INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Dynamic Attribute Tree for the Data Encryption and Third Party Auditing for Cloud Storage

M B Rajashekar[1]*, S Meenakshi Sundaram[1]

[1] Department of Computer Science & Engineering, GSSS Institute of Engineering & Technology for Women, Affiliated to VTU, Belagavi, Karnataka, India

OPEN ACCESS

Received: 28-12-2021

Accepted: 19-03-2022

Published: 10.05.2022

Citation: Rajashekar MB, Sundaram SM (2022) Dynamic Attribute Tree for the Data Encryption and Third Party Auditing for Cloud Storage. Indian Journal of Science and Technology 15(17): 798-805. https://doi.org/10.17485/IJST/v15i17.2413

* **Corresponding author**.

drsmsrmb@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Background: To reduce the burden of the users to perform cloud audits. The tree based key handling method provides the structure format of data and increases performance in the search process. **Methods: T**he Dynamic Attribute Tree method is proposed for Third-Party auditing to encrypt the data based on its attribute. The attributes are stored in tree structure along with data and key that helps to effectively improve the dynamic update of the data in the cloud. **Findings:** The proposed Dynamic Attribute Tree method has 6.7milliseconds computation time for 100 blocks of data and the existing Merkel Tree method has 7.5milliseconds computation time. **Novelty:** The Novelty of the project is the Dynamic Attribute Tree method uses bilinear mapping to verify the integrity of the data without retrieving the actual data from the cloud.

Keywords: Bilinear Mapping; Cloud audit; Cloud storage services; Dynamic Attribute Tree; and Third Party Auditing

1 Introduction

Cloud storage is an important part of the cloud computing that allows the data owners to store their data in cloud servers in low cost, powerful outsourcing, and scalable storage services. Many numbers of individuals and organization uses the cloud services to store their personal data due to cloud storage management and cost [(1)]. Security is the major concern in cloud storage and especially for storing location information in the cloud. For instance, the incomplete data damage the value of the data and affects the decision making process etc. The integrity check is essential in outsourced data in cloud storage to overcome the security issues like unauthorized access and data damage problems [(2)]. Outsourced data suffers from the various types of attacks of both internal and external in cloud computing. Malicious networks attacks are network attacks (external) that affects the cloud data. Hackers steal, corrupt and delete the user data in cloud that affects its availability, integrity, and confidentiality [(3)]. Cloud Service Provider requires the effective model to improve the security in the cloud computing.

Data integrity is major security constraints and setback for users in cloud storage. Cloud auditing provides the effective solution for the users to check the integrity of the cloud data. Cloud auditing method requires the process of verification and this involves in require the time for process. Cloud users have limited resources that hinder the auditing like high computation jobs. Third Party Auditing method is introduced to perform the cloud audit on behalf of the user to reduce the burden for users. The Third Party Auditing works on three-tier architecture and distributes the batch of auditing in multiple Third Party Auditing based on cloud users of load balancing techniques[4,5]. This solution considers TPA behaves in honest way and in practice; this is not a reliable premise. For instance, TPA colludes with CSP to hide data corruption or data owner to deceive for penalty[6]. Recently, a number of public cloud storage auditing have been proposed. Existing methods in Third Party Auditing have limitation of lower performance in the dynamic update and high computation time[7–10]. The paper is organized as the proposed Dynamic Attribute Tree method is explained in section 2, result is given in section 3 and conclusion is given in section 4.

Cloud auditing involves in checking the integrity of user data in cloud storage to ensure the data reliability. Third party auditing method performs cloud auditing on behalf of user to reduce the burden of user. Recent methods in Third party auditing in cloud storage were reviewed in this section.

The collaborative auditing block chain method[11] to increases the trust between data owner and cloud service provider. Single third party auditor in this framework substitutes all consensus nodes to perform auditing delegation and store them. This helps to prevent entities from each other deceiving. Security analysis shows that developed method protect remote data integrity from various attacks. The result shows that developed method has high security and resource friendly in Third party auditing. The data integrity verification is difficult in block chain and scalability of the model is low. Block-chain based data reduplication method[12] to improve the performance of auditing. The bi-linear pairing method is applied to perform client side reduplication to reduce the burden on user and service provider. The block-chain and bi-linear mapping method effectively improve the performance of data integrity check and increases the trustworthiness. The developed method records the data outsourcing and auditing process to analysis the immutable records that is used to find the unreliable third party auditor and ensure the credibility of auditing results. The developed method lower efficiency in finding unreliable third party auditor and require more resources.

The applied Homomorphism Verification Authenticator (HVA) in Dynamic Large Branching Hash Tree method[13] in public auditing. Aggregate signature method is used in HVA to reduce computation and communication overhead in public auditing. The developed method supports the batch dynamic updating operations and batch auditing at cloud server side. The bilinear aggregate signature method support batch auditing process and storing the signature of multiple users into single one helps to reduce communication overhead. The developed method supports the data dynamic update, block verification process and proper public auditing. The model performs secure and efficient public auditing that reduces the computation and communication overhead. The model has lower performance in traceability and recoverability in the cloud system. The applied a trusted third party auditing method[14] to reduce the complexity of users in cloud. The trusted proxy server is applied to process the tags under the delegation of users. The Merkel Hash Tree and B+ tree of authentication structure is developed to support dynamic operation support and improve the efficiency of data retrieval. The hybrid method is used to develop auditing scheme for integrity check against forge attacks. The computation Diffie-Hellman problem and discrete logarithm problem are used for bilinear mapping in the random oracle model. The security and performance analysis shows that the developed method has higher security and lower communication and computation complexity. The developed method has lower efficiency against the collision attacks due to structure maintains.

The applied block chain[15] to replace Third Party Auditing and designing the smart system for public cloud storage. A block chain based smart contract is applied in Cloud Service Provider (CSP) and data owner. The proof of data possessions is submitted regularly by CSP to the contract. The contract execution interaction is reduced based on block chain contract and data possessions of non-interactive public provable. The developed method has higher performance in cloud auditing. The communication overhead of the model is high and low performance for verification of integrity.

2 Proposed Method

In this research, dynamic auditing is performed based on the attribute tree encryption method to improve the performance of auditing. The attributes are stored in tree structure along with data and key to effective handle the data.

2.1 System Model

Cloud services helps users in the digital transformatn journey. It defines the architecture, integrating security, resiliency and management models into an implementable design that meets the user's needs. Cloud solutions protect and enable organizations to be benefited from hybrid, multi-cloud models[16].Third Party Auditing operates in three steps: Integrity audit, server integrity examination, and key generation[17,18]. The owner executes key generation and owner's private key is used for data encryption and public key with data is distributed. Third Party auditing performs in server to provide data integrity evidence in server integrity proof and server will give proof. Third Party Auditor verifies the honesty on receipt of server proof without encrypting the data. The data is effectively managing by Third Party Auditing data tag and data samples routinely carried out in the audit. Figure 1 System Model of Third party auditing the samples are collected and regularly check the samples.

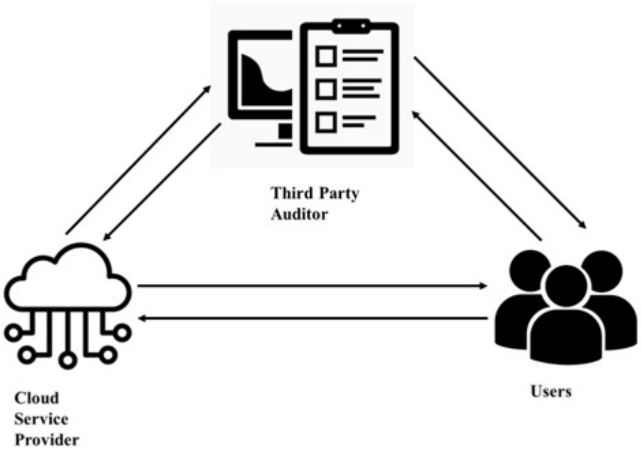

Fig 1. System model of Third party auditing

2.2 Dynamic auditing

Support for dynamic auditing is low in cloud data and it does not have sufficient static auditing. A dynamic audit is required to check dynamic data credibility. Auditing is not straightforward if data is volatile in cloud. In the auditing phase failure, server performs attack forging and attack replay. Elimination, insertion and modification are complex processes. If complex operation is done, the status of operation is sent to data owner in upgrade message. Auditor revise the table and message update of tag and new message will replace tag and m. The tag and new message m are added in the insertion process. In the index table, tag message and m is withdrawn and entries are shifted upward.

Data consistency is more important especially when two or more users use same data simultaneously. Unauthorized access by intruders enables modification of existing data, which is violation of data integrity(19). This problem has been addressed by applying bilinear mappings in Attribute-Based Encryption as discussed in section 2.3.

The public clouds data are accessed through an unauthorised person. The client has an access to check the data completeness of data that is stored in remote server. Cloud computing system have security and this is not sufficient for protect privacy of the data. Third Party auditing is used as public and private cloud services to validate the data quality to delivery different services to resolve the safety concerns.

In cloud computing, there are some lacking facets of security like misconfiguration, unauthorized access, insecure interfaces and lack of visibility. These parameters need to be improved to preserve privacy of data. Authentication method is less secure: End-customer authentication has weak authentication process like security is low in authentication management, so authentication is improved.

Cryptographic method is computationally costly: the cryptographic method applied in the cloud requires more computation time and memory to encrypt less data. A Third Party Auditor recommends the security measure for safety management that is more reliable to secure the data.

2.3 Attribute Tree Encryption

Attribute Based Encryption (ABE) consists of parties set $\{P_1, P_2, \ldots, P_n\}$. Monotonic is a collection of $A \subseteq 2^{\{P_1, P_2, \ldots, P_n\}}$if for $\forall B, C$ it is true, that if $B \in A$ and $B \subseteq C$ then$C \in A$.

Non-empty subsets of $\{P_1, P_2, \ldots, P_n\}$ of monotonic collection MAS in Monotonic access structure, i.e., $MAS \subseteq 2^{\{P_1, P_2, \ldots, P_n\}} / \{\varnothing\}$. Two sub-sets are split from whole set $2^{\{P_1, P_2, \ldots, P_n\}}$ called respectively authorized sets (belonging to MAS) and non-authorized sets.

The MAS consists of authorized attributes set and this is used to describe parties. Monotone access structures are used in this research.

Bilinear mappings are applied in Attribute-Based Encryption. Two cyclic multiplicative prime order p groups are G_1 and G_2. The G_1 generator is denoted as g. Bilinear map is denoted as e, $e : G_1 \to G_2$. This consists of following properties 1) for all $u, v \in G_1$ and $a, b \in Z_p$, have $e\left(u^a, v^b\right) = e(u,v)^{ab}$, 2) $e\left(g,g\right) \neq 1$.

An ABE method consists of four algorithms for the encryption and decryption process such as Encryption, Decryption, Key Generation and Setup.

Setup: In setup phase, a prime number p is randomly selected. The random number $t_1, \ldots, t_n, y$ is selected by trust center from Z_q finite field. The master key $MK = (t_1, \ldots, t_n, y)$ is created in this step.

Key Generation: User attributes set is applied for input of private key generation and user's private key is algorithm output. Each user U private key is generate trusted party. Random polynomial $q(x)$ is selected by trusted party such as $q(0) = y$. Equation (1) describes private key.

$$D = \left(D_i = g^{\frac{q(i)}{t_i}} \right\}_{\forall_{i \in A_U}} \tag{1}$$

Master key MK is public key in such equation (2).

$$PK = \left(T_1 = g^{t_1}, \ldots, T_n = g^{t_n}, Y = e(g,g)^y\right) \tag{2}$$

Encryption: Some input values are used in algorithm such as randomly selected number, a set of attributes, public key, and plain-text message. Users with a set of attributes are able to decrypt the data and encrypted message is an output.

Message to encrypt is denoted as $M \in G_2$, a set of attributes are denoted as A_{CT}, and a random number is denoted as $s \in Z_q$. Equation (3) process is applied to encrypt message.

$$CT = (A_{ct}, E = MY^s = e(g,g)^{ys}, (E_t = g^{t_i,s}\}_{\forall_{i \in A_U}}) \tag{3}$$

Decryption: Cipher text, private key, and attributes A_U user set are inputs of the decryption algorithm. If$(A_U \cap A_{CT}| \geq d$, chooses d attributes from $i \in A_U \cap A_{CT}$ to process values$e(g,g)^{q(i)s}$, $Y^s = e(g,g)^{q(0)s} = e(g,g)^{ys}$. Decrypted message is $M = E/Y^s$.

Access Trees are constructed for selected access structure based on identified private key. Tree each node is built as threshold gate and attributes are used for leaves. The "AND" and "OR" gates are represented by 2 of 2 and 1 of 2 threshold gates, respectively.

2.4 Access Tree Construction

Attribute-based encryption based on a set of binary trees for store the multi-dimensional data cubes. Set of access trees are build based on this technique and Attribute Based encryption is successfully applied to protect data privacy and stored in multi-dimensional object.

Model described using terms $f \in F$ is selected fact, $D_i \in D$ is model's dimension, and set of fact-dimensional relations are $R_i \in R$ respective to dimensionD_i.

The fact-dimension relation is denoted as $r = (f,e) \in R_t$ contains fact f to construct set of attributes under access model. This set of attributes contains:

C – From the whole hypercube, this attribute allows user to read information and user doesn't have restrictions for this hyper cube;

A_{D_i}- This attribute is required, if a user is able to receive any information stored in dimension D_i.

$A_{(D_i,C_j)}$ - This attribute enables read from specific dimension category. For instance, time is one dimension in cube. Some user has access to 'Year' and not for 'Day' category.

$A\_(D_i, C_j, f)$–This attribute not only restrict ability to read information from specific category and also set restrictions for selected fact value. Consider C is cube with healthcare data and cube store data related to patients, diagnoses and uses 'Patient' as fact type. This attribute is used to restrict access not only for selected diagnosis and also for specific patient. Patient and doctor are able to read information from the designated cell.

Every element $r = (f,e) \in R_i$ is mapped to attributes access set$\{A_{D_i}, A_{\left(D_i,C_j\right)}, A_{D_i,C_j,f}\}$, where dimension D_j linked withR_i, category from dimension D_j such that $e \in C_j$ and fact f binds to dimension using r.

Every dimension is performed with operation containing fact and receive a set of access attributes in equation (4).

$$A = \{C\} \cup U_1^n \{A_{D_i}, A_{D_i,C_j}, A_{D_i,G_j,f}\} \tag{4}$$

Access tree is constructing using this set for fact f in our model. The access tree is constructed using threshold gates nodes and attribute access with leafs. The "OR" and "AND" gates are used for construction.

This tree vertex set is described in equation (5).

$$V = (R, Dim\} \cup \{D_i\} \cup A \tag{5}$$

Model dimension produce $(Dim,\ D_i)$ edge and fact f of every record r in R_t produce three edges: (D_i, A_{D_i}), $\left(D_i, A_{D_i,C_j}\right)$ and$\left(D_i, A_{D_i,C_j,f}\right)$. Tree edge set for fact f produce dimension $D_i$of multi-dimensional structure is a set in equation (6).

$$E_{D_i} = \left\{(Dim, D_i), (D_i, A_{D_i}), \left(D_i, A_{D_i,C_j}\right), \left(D_i, A_{D_i,C_j,f}\right)\right\} \tag{6}$$

Tree edge set is described in equation (7).

$$E = \{(R,C), (R, Dim)\} \cup \textstyle\bigcup_1^n E_{D_i} \tag{7}$$

Mapping of object MO of multi-dimensional model to set of vertices and edges for fact$f : \varphi : \varphi(MO) \mapsto \left(V_f\right\}, \psi : \psi(MO) \longmapsto \{E_f\}$, where V_f and E_f are defined in equations (5, 6, 7), respectively. This technique easily convert access tree set to multi-dimensional objects set and apply attribute-based encryption to ensure data privacy that stored in multi-dimension database.

3 Results

The proposed dynamic attribute tree encryption method is applied for the Third Party Auditing to improve the trust among the users. The proposed method is applied in the simulated environment and compared with existing methods. The Simulation tool used for simulation is NS3. This is a distributed as source code, which means that the target system needs to have a software development environment to build the libraries first and then build the user's programs. NS-3 could be in principle distributed as pre-built libraries for selected systems.

Figure 2 shows the verification time in milliseconds for the proposed dynamic attribute tree method and the existing methods like Collaborative, Hash tree and Merkel Hash tree methods, the simulated results with number of blocks in the X axis and the verification times in the Y axis. For the 100 blocks of data that had been considered is in blocks of 10.The verification time of proposed dynamic attribute tree method has taken 207ms compared to Merkel hash tree which has a verification time of 273ms. The verification times of hash tree and collaborative method methods are 281 ms and 423 ms respectively. We observe that in verification time performance of the proposed method gives 31.8% efficiency for the Merkel and 35.7% efficiency for hash tree method. Bilinear mapping in the proposed method helps to protect the data in Third Party Audit and encrypt the data based on attributes. Attributes helps to provide effective update of the tree related to user data.

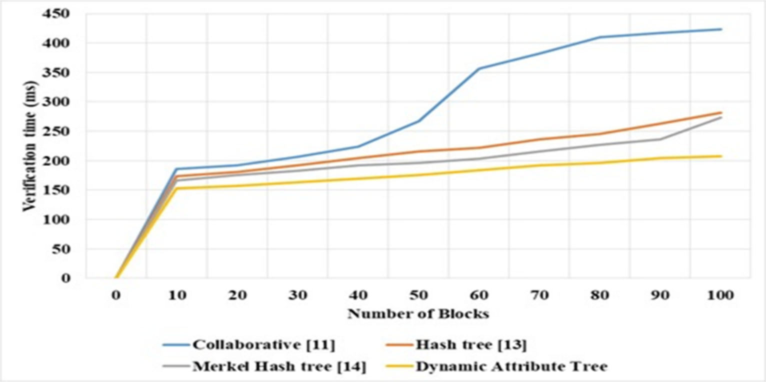

Fig 2. Verification time of the proposed Dynamic Attribute Tree method

Figure 3 shows the Challenge time in milliseconds for the proposed dynamic attribute tree method and the existing methods like Collaborative, Hash tree and Merkel Hash tree methods, the simulated results with number of blocks in the X axis and the Challenge times in the Y axis For the 100 blocks of data are considered in blocks of 10. The Challenge time of proposed dynamic attribute tree method has taken 293ms compared to Merkel hash tree which has a challenge time of 305ms. The challenge times of hash tree and collaborative method methods are 305 ms and 341 ms respectively. We observe that in challenge

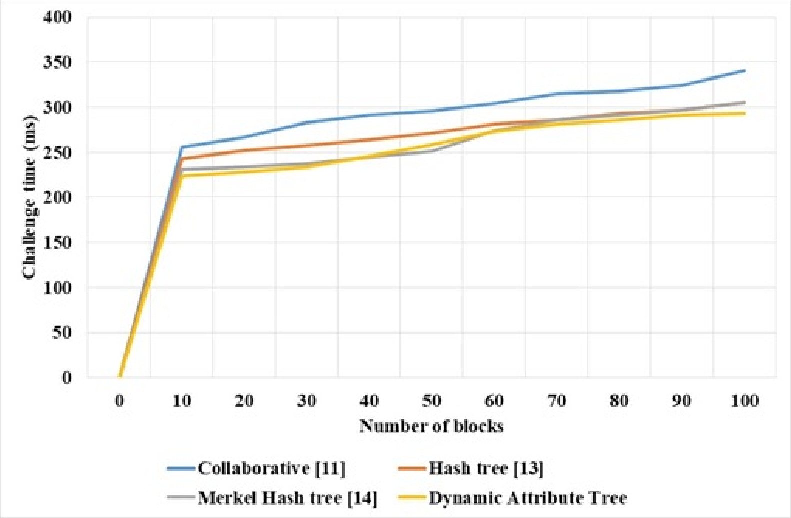

Fig 3. Challenge time of proposed and existing method in Third Party Auditing

time performance of the proposed method gives 4% efficiency for the Merkel and 16.3% efficiency for Collaborative method. The proposed Dynamic Attribute Tree method has advantage of storing the key in structure manner and dynamic update is effectively supported in this process.

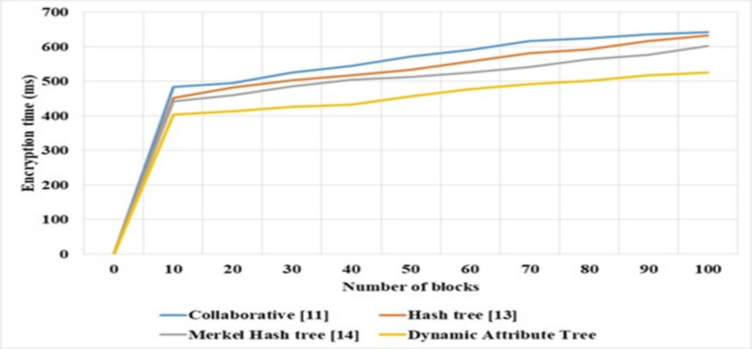

Fig 4. Encryption time of proposed and existing method in Third Party Auditing

Figure 4 shows the Encryption time in milliseconds for the proposed dynamic attribute tree method and the existing methods like Collaborative, Hash tree and Merkel Hash tree methods, the simulated results with number of blocks in the X axis and the Challenge times in the Y axis For the 100 blocks of data are considered in blocks of 10. The Encryption time of proposed dynamic attribute tree method has taken 526ms compared to Merkel hash tree which has a verification time of 602ms. The Encryption times of hash tree and collaborative method methods are 632 ms and 642 ms respectively. We observe that in Encryption time performance of the proposed method gives 14.4% efficiency for the Merkel and 20.1% efficiency for hash tree method. The key handling is carried out based on the tree structure and performs the dynamic update to encrypt and decrypt the data.

Figure 5 shows the Computation time in milliseconds for the proposed dynamic attribute tree method and the existing methods like Collaborative, Hash tree and Merkel Hash tree methods, the simulated results with number of blocks in the X axis and the Challenge times in the Y axis For the 100 blocks of data are considered in blocks of 10 The Computation time of proposed dynamic attribute tree method has taken 6.7ms compared to Merkel hash tree which has a Computation time of 7.5ms. The Computation times of hash tree and collaborative method methods are 8.3 ms and 9.5 ms respectively. We observe that in Encryption time performance of the proposed method gives 11.94% efficiency for the Merkel and 23.88% efficiency for hash tree method. The Dynamic Attribute Tree method has the superiority of encrypting data based on qualities and storing keys in a tree-like structure.

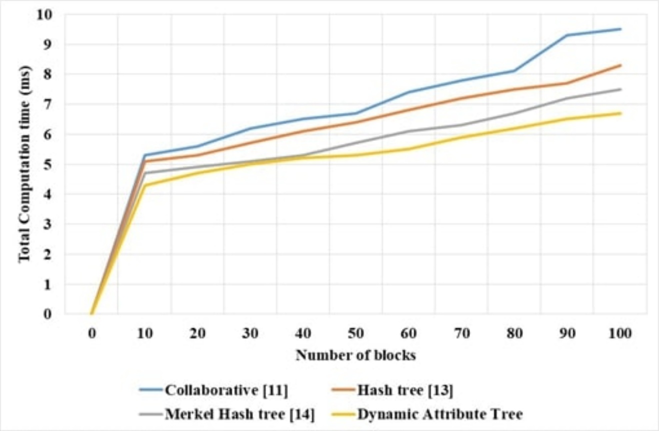

Fig 5. Total computation time for the proposed and existing method in Third Party Auditing

4 Conclusion

In this research, the Dynamic Attribute Tree method is proposed to improve the performance and user trust in Third Party Auditing. The proposed Dynamic Attribute Tree method has 6.7ms computation time for 100 blocks of data and the existing method Merkel Tree method has 7.5ms computation time. This proposed method helps to access the data in the structure format and update the data dynamically in the cloud by taking less time compare to the Existing methods in Third Party Auditing. The Dynamic Attribute Tree method has the superiority of encrypting data based on qualities and storing keys in a tree-like structure. Simulation is carried out to test the performance of the proposed and existing methods in cloud. Future work of the developed method involves in applying the optimization method to improve the efficiency for large dataset.

References

1) Wu J, Li Y, Wang T, Ding Y. CPDA: A Confidentiality-Preserving Deduplication Cloud Storage With Public Cloud Auditing. *IEEE Access*. 2019;7:160482–160497. Available from: https://dx.doi.org/10.1109/access.2019.2950750.
2) Li J, Wu J, Jiang G, Srikanthan T. Blockchain-based public auditing for big data in cloud storage. *Information Processing & Management*. 2020;57(6):102382–102382. Available from: https://dx.doi.org/10.1016/j.ipm.2020.102382.
3) Li X, Liu S, Lu R. Comments on "A Public Auditing Protocol With Novel Dynamic Structure for Cloud Data". *IEEE Transactions on Information Forensics and Security*. 2020;15:2881–2883. Available from: https://dx.doi.org/10.1109/tifs.2020.2978592.
4) Mishachandar B, Vairamuthu S, Pavithra M. A data security and integrity framework using third-party cloud auditing. *International Journal of Information Technology*. 2021;13(5):2081–2089. Available from: https://dx.doi.org/10.1007/s41870-021-00738-3.
5) Yang Z, Wang W, Huang Y, Li X. Privacy-Preserving Public Auditing Scheme for Data Confidentiality and Accountability in Cloud Storage. *Chinese Journal of Electronics*. 2019;28(1):179–187. Available from: https://dx.doi.org/10.1049/cje.2018.02.017.
6) Huang P, Fan K, Yang H, Zhang K, Li H, Yang Y. A Collaborative Auditing Blockchain for Trustworthy Data Integrity in Cloud Storage System. *IEEE Access*. 2020;8:94780–94794. Available from: https://dx.doi.org/10.1109/access.2020.2993606.
7) Naveena P, Soundarya R, S2, Mohanapriya KA. Light-weight secure auditing scheme for shared data in cloud storage. *IJERT*. 2020. Available from: https://www.ijert.org/research/light-weight-secure-auditing-scheme-for-shared-data-in-cloud-storageIJERTCONV8IS12001.pdf.
8) Daniel E, Vasanthi NA. LDAP: a lightweight deduplication and auditing protocol for secure data storage in cloud environment. *Cluster Computing*. 2019;22(S1):1247–1258. doi:10.1007/s10586-017-1382-6.
9) Frej MBH, Dichter J, Gupta N. Lightweight Accountable Privacy-Preserving Protocol Allowing the Cloud Client to Audit the Third-Party Auditor for Malicious Activities. *Applied Sciences*. 2019;9(15):3034–3034. Available from: https://dx.doi.org/10.3390/app9153034.
10) Xu Y, Zhang C, Wang G, Qin Z, Zeng Q. A Blockchain-Enabled Deduplicatable Data Auditing Mechanism for Network Storage Services. *IEEE Transactions on Emerging Topics in Computing*. 2021;9(3):1421–1432. Available from: https://dx.doi.org/10.1109/tetc.2020.3005610.
11) Huang P, Fan K, Yang H, Zhang K, Li H, Yang Y. A Collaborative Auditing Blockchain for Trustworthy Data Integrity in Cloud Storage System. *IEEE Access*. 2020;8:94780–94794. Available from: https://dx.doi.org/10.1109/access.2020.2993606.
12) Mishra R, Ramesh D, Edla DR. Dynamic large branching hash tree based secure and efficient dynamic auditing protocol for cloud environment. *Cluster Computing*. 2021;24(2):1361–1379. Available from: https://dx.doi.org/10.1007/s10586-020-03193-0.
13) Luo W, Ma W, Gao J. MHB*T based dynamic data integrity auditing in cloud storage. *Cluster Computing*. 2021;24(3):2115–2132. Available from: https://dx.doi.org/10.1007/s10586-021-03248-w.
14) Wang H, Qin H, Zhao M, Wei X, Shen H, Susilo W. Blockchain-based fair payment smart contract for public cloud storage auditing. *Information Sciences*. 2020;519:348–362. Available from: https://dx.doi.org/10.1016/j.ins.2020.01.051.
15) Deng L, Yang B, Wang X. A Lightweight Identity-Based Remote Data Auditing Scheme for Cloud Storage. *IEEE Access*. 2020;8:206396–206405. Available from: https://dx.doi.org/10.1109/access.2020.3037696.

16) Wang T, Mei Y, Liu X, Wang J, Dai HN, Wang Z. Edge-based auditing method for data security in resource-constrained Internet of Things. *Journal of Systems Architecture*. 2021;114:101971–101971. Available from: https://dx.doi.org/10.1016/j.sysarc.2020.101971.

17) Yu H, Lu X, Pan Z. An Authorized Public Auditing Scheme for Dynamic Big Data Storage in Cloud Computing. *IEEE Access*. 2020;8:151465–151473. Available from: https://dx.doi.org/10.1109/access.2020.3016760.

18) Kalluri RK, Guru CV. An effective analytics of third party auditing and Trust architectures for integrity in cloud environment. *Materials Today: Proceedings*. 2021. Available from: https://dx.doi.org/10.1016/j.matpr.2021.03.312.

19) Li J, Zhang Y, Ning J, Huang X, Poh GS, Wang D. Attribute Based Encryption with Privacy Protection and Accountability for CloudIoT. *IEEE Transactions on Cloud Computing*. 2020;p. 1–1. Available from: https://dx.doi.org/10.1109/tcc.2020.2975184.