniques only change hash functions such as $h(w_i)$ [45, 46], $h(id)$ [47], $h(i)$ [48], and $h(V_i \parallel T_i)$ [49]. To provide the optimum solution, the study [31] presented a common protocol development mechanism for cloud data auditing. They divide the data file F into n number of data blocks and divide these data blocks into sectors s. Then data owner computes the tag of every block of data as $\sigma_i = \left(h.\prod_{j=1}^{s} v^{m_i j}\right)(m, fid \parallel i)^x$, wherein fid denotes identifier of the data block, and m_{ij} identifies the j-th sectors of relative i-th data blocks and $\{v_1, v_2, \dots, v_s\}$ are arbitrarily selected by the client. A similar technique $\sigma_i = \left(h.\prod_{j=1}^{s} v^{m_i j}\right)(m_i)^x$ presented by Liu et al. [50]. Later this technique was enhanced by [51] for the multi-replicas file system that also supports data update operations by generating $\sigma_i = (h.v^{m_{ij}})(m_i)^x$ block tags for cloud storage, wherein m_{ij} denotes the j-th data blocks of i-th copy. The use of partition mechanism and distinctive hash function design led to developing the different BLS-based techniques for outsourcing data auditing such as for data replication $\sigma_{ij} = (h.v^{m_{ij}})(filename \parallel n \parallel m \parallel v)^x$ [52] and $\sigma_{ij} = \left(h\left(m_{ij}\right).\prod_{j=1}^{s} v^{m_i j}\right)^x$ [53].

5) ID-based homomorphic methods: This technique does not support certificate features compared with RSA-based and BLS-based signatures. Wang et al. [54] used ID-based cryptography to design a PDP solution; they adapted identity aggregation signatures to develop a proof of data possession technique and demonstrated the system and security models.

The Private Key Generator (PKG) chooses the secrete key as $x \in \mathbb{Z}_p$ and generates the public key as $y = g^x \in \mathbb{G}$ in initialization process. After getting user ID, PKG generates $R = g^r$ and $\sigma = r + xh(ID, R)$ and responds to the user with the private key (R, σ), wherein $r \in \mathbb{Z}_p$ is arbitrarily chosen integer. The user computes the tag $\sigma_i = (h.v^{m_i})(i)^{\sigma}$ for a data block m_i.

The signature construction procedure is similar to BLS-based solution [48]; the only change is that the secrete key is computed by PKG. Hence, study in [55] suggested another ID signature-based PDP public data auditing technique for multi-clouds. In this method, secrete key creation is similar to the study mentioned above [54]; however, the signature structure is different because of the partitioning mechanism and cooperation among different servers. The client generates a tag as $\sigma_{ij} = \left(h.\prod_{j=1}^{s} v^{m_i j}\right)\left(N_i, GS_{l_i}, i\right)^{\sigma}$ for tuple $\left(N_i, GS_{l_i}, i\right)$ can verify the every data blocks different from each other. Nonetheless, in this technique the data security is not sufficient. It does not provide requested blocks for tagGen queries; in other words, the attacker cannot retrieve any tags of these data blocks, which is conflicting with the real capacity of the cloud storage server.

The researchers in [56] proposed an ID-based publicly provable privacy-preserving data auditing solution. Using asymmetric group key agreement suggests a substantial ID-based data auditing technique. The PKG produces the secrete key as $\sigma = (ID)^x$ and send it to the user. Then the user generates the tag $\sigma_i = \sigma^{m_i}.h(filename \parallel i)^{\eta}$ for a data block m_i, wherein $\eta \in \mathbb{Z}_p$ is arbitrarily selected by the user. The serial number integrated with the tag makes data dynamics operations impossible inconceivable.

6) Certificateless-based homomorphic methods: The Boneh–Lynn–Shacham-based technique consistently needs a reliable party to compute the user's private key; hence, data signatures can easily tamper once the party is compromised. The certificate-less cryptography settles this issue. The authors in [57] suggested the first certificate-less publicly provable data auditing scheme to validate the outsourced integrity of cloud storage. A homomorphic verifiable CL signature is constructed for the user/auditor to validate the data integrity without retrieving the entire data file, which is impossible in a conventional CL signature.

In the initial phase, the Key Generation Center (KGC) chooses to secrete the key as $X \in \mathbb{Z}_p$ and generates the public key $y = g^x \in \mathbb{G}$. The PKG receives the user's ID and generates partial-key $D = h(ID)^x$ send it to the user. Then the user chooses another partial key $X \in \mathbb{Z}_p$ to make the secrete key complete and generates public-key $y = g^x \in \mathbb{G}$. The user generates $\sigma_{ij} = (h_2.v^{m_i})(ID \parallel y \parallel id)^x.D$ tag with complete private key for a data block m_i with identifier id. Even though this method provides the solution for the private-key escrow problem with the help of KGC to generate a partial key instead of the complete key, it fails to avoid the public-key replacement attack. He et al. [58] proposed another CL technique for outsourced data integrity verification. The KGC produces the partial key in a generally unpredictable manner as follows:

- KGC sends arbitrary integer $\eta \in \mathbb{Z}_p$ and generates $T = \eta.p,\ S = \eta + X.h(ID, T)\ modolus\ P$, where P denotes the multiplicative group $\mathbb{G}$.
- The KGC responds with computed partial-key (T, S), after that, the user creates a tag for the data block m_i along with id by generating $\sigma_i = (S + h_2(ID \parallel y \parallel Y).x).\left(h(id) + m_i.h(Y)\right)$ which is secure against public-key replacement and master-key retrieval attack.

The auditor can get the user's data inappropriately by solving the linear equation; this method does not ensure data privacy. Consequently, the study in [59] was introduced the CL privacy-preserving PDP technique by generating the public key with two private-key parts instead of using one part [57, 58]. The security evaluation shows that the proposed technique is verifiable and secure.

III. State-of-the-Art Data Integrity Auditng Techniques

This section presents a detailed comparative analysis of remote data integrity auditing protocols. Several security mechanisms have been projected for ensuring data integrity and overcoming privacy leakage in the literature. These schemes can be categorized according to data states: static, dynamic, privacy-preserving, single-cloud, multi-cloud, multi-owner, etc.

In study [34], the researchers proposed the first PDP protocol to address the issues of public data auditing for outsourcing the user's data on remote cloud storage servers and the authentication of data possession. They have presented two schemes: (1) the sampling-PDP method ensures strong authentication of data possession, and (2) the efficient-PDP approach supports improving proficiency with weak data possession. They used RAS-based homomorphic tags to achieve a public data auditing scheme. However, their methods only support static data, so direct data modification might raise significant security, privacy, and system design issues. The authors [60] improved their scheme and presented a dynamic version of the PDP protocol. However, their technique restricted unlimited challenge queries and supported only limited dynamic data operations, such as unable to provide block insertion.

Bowers et al. [61] introduced the HAIL protocol to provide high availability and ensure data integrity. Their protocol used the erasure codes on both single-server and multi-server layers correspondingly. It also supports the assurance for proof of data retrievability that is outsourced to remote cloud storage. Nevertheless, their protocol does not offer dynamic data and needs to store one segment of each data file locally. Moreover, it is restricted to the number of challenge queries. The authors [62] introduced the SW approach to improving the private data auditing technique presented by Shacham and Waters [63]. The server response size concerning the challenge request of the TPA is directed by the 't' set of elements along with the size of each group element in α bits, where α represented the group size in bit. They considered and improve the size of proof from

$O(\mathcal{S}\alpha)$ to $O(\alpha)$ in their scheme by splitting the 't' group components into two group components. It generates a longer public key compared to the technique designed by [34]. Furthermore, their approach is limited to the static nature of data and cannot ensure privacy leakage against TPA during the data auditing phase.

Stefanov et al. [64] considered the static data issue and introduced the first cloud-based PoR protocol along with a dynamic data operation named Iris. They used the MHT data structure to store and ensure outsourced data integrity. However, regardless of proficient read and write data procedure, it requires substantial bandwidth usage, server computation cost, local data space, and server input/output to verify data integrity. Like the protocol [64], Shi et al. [65] have introduced a standard MHT data structure-based publically provable technique.

Qian Wang et al. [42] improved the proof of the data possession scheme for dynamic data operations by using the MHT data structure to authenticate the block tag. They provide the solution for data dynamics and Blockless authentication. The previous studies [34, 44] describe that if the cloud server keeps a tempered data file, then the cloud server can identify this misconduct during the data authentication phase by using a data auditing algorithm with a probability of $O(1)$. Moreover, this scheme cannot detect and verify minor data modifications [42]. Similarly, in [30], the researchers introduced an extended version of the data auditing protocol using ranked-based MHT, as illustrated in Fig. 5.

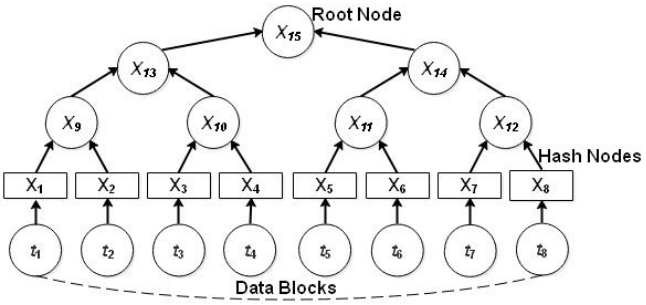

Fig. 5. The basic idea behind the MHT-based solution.

Their protocol used a signature-based approach [66] and supported authorized data auditing to avoid malicious third-party auditors. In their scheme, every node $\mathcal{N}$ should not be more than two child nodes. Every node can be denoted as $\{\mathcal{H}, r\mathcal{N}\}$, wherein $r\mathcal{N}$ representing the rank of the current node and $\mathcal{H}$ indicated hash value. Data blocks or messages m_i belongs to leaf nodes $\mathcal{LN}$, it computes hash value by $\mathcal{H}(m_i), r\mathcal{N}$, and generates the tag σ using following equation:

$$\sigma = \left(\mathcal{H}(m_a) \prod_{b=1}^{s_a} u_b m_{ab} \right)$$

where, $u_b \in \mathcal{U}, \mathcal{U} = \{\mathcal{U}_k \in \mathbb{Z}_{\mathbb{p}}\}, \mathcal{K} \in [1, s_{max}]$ the sector of data blocks s, the data file f is divided as $\{m_{ab}\}$, wherein m is the message or data block, a is the length of the current block, and b is the set of s sectors. One of the significant properties of this protocol is that an unauthorized user cannot generate a challenge for TPA without having this verification tag. Several proposed protocols cannot ensure data privacy against third-party auditors [32, 34, 42, 44]. The researchers already reported that the auditor could learn the data content by data block retrieval in the proof of possession step during the data audit procedure [67].

Several studies [68-71] introduced privacy protection protocols to determine the privacy leakage issues. In [69], the researchers designed a zero-knowledge proof-based privacy-preserving scheme with publicly provable data. Their method saves the computational and communication overhead compared to [72] and ensures data integrity without exposing the content of the user's data to the third-party auditor. However, it does not offer dynamic data operations and is limited to static data only. The above literature shows that the currently proposed data auditing schemes use the public key cryptography approach. So, the cloud needs to identify the user before hosting the data on remote storage servers for spam prevention.

Wang et al. [73] addressed this problem and suggested an ID-based outsourced data verification technique (ID-RDPC) to overcome this problem. It is the first scheme that considers being secured under Diffie-Hellman supposition. Their strategy is more optimized for communication overhead by comparing with [74]. However, their scheme cannot support public audit, generating an extra burden of computational and communication costs at the user-side during the data verification step. Thus, the ID-RDPC scheme is not suitable for resource-constrained devices.

Similarly, Zhang et al. [75] presented another user's identity-based remote data verification protocol using homomorphic tags. In their technique, the computational overhead remains consistent for the third-party auditor for the number of challenge queries. Moreover, their scheme is more proficient for computational and communicational overheads than [47, 76].

Jiang et al. [77] introduced a publically verifiable data integrity auditing technique with client revocation. They utilized a public-key cryptography database and aggregate signature to overcome collusion issues between the cloud and revoked users. It allows users to host data to the storage server and generates authentication codes to verify users from the revocation list. They also enhanced their technique with a batch auditing mechanism, which is challenging to implement in public auditing schemes. Besides, it incorporates other security features, for example, confidentiality, accountability, and traceability, to secure group user updates. The third-party auditor cannot authenticate the impersonation attack if the user does not exist in the revoked list.

Similarly, Fu et al. [78] were motivated to design a new external auditing approach to share data with group managers in CSP. They used standard MHT to maintain verifiable data blocks. This technique guarantees the users to trace the changes through an assigned hash-based binary tree and recover the most recent block of data when any existing data block corrupts. However, it is susceptible to numerous assaults, such as tag forgery, replace, replay, pollution, and data leakage

attacks. Later, an identity-based cloud storage technique for the clients to remotely store their data was securely presented by Wang et al. [79]. Their approach does not need to manage certificates and enables inclusive data auditing. They also permit the proxy server to process and host the data file on the user's behalf for efficiency. However, their technique does not support data privacy and recovery, treating all system parties as trusted entities.

A scheme to secure outsourced data in the cloud uses the re-computing codes suggested by Liu et al. [80]. A semi-authorized proxy server is accessible on the user's behalf. The proxy server keeps the compensation of the hashed data blocks and homomorphic verifiers dependent on BLS-based signatures. Furthermore, the proxy-server resolves unapproved validator issues generated by specific keys in the absence of a user who is not consistently accessible online. Consequently, this technique could nearly release the burden of data owners to become available online permanently. They used the coefficients encoding and a pseudorandom method to accomplish the privacy of the user's data. Nonetheless, their strategy is unable to support external data auditing.

Yuan and Yu [81] presented a new public data auditing PoR mechanism based on homomorphic linear authentication commitment with constant communication costs for the cloud and TPA. It provides proficiency for storage, communication, and computational costs and releases the user's burden from being always online for data auditing. Though, it cannot support dynamic data update procedures, batch auditing, preserving data privacy, and Blockless verification.

Later, Fan et al. [82] introduced an ID-based data verification technique using aggregate signatures named (SIBAS) to overcome the vulnerability of user's data to an adversary CSP. They introduce a trusted auditor (TEE) to verify the remotely stored data on the local side. Moreover, the scheme also accomplishes the management of security keys in the TEE environment by using Shamir's (t, n) mechanism. It used group Diffie-Hellman supposition under the random oracle model (ROM) to resist the adversary attacks that might select messages and target identities for security verification. It is also optimized for computation and communication costs compared to [38, 76]. However, it does not support privacy-preserving, Blockless validation, and dynamic data operation.

In study [83], the researchers presented a secure data deduplication and integrity verification protocol, which can decrease the data volume hosted on cloud storage by removing identical copies of the data file. It also allows users to delegate the computational procedures to the trusted third-party auditor to authenticate data integrity proficiently. Several research works have been directed toward these issues, while this study designs a new scheme by combining the features like deduplication and publicly provable data integrity. This study supports the third-party auditor in releasing the user-side burden, particularly for resource-constrained mobile devices. Moreover, it used the linear-homomorphic authenticators and BLS signature to perform challenge-response mechanisms. However, it generates high computational and communication costs in the data deduplication step on the user side. It does not provide data update operations and batch auditing.

A multi-agent and multi-copy-based data integrity authentication technique for big data files in single cloud storage servers with low audit efficiency was introduced by Chunbo Wang and Xiaoqiang Di [84]. It utilizes a bilinear mapping technique to build a key creation procedure and multi-branch confirmation tree for performing multi-copy data signature to deploy multi-copy validation, signature, and confirmation. Moreover, it addresses task association in the work process, task assignment, and asset allotment dependent on QoS request inclination settings to plan various tasks using a directed acyclic graph (DAG). Moreover, it reduces the communication cost and storage overhead and improves audit proficiency by 20% compared to [63, 85]. However, it cannot perform dynamic data updates, batch auditing to increase efficiency, and privacy-preserving against TPA.

Yang et al. [86] introduced a certificate-less signature-based scheme for multi-user privacy-preserving with traceability and confirmation for cloud data auditing. They addressed denial-of-service (DoS) attacks, single-supervisor misconduct, and identity revelation problems. In contrast to the conventional data integrity techniques, it preserves the secrecy of the user's identity without using a group and ring signature, which ensures the tag is minimal. Besides, it supports collaborative traceability of malevolent users with at least d managers, evading single-supervisor power maltreatment. It overcomes the DoS attack between TPA and cloud server providers by using identity verification measures. Any user can send a challenge to CSP to resolve the network congestion and waste of cloud resources. It also supports proficient user revocation and releases the burden of certificate management and key-escrow problems using certificate-less cryptography. However, TPA cannot prevent impersonation attacks if the user does not exist in the revocation list, imposing high communication and computation costs due to static data.

Later, a study [87] presented an efficient PDP scheme under the Diffie-Hellman assumption to verify data integrity in storage servers by preserving users' anonymity against TPA. Therefore, the auditor cannot get the user's identity in the data audit process. It avoids certificate management by using an identity-based cryptographic approach. It ensures the connection between data and the owner in the proof creation step, not the integrity audit phase. Hence, TPA is unable to know liaison to find challenged data usage. Simultaneously, the CSP creates a relation for the proofs in the initialize phase to diminish TPA's computational overhead significantly. Moreover, use arbitrary requested data blocks in the proofs step to strengthen the security of the technique. Though, it is unable to support dynamic data updates and batch auditing.

Neela et al. [88] introduced a technique by improving Rivest-Shamir-Adleman (RSA) algorithm with Cuckoo Filter for secure cloud storage in semi-trusted CSPs. They eliminate the third-party auditor to overcome privacy issues and reduce communication overhead. However, they impose high computation costs on the user side and cannot support public data auditing. Later, Chaudhari et al. [89] suggested the data auditing technique based on modern Indistinguishability Obfuscation, a modern encryption construct that used one-way hash functions. The main goal of their study is to address public verification, dynamic data, collusion resistance, and

privacy-preserving. Though, it cannot support batch auditing to reduce computation overhead at TPA, including high verification time, especially for mobile devices.

IV. Comparison of Data Integrity Auditing Techniques

This section offers a comprehensive comparative analysis of the techniques discussed in Section III, focusing on the strengths, weaknesses, and key attributes of remote data integrity audit protocols. The analysis is systematically presented in Table I and Table II. Table I details various data integrity schemes, examining their advantages and disadvantages. Meanwhile, Table II outlines the essential characteristics of state-of-the-art data integrity auditing protocols. These characteristics encompass Public Auditing, Dynamic Data Support, Privacy Preservation, Blockless Verification, Support for Unlimited Queries, Batch Verification, and Data Sharing capabilities.

TABLE I. Merits and Demerits of Data Integrity Schemes

Schemes	Merits	Demerits
Ateniese et al. [34]	Used RSA-based homomorphic encryption to provide public data auditing. Their S-PDP shame ensures data possession, and E-PDP improved proficiency with weak data possession.	This work does not address dynamic data operation. Hence, direct extension raises security, privacy, and system design issues.
Ateniese et al. [60]	Improve the PDP protocol proposed by [34] and provide some dynamic data operations.	It is limited to the number of challenges during the audit phase and supports only limited dynamic data operations; for instance, unable to support block insertion operations.
Bowers et al. [61]	Improves proficiency and security of the existing techniques and efficiency against the active mobile adversary.	Restricted to static data and required to store one segment of each file locally, limited to the number of challenge queries.
Zhu et al., [90]	It used the same construction of a message authentication code proposed in [61] by combining universal hashing with PRFs.	It is unappropriated for the system that requires substantial data read operations. It minimized the storage cost by increasing the communication overhead.
Xu et al. [62]	They considered the private data auditing and efficiency issues of the scheme designed by [63] and provided an efficient public data auditing protocol.	Their protocol is restricted to static data only; it does not consider the privacy leakage to the TPA
Stefanov et al. [64]	Iris supports publicly verifiable dynamic data auditing by using the MHT. It provides a proficient read and writes data operation.	It requires substantial bandwidth, server computation time, local data space, and server input/output to verify the data integrity.
Shi et al. [65]	It supports efficient reading and writing operations and provides a publicly provable dynamic nature of data using standard MHT.	Validate the integrity requires the high computational power of the cloud server, bandwidth, and local storage space.
Zhang et al. [70]	It provides privacy-preserving public data auditing PoR-based scheme with aggregate verification.	It is not suitable for dynamic data environments and is only limited to static files.
Yu et al. [69]	It provides a zero-knowledge proof-based method to prevent data privacy leakage in the audit process.	The proposed scheme is limited to static data only.
Tan and Jia [76]	Identity-based cumulative signature is used to generate homomorphic tags. Eliminates data auditing burden on cloud users and is proficient to computational and communicational overheads compared with [47, 76].	It cannot perform dynamic data operations; the TPA can expose data content during the audit phase, which raises privacy issues.
Wang et al. [73]	It generates homomorphic aggregate tags using the user identity; it is optimized in terms of communication and computation cost compared with [34, 74].	Not suitable for resource constraints devices because it supports private audit, which generates computation and communication overheads on user-side in the audit phase.
Jiang et al. [77]	Used public-key cryptography and group signature to overcome collusion issues between the cloud and revoked users, generating the authentication code to verify the user from the revocation list.	It is unable to provide data recovery, imposes high computation and communication overhead, and the auditor is not capable of verifying impersonation attacks.
Yuan and Yu [81]	Used homomorphic authenticators with constant communication for cloud and TPA. Proficient in computation cost and reduced the storage overhead.	It is unable to provide dynamic data procedures.
Liu et al. [80]	Proxy resolves the user's absence issue by generating specific keys, using the coefficients encoding and a pseudorandom method to accomplish the privacy of the user's data.	Their approach is unable to support public data audits.
Fu et al. [78]	Use MHT to preserve verifiable data blocks. Ensures users trace the changes and recover the most recent block of data when an existing data block corrupts.	It is vulnerable to tag forgery, replace, replay, pollution, data leakage attacks, etc.
Wang et al. [79]	It does not need to manage certificates and enables inclusive data auditing, more proficient by incorporating a proxy server for processing and hosting the data file on the user's behalf.	It does not provide data privacy or recovery, and all the involved entities in the system are treated as trusted.
Fan et al. [82]	Using aggregate signatures to overcome vulnerability against untrusted CSP resists adversary attacks that select its data and target identities, optimizing computation and communication costs more than [38, 76].	Restricted to the static nature of data, privacy-preserving, Blockless verification, dynamic data operation, replay, replace, tag-forgery attacks, etc.
Youn et al. [83]	It performs secure deduplication and data integrity verification, reduces the storage overhead by removing duplicated copies, and uses linear-homomorphic authenticators with BLS signature to perform challenge-response mechanisms.	Generating high communication and computation costs on the user-side during the data deduplication phase is unsuitable for mobile device resource constraints. It does not support dynamic data updates and batch auditing.
Wang and Di [84]	it reduces the communication cost and storage overhead and improves audit proficiency by 20% compared to [63, 85].	It is unable to perform dynamic data updates, batch auditing to increase efficiency, and privacy-preserving against TPA.

Yang et al. [86]	Provides collaborative traceability of malevolent users to minimize the network congestion and waste of cloud resources, preserves identity revelation and DoS attack, and uses certificate-less cryptography.	The auditor cannot authenticate impersonation attacks if the user does not exist in the revoke list, imposing high communication and computation costs due to the static data.
Yan and Gui [87]	The auditor is unable to get the user's identity in the data audit process; it avoids the certificate management and cloud setting up a connection in the proofs creation phase to minimize the computation cost of TPA.	It is unable to provide data-dynamic and batch auditing.
Neela et al. [88]	Improve RSA algorithm with Cuckoo Filter for secure storage in semi-trusted CSPs.	Impose high computation costs on the user side and cannot support public data auditing.
Chaudhari et al. [89]	The auditor cannot get the user's identity in the data audit process; it avoids the certificate management and cloud setting up a connection in the proofs creation phase to minimize the computation cost of TPA.	It does not support batch auditing and imposes high computation and communication costs, which is unsuitable for energy-constrained mobile devices.

TABLE II. DESIRABLE FEATURES FOR STATE-OF-THE-ART DATA INTEGRITY AUDITING PROTOCOLS

Schemes	Auditing Requirements						
	Public Auditing	Dynamic Data	Privacy-Preserving	Blockless Verification	Unlimited Queries	Batch Verification	Data Sharing
Ateniese et al. [34]	✓	×	×	✓	×	×	×
Ateniese et al. [60]	×	✓	✓	×	×	×	×
Bowers et al. [61]	×	×	×	×	×	×	×
Zhu et al. [90]	×	✓	×	✓	✓	×	×
Xu et al. [62]	✓	×	×	×	×	×	×
Stefanov et al. [64]	×	✓	×	×	×	×	×
Shi et al. [65]	×	×	×	×	×	×	×
Zhang et al. [70]	✓	×	✓	×	×	×	×
Yu et al. [69]	×	×	✓	×	×	×	×
Tan and Jia [76]	✓	×	×	✓	✓	×	×
Wang et al. [73]	×	×	×	×	×	×	×
Jiang et al. [77]	✓	×	×	×	×	✓	✓
Yuan and Yu [81]	✓	×	×	×	✓	×	✓
Liu et al. [80]	✓	×	×	✓	✓	×	×
Fu et al. [78]	✓	×	×	×	×	×	✓
Wang et al. [79]	✓	×	×	×	×	×	×
Fan et al. [82]	✓	×	×	×	×	✓	×
Youn et al. [83]	✓	×	×	×	✓	×	×
Wang and Di [84]	✓	×	×	✓	✓	×	×
Yang et al. [86]	✓	×	×	×	✓	×	✓
Yan and Gui [87]	✓	×	×	✓	✓	×	✓
Neela et al. [88]	×	✓	×	×	✓	×	×
Chaudhari et al. [89]	✓	✓	✓	×	×	×	×

V. PERFORMANCE-BASED COMPARISON TECHNIQUES

A performance comparison to evaluate different protocols and data structure involved in these schemes are presented in this section. The following section will explain several open research issues to design a proficient data auditing protocol.

The cloud data storage is not indisputable and is constantly modified according to the user's request, for example, insertion, deletion, modification, and append procedures. For example, when the user's first-time outsourced data may not be complete, the user needs to update and complete it after uploading it to CSP. Alternatively, the client may want to delete obsolete or useless data from cloud storage servers, which is unavoidable for utilizing cloud services. In short, data update operations are necessary to protect cloud data storage, and several techniques have been developed for this perspective. Generally, the subsequent data techniques are commonly utilized to develop data update verification techniques for outsourcing users' data.

A. Merkle Hash Tree (MHT)

It is a famous binary hash tree data structure introduced by [42]. It can proficiently ensure and verify whether a series of data blocks is compromised or not. The tree nodes hold the hash value of relating sibling data blocks. The MHT is designed by computing the has value in pair from bottom to up and obtaining the root node's distinct hash value. The MHT is a provable and broadly explored data structure supporting data dynamics operations. The MHT-based secure cloud data verification technique was suggested in [42].

Nevertheless, the curious storage servers may pass the audit process without appropriate authentication of the block indices by generating proof with other authentic data blocks when the assaulted data blocks are compromised. In [50], researchers proposed a fine-grained data dynamics technique for remotely storing user's data. However, they assume that the cloud storage server remains honest in responding to the challenging queries over outsourced data. Later, in [53], researchers utilized the MHT in data duplication auditing; it develops an MHT for every copy of the data file and uses the root node to construct the two-level data structure. However, in this way, the usage of MHT cannot support sequence number authentication issues.

To overcome these issues, the study [51] proposed a multi-replica technique that integrates the other parameters, including node level and node number accessible from the node. Boolean value denotes the node either on left or right node location to its parent node in auxiliary authentication path. In this manner, multiple-replica MHT can design to validate data update operations and authenticate the block indices proficiently.

B. Ranked-Based Skip List (RASL)

It is an authentication model that verifies the integrity of data blocks and provides data dynamics operations. It is a hierarchical key-value pair storage data structure like a tree, where nodes are arranged concerning their corresponding keys. By using this technique, every node v stores two parameters $right(v)$ and $down(v)$ with the goal that a particular data node can position in searching procedure. The study [91] introduced the RSAL-based first PDP technique. In their proposed structure, every node stores $right(v)$, $down(v)$, and produced $f(v)$ by iteratively using hash method to $f(right(v))$ and $f(down(v))$. The root value use to validate server's response in challenge-proof phase. The primary disadvantage of this technique is the absence of checking the single block integrity. In [41], the researchers enhanced the RSAL to multi-replica data dynamics. The enhancement integrates all data blocks of all data copies in an identical structure, supporting the proficient data auditing mechanism for copies. Moreover, using M-RASL can minimize the communication cost for updating authentication of outsourced data with multi-replicas.

C. Index Hash Table (IHT)

IHT stores the data block modification and supports to creation hash value of every data block during authentication. The IHT design is identical to an array, which involves indices sequence i_{no}, block sequence b_{no}, block version sequence v_{no} and arbitrary value r_{no}. It was first proposed in [85] for cloud storage, decreasing computational and communicational costs by keeping the index hash table on the auditor rather than the storage server. Insertion and deletion procedures reason for the change of b_{no}, thus suffering re-generation of compromised data blocks.

A subsequent procedure is to change the arrangement of specific components, such as $(i_{no}, b_{no}, v_{no}, r_{no})$ [31]. In [47], researchers modified the components to (i_{no}, V_i, R_i, S_i), where V_i denotes the virtual index, S_i represents the signer identity of the data block. In particular, guarantees that every data block is in the proper order, and V_i of the inserted data block m_i' use the smallest number of $m_i' = (V_i + V_{i-1})/2$. Thus, the user can adequately run the data insert procedure for shared data.

Additionally, there are two more distinctive data structures, like Dynamic Hash Table (DHT) [92] and Novel Data Structure (NDS), introduced by [49], respectively. Endeavoring enhanced data update operations proficiency, a DHT-based publicly provable data auditing technique was introduced to ensure files and block-level modification. The DHT is innovative two-dimensional information stored by the auditor, minimizing communicational costs during data auditing and modification authentication.

Even though it provides better results than the existing auditing techniques, it still has a few deficiencies. First, the cloud storage server may suffer a collision attack between user and auditor because the user creates a time-stamp for validation. The auditor has just approved the user. Furthermore, it does not integrate the indices and position sequence of the data blocks. To address these problems, NDS was intended to support data dynamics operations comprising doubly-link information and position array.

The user with a novel data structure can perform insertion and deletion procedures without affecting other data blocks. Likewise, NDS effectively controls the connection between the given blocks of data and their particular position can also be appropriate for multi-copies. The user may change auditor inappropriately due to geographic location or price effect. NDS should be re-structured for every change, leading to extra overhead compared to implementing the developed data structure stored by the cloud server.

VI. Open Research Issues and Challenges

Several RDA schemes are currently proposed to ensure outsourced users' data in cloud environments. However, some problems that require attention as an open research direction are described below:

A. Dynamic Data Update

In the static data auditing approach, to change the location or update a single data bit, the user needs to modify more than half bits of files after downloading the entire data and again hosted on the cloud storage server. It creates high computation, communication, and storage overheads for the user, cloud, and the auditor. Consequently, dynamic data operation is a fundamental property of the outsourced data integrity audit techniques, for example, electronic records and log files in cloud computing and mobile cloud computing. Nonetheless, the main restriction of PoR and POW-based schemes is the data updates procedure [91].

Even though the study [93] introduced a PoR-based scheme for avoiding data update issues in cloud infrastructure, private data auditing makes this technique unusable for mobile cloud computing. Furthermore, existing dynamic data update strategies also suffer computation costs on the user side, particularly for resource constraints devices. Accordingly, allowing mobile clients to update their data effectively requires future research and improvements.

B. Shared Data Access Control

Currently, several CSPs offer services, including online blogs, web services, and web applications required to store their data on remote cloud storage servers. These clients must gain access to their data anytime, anywhere, and simultaneously execute data update operations. For example, most hosted websites support their users to update data freely. Nonetheless, most existing techniques guaranteeing data integrity cannot fully achieve these requirements or impose high computation and storage overhead on the user side.

C. Data Privacy Issue

Supporting data privacy has been essential to meeting the SLA for cloud storage services. A public data auditing approach must not expose data privacy to third-party auditors. An auditor must be proficient in securely conducting the audit process regardless of any security risk of exposing the data content. For privacy-preserving, a MAC-based solution can use for users' data. The TPA generates cloud storage challenges for data integrity by arbitrarily selecting data blocks and respective MAC. The cloud storage replies with data blocks and MAC, and then the TPA verifies it using the secret key. However, as the server responds, a linear data blocks sequence to TPA, so the auditor can breach the SLA between the user and TPA. Moreover, it supports static data and restricts the number of challenges. The user must retrieve the complete data, generate the new keys, and host it on the storage servers, imposing significant computation and communication costs. Homomorphic authenticators with arbitrary masking can use to overcome these issues [38].

D. Blockless Data Auditing

An audit strategy that does not utilize accumulation signature and verification tag need the cloud server to respond to the requested data blocks for integrity assurance. This approach imposes a high communication cost on the cloud server and affects audit proficiency. The study Wang et al. [42] introduced a Blockless auditing approach that verifies tags rather than validating actual data blocks in the auditing phase. Even though this approach can improve the proficiency of the data auditing strategy and highly minimizes the communication cost, it might allow the CSP to cheat. Assume the user wants to achieve the data update procedure, which is possible because the storage server possesses old data and signatures. As the signatures and data are legitimate, the auditor might not recognize whether they are appropriately updated or not.

E. Deletion Assurance

It is required to assure the data delete operation; else, it might result in a data breach for the cloud storage [94]. Another significant perspective in guaranteed data deletion is that different versions of data files with similar data contents of the deleted identical copies or files must be kept secure. Rahumed et al. [95] introduced a fine-grained version control backup system named "FADE" to ensure deletion operation. The deleted versions of files are guaranteed to be forever unavailable. In [96], the researchers describe mechanisms for securely assuring data deletion, the classification of adversaries, and capacities for exploiting the cloud servers without secured deletion. This term can also be named self-destruction in [97],[98],[99] and [100].

VII. Future Research Work

Cloud computing has a rapidly growing business model and evolving new features to facilitate users. The researchers can take advantage to improve the data auditing schemes. The future research direction concerning data auditing with cloud computing is following.

A. Geolocation Assurance

The CSPs are restricted by Service Level Arrangement (SLA) to outsource users' data in a specific geographic area with a particular time zone, political border, state, or city level. For administration reasons and law variance, some cloud clients needed to guarantee data geolocation [101], [102], [103] and [104]. The CSP may transfer users' data to abroad servers for less expensive IT costs, disregarding the SLA agreement. Such activities of the CSP may disclose the client's data to foreign governments, who can assess it via court orders or any lawful approach. In this situation, timely recognition proof of purposeful deceitfulness or breach of SLA with CSPs is a fundamental need for cloud clients. The data integrity verification technique must incorporate geolocation confirmation for future research work.

B. Big Data Auditing

Big data auditing of RDA schemes is challenging because storage, communication, and computation costs may exponentially increase when data size increases. The cloud providers may also delete rarely used data to save storage costs without intimating the user and try to hide data loss info for the sack of their repute [105], [106], [107] and [108]. Consequently, it needs to develop an RDA scheme for supporting big-data auditing to minimize storage overhead, computation complexity, and communication cost.

C. Support for Collaborative Auditing

To audit the user's data in a multi-cloud environment is named collaborative auditing. Several data auditing schemes [34], [38], [42], [44], [60], [63], and [91] proposed for a single cloud cannot support a multi-cloud environment. Distributed File Systems were introduced to cloud storage systems for local independence and low cost for owner's data. However, collaborative auditing is challenging for job assignments, security assurance, and communication costs. Furthermore, reducing the computation overhead, storage cost, and system usability should also be considered when designing data auditing protocols.

D. Data Auditing with Deduplication

The cloud server store duplicates data, especially in multi-replica that generates different copies of the same data file, creating storage overhead for identical copies. It is significant to resolve this problem by improving the RDA schemes. Currently, [109], [110], and [111] were suggested data auditing techniques with PoW to overcome this issue considering only single data files. However, how to improve the multi-replica RDA scheme with PoW mechanism to ensure data security still seems a vital research contribution in the future.

E. Blockchain-Based Data Auditng

The blockchain is an undisputable distributed accounting, consensus approach, and intelligent contract technology with

asymmetric cryptography, ensuring data security and privacy. The researchers can take advantage of decentralizing features that could generally use in data integrity auditing schemes [112], [113] and [114] for single and distributed infrastructures. However, improving the security framework of data auditing techniques for single and multiple copies scenario could be an auspicious future research direction.

F. Data Auditing in Fog Computing

Fog computing is an extension of cloud computing to facilitate the edge network for easy access and fast corresponding services for end-users. Users can use multiple devices as fog nodes to support cloud servers and reduce CSP charges. Recently, Wang et al. [115] suggested a fog-based secure and anonymous data auditing scheme. Even though the scheme is efficient and secure; however, they consider the cloud a fully trusted entity. Therefore, secure fog-based data auditing is still an open research direction for cloud computing.

G. Data Auditing in Edge Computing

Edge computing is a distributed computing paradigm that provides computational and storage facilities closer to the end-user for fast response times (low latency) and reducing bandwidth utilization. It is an ongoing business model that requires improving data auditing techniques to incorporate cloud computing changes. However, duplicated data may lead to storage problems for single and shared clouds; auditing techniques must include deduplication [116] and [117] and ensure data security and privacy in edge computing. Furthermore, the researchers can explore the edge computing model for data auditing with blockchain technology in future research. Cloud computing saves time and monitoring costs for any organization and turns technological solutions for large-scale systems into server-to-service frameworks [118].

VIII. Conclusion

The paper has highlighted the significance of remote data auditing techniques, involved entities, and the concerns related to public and private data auditing. We thoroughly reviewed existing data integrity techniques by exploring their merits and demerits. Also, we introduced the qualitative comparison of auditing techniques and compared their performance regarding communication and computational overhead. Lastly, we highlighted the desirable features of different schemes, open issues, and future research directions for designing efficient and secure data auditing schemes for the cloud environment.

Acknowledgment

This research was supported by Ministry of Higher Education (MoHE) of Malaysia through Fundamental Research Grant Scheme (FRGS/1/2018/ICT04/UUM/02/17).

References

[1] X. Ma, H. Gao, H. Xu, and M. Bian, "An IoT-based task scheduling optimization scheme considering the deadline and cost-aware scientific workflow for cloud computing," EURASIP Journal on Wireless Communications Networking, vol. 2019, no. 1, pp. 1-19, 2019.

[2] Y. Yin, L. Chen, Y. Xu, J. Wan, H. Zhang, and Z. Mai, "QoS prediction for service recommendation with deep feature learning in edge computing environment," Mobile Networks and Applications, pp. 1-11, 2019.

[3] H. Gao, L. Kuang, Y. Yin, B. Guo, and K. Dou, "Mining consuming behaviors with temporal evolution for personalized recommendation in mobile marketing apps," Mobile Networks and Applications, vol. 25, pp. 1233-1248, 2020.

[4] Linode. "Linode LCC company." https://www.linode.com/ (accessed May 25, 2021).

[5] Amazon. "Amazon Elastic Compute Cloud." https://aws.amazon.com/ec2/ (accessed 24 May, 2021).

[6] Google. "Accelerate your transformation with Google Cloud." https://cloud.google.com/ (accessed 24 May, 2022).

[7] Azure. "Microsoft Azure." https://azure.microsoft.com/en-us/ (accessed March 14, 2022).

[8] X. Yang, S. Zhou, and M. Cao, "An approach to alleviate the sparsity problem of hybrid collaborative filtering based recommendations: the product-attribute perspective from user reviews," Mobile Networks and Applications, pp. 1-15, 2019.

[9] H. Gao, C. Liu, Y. Li, and X. Yang, "V2VR: reliable hybrid-network-oriented V2V data transmission and routing considering RSUs and connectivity probability," IEEE Transactions on Intelligent Transportation Systems, 2020.

[10] P. Mell and T. Grance, "The NIST definition of cloud computing," 2011.

[11] E. Simmon, "Evaluation of cloud computing services based on NIST SP 800-145," NIST Special Publication, vol. 500, p. 322, 2018.

[12] C. M. Mohammed and S. R. Zebaree, "Sufficient comparison among cloud computing services: IaaS, PaaS, and SaaS: A review," International Journal of Science Business, vol. 5, no. 2, pp. 17-30, 2021.

[13] S. Srinivasan, Cloud computing basics. Springer, 2014.

[14] A. Alrabea, "A modified Boneh-Lynn-Shacham signing dynamic auditing in cloud computing," Journal of King Saud University-Computer Information Sciences, 2020.

[15] I. E. Baciu, "Advantages and disadvantages of cloud computing services, from the employee's point of view," National Strategies Observer No.2/Vol.1, 2015, vol. 2, 2015.

[16] Statista, "Public cloud services end-user spending worldwide from 2017 to 2022," 2019. [Online]. Available: https://www.statista.com/ statistics/ 273818/global-revenue-generated-with-cloud-computing-since-2009/.

[17] B. Nedelcu, M.-E. Stefanet, I.-F. Tamasescu, S.-E. Tintoiu, and A. Vezeanu, "Cloud computing and its challenges and benefits in the bank system," Database Systems Journal, vol. 6, no. 1, pp. 44-58, 2015.

[18] B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," IEEE transactions on cloud computing, vol. 2, no. 1, pp. 43-56, 2014.

[19] K. Thomas. "Microsoft cloud data breach heralds things to come." https://www.pcworld.com/article/214775/microsoft_cloud_data_breach_ sign_of_future.html (accessed July 2021).

[20] I. Orton, A. Alva, and B. Endicott-Popovsky, "Legal process and requirements for cloud forensic investigations," in Cybercrime and Cloud Forensics: Applications for Investigation Processes: IGI Global, 2013, pp. 186-229.

[21] R. Yeluri and E. Castro-Leon, "Cloud computing basics," in Building the infrastructure for cloud security: Springer, 2014, pp. 1-17.

[22] R. Ko, S. G. Lee, and V. Rajan, "Cloud computing vulnerability incidents: A statistical overview," Cloud Security Alliance, 2013.

[23] ISACA, "Isaca Glossary," 2015.

[24] A. Team. "Amazon S3 availability event." https://status.aws.amazon.com/s3-20080720.html (accessed April 14, 2022).

[25] A. iCloud. "Apple iCloud." https://www.icloud.com/ (accessed May 29, 2021).

[26] T. Mather, S. Kumaraswamy, and S. Latif, Cloud security and privacy: an enterprise perspective on risks and compliance. " O'Reilly Media, Inc.", 2009.

[27] S. Yu, W. Lou, and K. Ren, "Data security in cloud computing," Morgan Kaufmann/Elsevier, Book section, vol. 15, pp. 389-410, 2012.

[28] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in Annual international cryptology conference, 1996: Springer, pp. 1-15.

[29] M. Zhou, R. Zhang, W. Xie, W. Qian, and A. Zhou, "Security and privacy in cloud computing: A survey," presented at the 2010 Sixth International Conference on Semantics, Knowledge and Grids, 2010.

[30] G. Ateniese, S. Kamara, and J. Katz, "Proofs of storage from homomorphic identification protocols," presented at the International conference on the theory and application of cryptology and information security, 2009.

[31] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," IEEE transactions on parallel and distributed systems, vol. 24, no. 9, pp. 1717-1726, 2013.

[32] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," IEEE transactions on computers, vol. 62, no. 2, pp. 362-375, 2013.

[33] C. Gentry, "Fully homomorphic encryption using ideal lattices," presented at the Proceedings of the forty-first annual ACM symposium on Theory of computing, 2009.

[34] G. Ateniese et al., "Provable data possession at untrusted stores," presented at the Proceedings of the 14th ACM conference on Computer and communications security, 2007.

[35] L. A. B. Silva, C. Costa, and J. L. Oliveira, "A common API for delivering services over multi-vendor cloud resources," Journal of Systems and Software, vol. 86, no. 9, pp. 2309-2317, 2013.

[36] D. Boneh and D. M. Freeman, "Homomorphic signatures for polynomial functions," presented at the annual international conference on the theory and applications of cryptographic techniques, 2011.

[37] A. L. Ferrara, M. Green, S. Hohenberger, and M. Ø. Pedersen, "Practical short signature batch verification," in Cryptographers' Track at the RSA Conference, 2009: Springer, pp. 309-324.

[38] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," presented at the Infocom, 2010 proceedings ieee, 2010.

[39] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple-replica provable data possession," presented at the the 28th international conference on distributed computing systems, 2008.

[40] Y. Zhang, J. Ni, X. Tao, Y. Wang, and Y. Yu, "Provable multiple replication data possession with full dynamics for secure cloud storage," Concurrency Computation: Practice Experience, vol. 28, no. 4, pp. 1161-1173, 2016.

[41] A. Abo-alian, N. L. Badr, and M. F. Tolba, "Integrity as a service for replicated data on the cloud," Concurrency Computation: Practice Experience, vol. 29, no. 4, p. e3883, 2017.

[42] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," IEEE transactions on parallel and distributed systems, vol. 22, no. 5, pp. 847-859, 2011.

[43] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "A survey of two signature aggregation techniques," ed: Citeseer, 2003.

[44] A. Juels and B. S. Kaliski Jr, "PORs: Proofs of retrievability for large files," presented at the Proceedings of the 14th ACM conference on Computer and communications security, 2007.

[45] H. Wang, "Proxy provable data possession in public clouds," IEEE Transactions on Services Computing, vol. 6, no. 4, pp. 551-559, 2013.

[46] H. Wang, Q. Wu, B. Qin, and J. Domingo-Ferrer, "FRR: Fair remote retrieval of outsourced private medical records in electronic health networks," Journal of biomedical informatics, vol. 50, pp. 226-233, 2014.

[47] B. Wang, B. Li, and H. Li, "Panda: Public auditing for shared data with efficient user revocation in the cloud," IEEE Transactions on services computing, vol. 8, no. 1, pp. 92-106, 2015.

[48] G. Wu, Y. Mu, W. Susilo, and F. Guo, "Privacy-preserving cloud auditing with multiple uploaders," presented at the International Conference on Information Security Practice and Experience, 2016.

[49] J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data," IEEE Transactions on Information Forensics and Security, vol. 12, no. 10, pp. 2402-2415, 2017.

[50] C. Liu et al., "Authorized public auditing of dynamic big data storage on cloud with efficient verifiable fine-grained updates," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 9, pp. 2234-2244, 2014.

[51] C. Liu, R. Ranjan, C. Yang, X. Zhang, L. Wang, and J. Chen, "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, vol. 64, no. 9, pp. 2609-2622, 2015.

[52] A. F. Barsoum and M. A. Hasan, "Provable possession and replication of data over cloud servers," Centre For Applied Cryptographic Research , University of Waterloo, Report, vol. 32, p. 2010, 2010.

[53] A. F. Barsoum and M. A. Hasan, "On Verifying Dynamic Multiple Data Copies over Cloud Servers," IACR Cryptol. ePrint Arch., vol. 2011, p. 447, 2011.

[54] H. Wang, Q. Wu, B. Qin, and J. Domingo-Ferrer, "Identity-based remote data possession checking in public clouds," IET Information Security, vol. 8, no. 2, pp. 114-121, 2014.

[55] H. Wang, "Identity-based distributed provable data possession in multicloud storage," IEEE Transactions on Services Computing, vol. 8, no. 2, pp. 328-340, 2015.

[56] Y. Yu et al., "Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage," IEEE Transactions on Information Forensics Security, vol. 12, no. 4, pp. 767-778, 2016.

[57] B. Wang, B. Li, H. Li, and F. Li, "Certificateless public auditing for data integrity in the cloud," presented at the 2013 IEEE conference on communications and network security (CNS), 2013.

[58] D. He, S. Zeadally, and L. Wu, "Certificateless public auditing scheme for cloud-assisted wireless body area networks," IEEE Systems Journal, vol. 12, no. 1, pp. 64-73, 2015.

[59] D. He, N. Kumar, H. Wang, L. Wang, and K.-K. R. Choo, "Privacy-preserving certificateless provable data possession scheme for big data storage on cloud," Applied Mathematics Computation, vol. 314, pp. 31-43, 2017.

[60] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," presented at the Proceedings of the 4th international conference on Security and privacy in communication netowrks, 2008.

[61] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A high-availability and integrity layer for cloud storage," presented at the Proceedings of the 16th ACM conference on Computer and communications security, 2009.

[62] J. Xu and E.-C. Chang, "Towards efficient proofs of retrievability," presented at the Proceedings of the 7th ACM symposium on information, computer and communications security, 2012.

[63] H. Shacham and B. Waters, "Compact proofs of retrievability," presented at the International Conference on the Theory and Application of Cryptology and Information Security, 2008.

[64] E. Stefanov, M. van Dijk, A. Juels, and A. Oprea, "Iris: A scalable cloud file system with efficient integrity checks," presented at the Proceedings of the 28th Annual Computer Security Applications Conference, 2012.

[65] E. Shi, E. Stefanov, and C. Papamanthou, "Practical dynamic proofs of retrievability," presented at the Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 2013.

[66] R. C. Merkle, "A certified digital signature," presented at the Conference on the Theory and Application of Cryptology, 1989.

[67] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," Future Generation computer systems, vol. 28, no. 3, pp. 583-592, 2012.

[68] C. Li, Y. Chen, P. Tan, and G. Yang, "Towards comprehensive provable data possession in cloud computing," Wuhan University Journal of Natural Sciences, vol. 18, no. 3, pp. 265-271, 2013.

[69] Y. Yu et al., "Enhanced privacy of a remote data integrity-checking protocol for secure cloud storage," International Journal of Information Security, vol. 14, no. 4, pp. 307-318, 2015.

[70] J. Zhang, W. Tang, and J. Mao, "Efficient public verification proof of retrievability scheme in cloud," Cluster computing, vol. 17, no. 4, pp. 1401-1411, 2014.

[71] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, and H. Hu, "Zero-knowledge proofs of retrievability," Science China Information Sciences, vol. 54, no. 8, p. 1608, 2011.

[72] Z. Hao, S. Zhong, and N. Yu, "A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability," IEEE transactions on Knowledge and Data Engineering, vol. 23, no. 9, pp. 1432-1437, 2011.

[73] H. Wang, Q. Wu, B. Qin, and J. Domingo-Ferrer, "Identity-based remote data possession checking in public clouds," IET Information Security, vol. 8, no. 2, pp. 114-121, 2014.

[74] F. Sebé, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 8, pp. 1034-1038, 2008.

[75] J. Zhang, P. Li, and J. Mao, "IPad: ID-based public auditing for the outsourced data in the standard model," Cluster Computing, vol. 19, no. 1, pp. 127-138, 2016.

[76] S. Tan and Y. Jia, "NaEPASC: a novel and efficient public auditing scheme for cloud data," Journal of Zhejiang University SCIENCE C, vol. 15, no. 9, pp. 794-804, 2015.

[77] T. Jiang, X. Chen, and J. Ma, "Public integrity auditing for shared dynamic cloud data with group user revocation," IEEE Transactions on Computers, vol. 65, no. 8, pp. 2363-2373, 2016.

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

[79] Y. Wang, Q. Wu, B. Qin, W. Shi, R. H. Deng, and J. Hu, "Identity-based data outsourcing with comprehensive auditing in clouds," IEEE Transactions on Information Forensics and Security, vol. 12, no. 4, pp. 940-952, 2017.

[80] J. Liu, K. Huang, H. Rong, H. Wang, and M. Xian, "Privacy-preserving public auditing for regenerating-code-based cloud storage," IEEE Transactions on Information Forensics and Security, vol. 10, no. 7, pp. 1513-1528, 2015.

[81] J. Yuan and S. Yu, "Public integrity auditing for dynamic data sharing with multiuser modification," IEEE Transactions on Information Forensics and Security, vol. 10, no. 8, pp. 1717-1726, 2015.

[82] Y. Fan, X. Lin, G. Tan, Y. Zhang, W. Dong, and J. Lei, "One secure data integrity verification scheme for cloud storage," Future Generation Computer Systems, vol. 96, pp. 376-385, 2019.

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

[84] C. Wang and X. Di, "Research on Integrity Check Method of Cloud Storage Multi-Copy Data Based on Multi-Agent," IEEE Access, vol. 8, pp. 17170-17178, 2020.

[85] Y. Zhu, G.-J. Ahn, H. Hu, S. S. Yau, H. G. An, and C.-J. Hu, "Dynamic audit services for outsourced storages in clouds," IEEE Transactions on Services Computing, vol. 6, no. 2, pp. 227-238, 2013.

[86] X. Yang, M. Wang, T. Li, R. Liu, and C. Wang, "Privacy-Preserving Cloud Auditing for Multiple Users Scheme With Authorization and Traceability," IEEE Access, vol. 8, pp. 130866-130877, 2020.

[87] H. Yan and W. Gui, "Efficient Identity-Based Public Integrity Auditing of Shared Data in Cloud Storage With User Privacy Preserving," IEEE Access, vol. 9, pp. 45822-45831, 2021.

[88] K. Neela and V. Kavitha, "An Improved RSA Technique with Efficient Data Integrity Verification for Outsourcing Database in Cloud," Wireless Personal Communications, pp. 1-18, 2022.

[89] S. Chaudhari and G. Swain, "Towards Lightweight Provable Data Possession for Cloud Storage Using Indistinguishability Obfuscation," IEEE Access, vol. 10, pp. 31607-31625, 2022.

[90] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, "Efficient provable data possession for hybrid clouds," presented at the Proceedings of the 17th ACM conference on Computer and communications security, 2010.

[91] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," presented at the Proceedings of the 16th ACM conference on Computer and communications security, Chicago, Illinois, USA, 2009.

[92] H. Tian et al., "Dynamic-hash-table based public auditing for secure cloud storage," IEEE Transactions on Services Computing, 2015.

[93] D. Cash, A. Küpçü, and D. Wichs, "Dynamic proofs of retrievability via oblivious RAM," Journal of Cryptology, vol. 30, no. 1, pp. 22-57, 2013.

[94] Y. Tang, P. P. Lee, J. C. Lui, and R. Perlman, "Secure overlay cloud storage with access control and assured deletion," IEEE Transactions on dependable secure computing, vol. 9, no. 6, pp. 903-916, 2012.

[95] A. Rahumed, H. C. Chen, Y. Tang, P. P. Lee, and J. C. Lui, "A secure cloud backup system with assured deletion and version control," presented at the 2011 40th International Conference on Parallel Processing Workshops, 2011.

[96] J. Reardon, D. Basin, and S. Capkun, "Sok: Secure data deletion," presented at the 2013 IEEE symposium on security and privacy, 2013.

[97] C. Cachin, K. Haralambiev, H.-C. Hsiao, and A. Sorniotti, "Policy-based secure deletion," presented at the Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 2013.

[98] C. Li, Y. Chen, and Y. Zhou, "A data assured deletion scheme in cloud storage," China Communications, vol. 11, no. 4, pp. 98-110, 2014.

[99] A. B. Habib, T. Khanam, and R. Palit, "Simplified file assured deletion (sfade)-a user friendly overlay approach for data security in cloud storage system," presented at the 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013.

[100] J. Xiong, Z. Yao, J. Ma, X. Liu, and Q. Li, "A secure document self-destruction scheme: an ABE approach," presented at the 2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing, 2013.

[101] A. Albeshri, C. Boyd, and J. G. Nieto, "Enhanced geoproof: improved geographic assurance for data in the cloud," International Journal of Information Security, vol. 13, no. 2, pp. 191-198, 2014.

[102] D. L. Fu, X. G. Peng, and Y. L. Yang, "Trusted validation for geolocation of cloud data," The Computer Journal, vol. 58, no. 10, pp. 2595-2607, 2015.

[103] M. Gondree and Z. N. Peterson, "Geolocation of data in the cloud," presented at the Proceedings of the third ACM conference on Data and application security and privacy, 2013.

[104] T. Jiang, W. Meng, X. Yuan, L. Wang, J. Ge, and J. Ma, "ReliableBox: Secure and Verifiable Cloud Storage With Location-Aware Backup," IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 12, pp. 2996-3010, 2021.

[105] L. Liu, O. De Vel, Q.-L. Han, J. Zhang, and Y. Xiang, "Detecting and preventing cyber insider threats: A survey," IEEE Communications Surveys and Tutorials, vol. 20, no. 2, pp. 1397-1417, 2018.

[106] H. Tabrizchi and M. K. Rafsanjani, "A survey on security challenges in cloud computing: issues, threats, and solutions," The journal of supercomputing, vol. 76, no. 12, pp. 9493-9532, 2020.

[107] J. Ni, Y. Yu, Y. Mu, and Q. Xia, "On the security of an efficient dynamic auditing protocol in cloud storage," IEEE Transactions on Parallel Distributed Systems, vol. 25, no. 10, pp. 2760-2761, 2014.

[108] N. Dhakad and J. Kar, "EPPDP: An Efficient Privacy-Preserving Data Possession With Provable Security in Cloud Storage," IEEE Systems Journal, 2022.

[109] C. Li and Z. Liu, "A Secure Privacy-Preserving Cloud Auditing Scheme with Data Deduplication," Int. J. Netw. Secur., vol. 21, no. 2, pp. 199-210, 2019.

[110] E. Daniel and N. Vasanthi, "LDAP: a lightweight deduplication and auditing protocol for secure data storage in cloud environment," Cluster Computing, vol. 22, no. 1, pp. 1247-1258, 2019.

[111] J. Yuan and S. Yu, "Secure and constant cost public cloud storage auditing with deduplication," presented at the 2013 IEEE Conference on Communications and Network Security (CNS), 2013.

[112] C. Li, J. Hu, K. Zhou, Y. Wang, and H. Deng, "Using blockchain for data auditing in cloud storage," in International Conference on Cloud Computing and Security, 2018: Springer, pp. 335-345.

[113] J. Xue, C. Xu, J. Zhao, and J. Ma, "Identity-based public auditing for cloud storage systems against malicious auditors via blockchain," Science China Information Sciences, vol. 62, no. 3, pp. 1-16, 2019.

[114]Y. Qi and Y. Huang, "DIRA: Enabling decentralized data integrity and reputation audit via blockchain," Sci. China Technological Sci, vol. 62, pp. 698-701, 2019.

[115]H. Wang, Z. Wang, and J. Domingo-Ferrer, "Anonymous and secure aggregation scheme in fog-based public cloud computing," Future Generation Computer Systems, vol. 78, pp. 712-719, 2018.

[116]D. Liu, Z. Yan, W. Ding, and M. Atiquzzaman, "A survey on secure data analytics in edge computing," IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4946-4967, 2019.

[117]B. Cao, L. Zhang, Y. Li, D. Feng, and W. Cao, "Intelligent offloading in multi-access edge computing: A state-of-the-art review and framework," IEEE Communications Magazine, vol. 57, no. 3, pp. 56-62, 2019.

[118]M. Dawood, S. Tu, C. Xiao, H. Alasmary, M. Waqas, and S. U. Rehman, “Cyberattacks and Security of Cloud Computing: A Complete Guideline,” Symmetry, vol. 15, no. 11, p. 1981, Oct. 2023, doi: 10.3390/sym15111981.