DATA INTEGRITY AUDIT BASED ON DATA BLINDING FOR CLOUD AND FOR ENVIRONMENT

Salman Bin Salmin Bahakam[1], Md. Ateeq Ur Rahman[2], Subramanian K.M[3]
[1]*PG Scholar, Department of Information Technology, Shadan College of Engineering and Technology, Hyderabad, Telangana, India - 500086*
Email: fjsalman2@gmail.com
[2]*Professor, Department of Computer Science and Engineering, Shadan College of Engineering and Technology, Hyderabad, Telangana, India – 500086.*
Email: mail_to_ateeq@yahoo.com
[3]*Professor, Department of Computer Science and Engineering, Shadan College of Engineering and Technology, Hyderabad, Telangana, India - 500086*
Email: kmsubbu.phd@gmail.com

ABSTRACT

A cutting-edge computing architecture called cloud-fog computing enhances the capabilities of cloud computing, which offers numerous services using fog nodes. Low data security, delayed data processing, and ineffective communication are problems with traditional data integrity audits. This study suggests a data integrity audit scheme based on data blinding to address these issues. In order to reduce transmission delay, this technique creates a fog computing layer between the cloud service provider and the data owner using the edge devices in the transmission node. To reduce transmission time, the fog nodes' weight and subordinate distribution relationship dynamically assign the best path and send the data. In addition, the integrity check in the evidence generating process now includes a blind aspect to prevent data leaking. Based on computational Diffie-Hellman (CDH) assumptions, this study provides a security model and security proof. According to the experimental findings, the data integrity audit procedure now includes a blind component and a fog computing layer, which may significantly shorten data transmission lag times while also enhancing data audit security.

1. INTRODUCTION

The storage and processing demands on smartphones, laptops, and other terminal devices have surged recently as a result of the amount of information. Some users store their data on the cloud to ease the storage burden on terminal devices. To lessen server load, some cloud service providers may choose to delete some rarely used data. Cloud data loss happens when data that has been deleted cannot be recovered. The cloud server instead of the local device stores the data that users upload. It has become necessary to remotely verify the accuracy of the data that users upload.

The idea of Remote Data Possession Checking (RDPC), which incorporates proven data processing (PDP) and proof of retrievability (POR), is put forth as a solution to the aforementioned issues. However, it can be divided into private and public audits from the standpoint of data audit. The data owner serves as the auditor for private audits, but any approved third party audit can serve as the auditor for public audits. The majority of them will opt for public auditing because of its greater flexibility.

As more people have access to the internet in their daily lives, cloud computing is becoming more and more popular with people of all backgrounds. More consumers are storing their data in the cloud for convenient access from any location at any time. The typical cloud storage architecture, on the other hand, requires the cloud service provider to connect to each user, which subtly raises the workload burden on the cloud service provider. Therefore, it has become urgently necessary to find a solution for how to lessen the computing and load burden on cloud service providers.

Cloud servers are typically located distant from the user end when it comes to data integrity assessments. transfer delays would grow and network bandwidth would be used up by long-distance data transfer. Fog computing is a suggested solution to this issue. The idea of cloud computing is expanded by fog computing. It is more in close proximity to the data owner than cloud computing. The fog node layer is added to data transmission to decrease bandwidth and delay. Hu et al. presented a security and privacy protection system based on the fog computing framework, but they neglected to take the fog computing framework's data transmission model into account. The document label aggregation approach used during the evidence generation step of Yan et al.'s proposed remote data ownership audit the information and coefficients of the contained files are referred to by the scheme. By repeatedly requesting file labels, malicious attackers can use the revealed coefficients to calculate information, causing information leaking.

B. MOTIVATION AND CONTRIBUTION

This study gives a data transmission model in the cloud and fog network while also proposing a data integrity audit scheme based on the cloud and fog architecture. In

this concept, fog nodes broadcast data while also calculating the best communication channel to minimize communication overhead. In order to increase the security of the integrity audit and stop the adversary from calculating the ciphertext in the two interrogations, a blind factor is simultaneously added to the evidence production stage of the integrity audit.

The following are this paper's significant contributions.

1) In a cloud and fog environment, this study suggests a data integrity audit methodology that can efficiently lower communication costs associated with data transfer and ease the burden on cloud service providers' computing resources.

2) To prevent data leaking brought on by repeated submissions of nefarious auditors while challenging data, a blind element is added to the data integrity audit.

3) This article demonstrated the security of this technique using the provided security model. Experimental findings demonstrate the superior performance and viability of this approach.

Organization: The rest of the paper is organized as follows. In section 2, we review the related work which deals and the application security. We present the proposed system in section 3. We test the proposed system method in section 4. We analysis the system result in section 5. We conclude the paper in section 6.

The second section introduces the preliminary work of our proposed scheme. The third section defines the specific structure of the data blinding for cloud and fog (DBCF) system model and main steps. Section IV displays the safety analysis of DBCF. In the fifth section, the paper presents performance analysis, which includes theoretical complexity analysis and experimental performance. Section VI concludes the article.

2. RELATED WORK

A PDP approach was put forth by Ateniese et al. in 2007 [15] and allows a client saving data on an untrusted server to confirm that the server actually contains the original data. After that, Juels et al. [16] established a POR model that can produce succinct proof that the user can get the target file by backing up or archiving huge files and enables the user to restore the complete file's data. In 2008, Ateniese [17] developed a provably secure PDP system that allows block modification, deletion, and append operations and is fully based on symmetric key encryption. In addition to the first retrievability proof system that allows anybody, not only the file owner, to serve as a verifier, Shacham and Waters [18] also introduced a technique that exclusively allows private verification. PDP) when the client cannot perform remote data possession inspection. Yan et al. [14] propose a new RDPC scheme with a designated validator, in which the data owner designates a unique validator to check data integrity.

In 2014, Cisco presented the idea of fog computing. In this approach, devices at the network's edge are where data and its processing are focused. In order to guarantee data integrity, Mohammed et al. [21] later presented an authentication protocol for the fog computing environment. A novel chaotic map picture secret writing formula was put forth by Alzubi et al. [22] that used the security of improving the metric of cryptosystems to permutations at the pixel and bit levels. With the user holding the private key separate in the fog center, Tian and Wang [23] suggested a data audit method based on the Internet of Things (IoT) and cloud-fog computing. The two-time signature method was then put forth, which separates the signature process into the original signature and the final signature. A secure data query architecture for cloud and fog computing was introduced by Gu [24]. Cloud services are utilized to verify the query data from the fog network before it is sent to consumers Alzubi et al. [26] created a strong cryptosystem based on Hermite curves that is more suited for Internet of Things (IoT) devices with constrained processor and storage capabilities. The Hashed Needham Schroeder Cost Optimized Deep Machine Learning (HNS-CODML) approach, which increases the security of data transported from the cloud, was proposed by Alzubi et al. [27] in the same year. Noura et al.'s [28] new encryption system, which offers data confidentiality, integrity and availability, as well as source authentication, was proposed to safeguard data in fog computing.

N. Wu, B. Yin, W. Jia, and K. Gu, 2020. Fog computing is mostly utilized to analyze massive amounts of data generated by terminal devices. Fog nodes are the closest acquirers to the terminal devices, therefore while the processed data is being transported or aggregated, some malevolent nodes may tamper with it or illegally grab it. When real-time processes with high security are necessary for particular applications, cloud services may sample some data from fog services to verify final outcomes. We suggest a secure data query framework for cloud and fog computing in this study. When the fog network supplies users with requested data, we use a cloud service to verify the requested data. The framework allows fog networks to acquire related data from fog nodes in accordance with one of the pre-designated data aggregation trees after receiving some data aggregation topology trees from cloud servers. Some fog nodes are also designated as sampled nodes so they can provide relevant data back to the cloud server. We assess the security of our suggested framework in light of the security needs for fog computing. Our architecture efficiently guards data against man-in-the-middle assault, single node attack, and collusion attack by malevolent users while also ensuring the dependability of the necessary data. The experiments

also demonstrate the effectiveness and efficiency of our approach.

In many other industrial domains, including the Internet of Things (IoT), cloud computing has emerged as a crucial application service due to the rapid development of networks. IoT, however, is becoming more and more prevalent, which could result in daily production of vast and varied amounts of data. Therefore, it is challenging for the cloud computing paradigm to satisfy IoT's requirements for quick responses, high mobility, global spread, location awareness, and other factors. A novel computing idea, known as fog computing, was put forth by the Cisco Corporation. It transfers cloud computing's computation, storage, and other operations to the network's edge, where they are all more accessible to terminal users.

3. METHODOLOGIES

A. NOTATIONS

Let k be a safety parameter, and q is a large prime number which's order is k. G_1 and G_2 are multiplicative cyclic groups, and their order is k. g is the generator of G_1, and u is a random element of the multiplicative cyclic group. e is the bilinear mapping $G_1 \times G_1 \rightarrow G_2$, H is a secure

TABLE 1. Frequently used notations.

Notations	Description
k	a security parameter
q	a large prime
$\mathbb{G}_1, \mathbb{G}_2$	the multiplicative cyclic groups of order q
g	a generator of multiplicative cyclic group $\mathbb{G}_1$
u	a random group element of G_1
H	a secure hash function $\{0,1\}^* \rightarrow \mathbb{G}_1$
x, r	$x, r \in Z_q^*$
e	$\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$
ϕ	$Z_q^* \times \{1, \cdots, n\} \rightarrow \{1, \cdots, n\}$
φ	$Z_q^* \times Z_q^* \rightarrow Z_q^*$
fn_m	a fog node device
m	the number of fog node devices
$D = (V, E)$	the undirected graph with vertex set V and edge set E
w_{fn_i, fn_j}	the delay between nodes $\{fn_i, fn_j\}$
com_{fn_i, fn_j}	the communication delay between nodes $\{fn_i, fn_j\}$
pro_{fn_i, fn_j}	the processing delay between nodes $\{fn_i, fn_j\}$
que_{fn_i, fn_j}	the queuing delay between nodes $\{fn_i, fn_j\}$
$tran_{fn_i}$	the transmission speed of each fog node device fn_i
z_i	the divided sub-transmission data
$dist_{fn_i, fn_j}$	the relationship between $\{fn_i, fn_j\}$

hash functions, and φ, ϕ are pseudo-random permutation and pseudo-random function. Besides, some frequently used notations are given in Table 1.

B. BILINEAR MAPS

Specify that the multiplicative cyclic groups G1 and G_2 have the same prime order q, g is a generator of G_1. e is the mapping of $G_1 \times G_1 \rightarrow G_2$, which has the following properties:

1) Bilinearity: for ∀u, v ∈ G_1, ∀a, b ∈ Zq∗, there is an equation $e(u^a, v^b) = e(u, v)^{ab}$.

2) Non-Degeneracy: ∃u, v ∈ G_1 such that e(u, v) , here 1_{G2} represents the identity element of the G_2 group.

3) Computability: for ∀u, v ∈ G_1, there is an algorithm that calculates the mapping e(u, v).

C. CDH ASSUMPTION

Numerous cryptographic techniques, including public-key encryption, digital signatures, and authentication key exchange, are built using the CDH assumption, which is a widely accepted cryptographic supposition [29]. Furthermore, this premise is used in sophisticated agreements like those for cloud storage that refuse authentication.

Specifically, the CDH assumption on a cyclic group G with generator g refers to that it is hard to compute g^{ab} for any polynomial-time adversary A when given the items g, g^a, and g^b, which can be defined as:

$$ADV^{CDH}_{\mathbb{G}_1,\mathcal{A}} = Pr[\mathcal{A}(g, g^a, g^b) = g^{ab} : a, b \overset{R}{\leftarrow} Z_q^*] \leq \varepsilon \quad (1)$$

D. SYSTEM MODEL

Four entities make up the data integrity audit model based on data blinding in the cloud and fog environment: data owners, fog computing nodes, cloud service providers, and fog computing nodes,

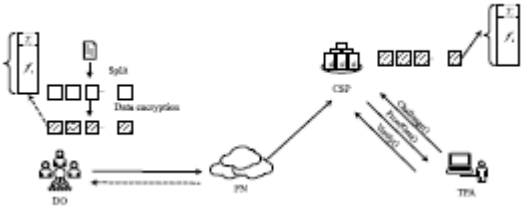

Fig. 1. DCBF system model.

and independent auditors. The system model for our suggested DCBF model is shown in Figure 1.

1) To achieve the goal of storing data remotely and having access to it at any time, the Data Owner (DO) rents cloud storage services and uploads a significant volume of data to the cloud storage server. The consumer of the data can be an individual or an organization.

2) Gateways, switches, and routers are examples of networked edge devices with precise computing capabilities known as fog computing nodes (FN). This model lessens the workload of the cloud service provider by preprocessing the data before it is communicated through the fog computing node.

3) Cloud service providers (CSP) have substantial computational power and enormous storage capacity. Cloud service providers receive user-uploaded data through fog nodes, offer cloud computing and storage to data owners, and send data integrity certificates to outside auditors after receiving data challenges. The cloud service provider splits users into blocks and keeps data that has been labeled. They only need to aggregate and generate proofs through tags when they are required [30].

4) For the benefit of the data owner, the third-party auditor (TPA) will examine the accuracy of the outsourced data. And both the cloud storage server and the data owner have confidence in TPA. In the subsequent data integrity audit procedure, the third-party

auditor will communicate the audit findings to the data owner.

The DBCF model includes the five polynomial time algorithms.

1) **Setup (1^k) → (sk, pk):** This algorithm is used to initialize the system and generates the user's public and private key pair. Enters the security parameter k, and output the corresponding public key and private key.
2) **TagGen (F, x) → T:** The data owner executes this algorithm to generate the tag set of the uploaded file, and the data owner uploads the tag set and data block to the cloud accordingly.
3) **Challenge (cb) → chal:** This algorithm is executed by a third-party auditor, inputs the number of blocks to be challenged, and outputs challenge information to the cloud service provider.
4) **ProofGen (F, T, chal) → P:** This algorithm is executed by the cloud service provider and generates evidence. According to the challenge information, read the files stored in the cloud and the corresponding tag information to calculate the evidence and return it to the third-party auditor.
5) **Verify (X, chal, P) → {0, 1}**: The third-party auditor executes this algorithm and judges whether the data is entirely based on the evidence returned by the cloud service provider. If it is completed, outputs 1 to indicate.

E. SECURITY MODEL

In this subsection, the security model of DBCF is defined. This scheme is characterized by indistinguishability under chosen-plaintext attack (IND-CPA) game plaintext attack in the random oracle model [31]. The specific steps are as follows.

1) Initialization: Challenger B generates the system environment and initializes public parameters, and the adversary (denoted as A) obtains these parameters.

2) Query: The adversary A can make the following query in the bounded order of the polynomial.

a) H-Query: Challenger B establishes a hash query table to record and answer the adversary's hash query.

b) Tag-Query: Adversary A submits file information to challenger B, and the challenger runs the following formula and returns the result to adversary A.

TagGen(F, x) → T

c) Verify-Query: The audit query is based on the tag query in the previous step. Challenger B runs Challenge(cb)→ chal and sends the challenge block information chal to adversary. The adversary A calculates the evidence P by running ProofGen(F, T , chal) → P. Then, the adversary A returns the result. Challenger A calculates Verify(X, chal, P) → {0, 1} after receiving evidence P, and the final result will be returned to adversary A.

3) Final phase: At this stage, challenger B submits challenge information chal∗ to adversary A, then adversary A returns evidence P∗.

If Verify(X, chal, P) → 1, the following conditions hold.

1) If the challenge information chal/chal∗ is submitted, the challenge file block has previously calculated the tag T.

2) The returned evidence P∗ is not equal to P, and P∗ will be calculated by ProofGen(F, T , chal∗) → P∗.

4. OUR PROPOSED DBCF MODEL

A. CLOUD AND FOG COMPUTING MODEL

A cloud service layer and a fog computing layer can be parts of the DBCF model's cloud and fog computing concept. The network structure of the fog computing layer's m fog node devices (fn_1, fn_2, m, and fn_m) is seen in Figure 2.

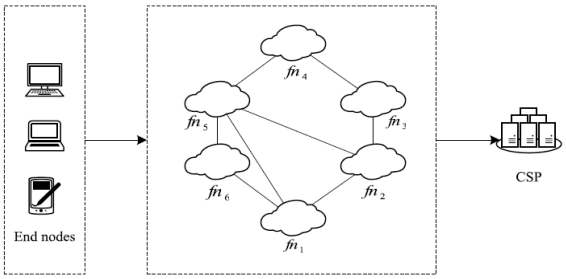

Fig 2: Network Structure.

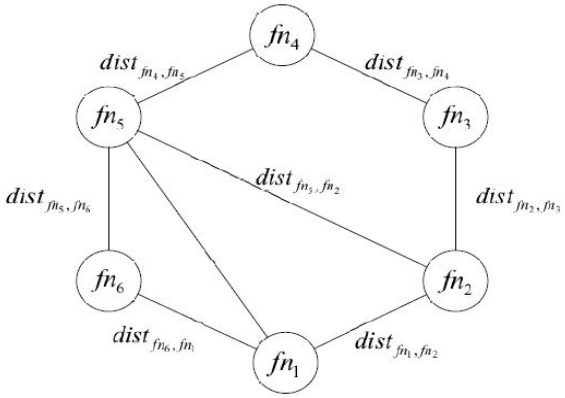

Fig 3: Weighted undirected graph.

According to the above figure, it can be abstracted as a weighted undirected graph D = (V, E), V is a set of vertices in the graph D, representing the fog node device, and E is a set of edges represents the communication link between nodes. w_{fni}, fn_j represents the delay between nodes {fn_i, fn_j}, including communication delay, processing delay and queuing delay. The weighted undirected graph is shown in Figure 3.

Assuming that the transmission speed of each fog node device fn_i is $tran_{fni}$, during the data transmission process, the data owner divides the transmission data Z into $z_i = \lambda_i Z$, and z_i represents the divided sub-transmission data. The transmission time of the entire transmission data Z at the fog computing layer can be expressed as:

$$t(\lambda_i) = \max\left\{\frac{\lambda_i Z}{tran_{fn_i}} + w_{fn_i,fn_j} dist_{fn_i,fn_j}\right\} \quad (2)$$

$$w_{fn_i,fn_j} = com_{fn_i,fn_j} + pro_{fn_i,fn_j} + que_{fn_i,fn_j} \quad (3)$$

Among them, $\lambda_i Z/tran_{fni}$ indicates the time for the fog node to process the subtask z_i, $w_{fni,fnj}$ $dist_{fni,fnj}$ indicates the delay between $\{fn_i, fn_j\}$, $dist_{fni,fnj}$ indicates whether there is a subordinate allocation relationship between $\{fn_i, fn_j\}$, and $dist_{fni,fnj} = 1$ indicates an allocation relationship existing, and vice versa.

Since the total transmission time in the fog calculation is equal to the most extensive transmission delay among all transmission times, in order to achieve the minimum delay, a set of optimal λ_i is required to minimize the objective function. The fog node calculation optimization model can be established as follows:

$$\min\left\{\max\left[\frac{\lambda_i Z}{tran_{fn_i}} + w_{fn_i,fn_j} dist_{fn_i,fn_j}\right]\right\}, i,j \in [1,m]$$

$$s.t.\ dist_{fn_i,fn_j} = \begin{cases} 1, & \lambda_i \neq 0 \\ 0, & \lambda_i = 0 \end{cases}, \sum_{i=1}^{m} \lambda_i = 1 \quad (4)$$

The task processed on each fog node is $z_i = \lambda_i Z$, then the task to be processed on each node can be constructed into a m dimensional vector $z = [z_1, z_2, \cdots, z_m]^T$. Then the total time from node fn_r to transmit data Z` at the fog computing layer can be expressed as:

$$t(\mathbf{z}) = \max\left\{\frac{z_1}{tran_{fn_1}} + w_{fn_r,fn_1} dist_{fn_r,fn_1}, \cdots, \frac{z_m}{tran_{fn_m}} + w_{fn_r,fn_m} dist_{fn_r,fn_m}\right\} \quad (5)$$

Therefore, in the search space $\Gamma = \prod_{i=1}^{m}[Z_{min}, Z_{max}] = \prod_{i=1}^{m}[0, Z]$, Z_{min} and Z_{max} and Z^{max} represent the maximum and minimum values that the subtask z_i can take. Solving the corresponding transmission task z_i of each node on the fog node can be transformed into the following optimization problem:

$$\mathbf{z} = \arg\min_{\mathbf{z}\in\Gamma}\{t(\mathbf{z})\}$$

$$s.t.\ z_i \geq 0, \sum_{i=1}^{m} z_i = Z \quad (6)$$

B. MAIN STEPS OF INTEGRITY AUDIT

A data integrity audit approach based on data blinding is provided in this section. The original data cannot be known by anybody other than the data owner thanks to this paradigm. First, choose at random a big prime integer q with order k, where q is the security parameter. Two multiplicative cyclic groups, G1 and G2, exist. The generator is g, the length of the groups is q, and the G1's random group element is u. G1*G1 G2 is a safe hash function, and e is a bilinear mapping of the two. and are a pseudo-random permutation and a pseudo-random function, respectively. (q, g, u, G1, G2, e, H,) are public parameters.

Setup(1^k) → (sk, pk): The data owner randomly selects a number x as the private key, where $x \in Z^*_q$. Calculate $X = g^x$, and the data owner publishes X as the public key.

TagGen(F; x) → T : First, before uploading the file F, the ndata owner divides the file F into n small pieces, denoted as F D ($f_1, f_2, \ldots ; f_n$). The data owner calculates the label T_i for each small file, and the calculation label equation is:

$$T_i = (H(F_{id}\|i) \cdot u^{f_i})^x \quad (7)$$

Among the equation, F_{id} represents a specific file identifier. Finally, the data owner calculates the tag set T of the le F, in which $T = (T_1, T_2, \ldots, T_n)$. Then, uploads the pairs $\{(T_i, f_i/i \in [i\ n])\}$ to the cloud service provider (CSP).

Challenge(cb) → chal: The third-party auditor randomly selects two numbers (k_1, k_2), where k_1, k_2 are the seeds of pseudo-random permutation and pseudo-random function. The third-party auditor sends the total challenge block count cb $\in$ [1. n] together with the pseudo-random seeds as a challenge to the CSP, where challenge denotes chal =(k_1, k_2, cb).

ProofGen(F, T, chal) → P: After receiving the challenge information, the cloud service provider calculates the indexs of challenge blocks according to k_1, the challenge blocks index $v_i = \emptyset(k_1, i)$. Then uses k_2 to calculate the random parameter $a_i = \emptyset\ (k_2, i)$, where $1 \leq i \leq cb$. At the same time, the cloud service provider randomly selects a number r, calculates $R = u^r$, publishes R and saves r as a blinding factor. Then, the cloud service provider calculates T and F as Follows:

$$\overline{T} = \prod_{i=1}^{cb} T_{v_i}^{a_i} \quad (8)$$

$$\overline{F} = \sum_{i=1}^{cb} a_i f_{v_i} + r \quad (9)$$

Finally, the CSP returns the proof $P = (\overline{F}, \overline{T})$ to the third-party auditor as a response to the challenge.

Verify(X, chal, P) → {0, 1}: After receiving the evidence named P, the third-party auditor checks the equation $e(\prod_{i=1}^{cb}(H(F_{id}\|i)^{a_i} \cdot u^{\overline{F}}, X) = e(\overline{T}, g) \cdot e(R, X)$. If it holds, it outputs 1 to indicate that the challenged data block information is complete, otherwise, it outputs 0.

If the cloud service provider complies with the rules of this agreement, it verifies the correctness of the data integrity equation as follows:

$$
\begin{aligned}
& e(\prod_{i=1}^{cb}(H(F_{id}\|i)^{a_i}\cdot u^{\overline{T}}, X) \\
&= e(\prod_{i=1}^{cb}(H(F_{id}\|i)^{a_i}\cdot u^{\sum_{i=1}^{cb} a_i f_{v_i}+r}, X) \\
&= e(\prod_{i=1}^{cb}(H(F_{id}\|i)^{a_i}\cdot (\prod_{i=1}^{cb} u^{a_i f_{v_i}})\cdot u^{r}, X) \\
&= e(\prod_{i=1}^{cb}((H(F_{id}\|i)\cdot u^{f_{v_i}})^{a_i}\cdot u^{r}, g^{x}) \\
&= e(\prod_{i=1}^{cb}((H(F_{id}\|i)\cdot u^{f_{v_i}})^{xa_i}\cdot (u^{r})^{x}, g) \\
&= e(\prod_{i=1}^{cb}(T_{v_i})^{a_i}\cdot (u^{r})^{x}, g) \\
&= e(\overline{T}\cdot (u^{r})^{x}, g) \\
&= e(\overline{T}, g)\cdot e(u^{r}, g^{x}) \\
&= e(\overline{T}, g)\cdot e(R, X) \qquad (10)
\end{aligned}
$$

5. RESULT

Fig 4. Admin Login Page

Fig 5. Admin Home Page

Fig 6: Data User Register

Fig 7. Cloud Server Page

6. CONCLUSION AND FUTURE ENHANCEMENT

In the cloud and fog environment, a DBCF technique is suggested in this study. In the situation of data integrity auditing, this protocol can guarantee data security. This method avoids the adversary's repeated requests for user information by introducing a blind factor into the data verification process and adding random values to each verification. The establishment of the fog computing layer coincides with the modification of the transmission network's design using the cloud and fog structures, which can significantly cut down on communication overhead. Additionally, the security model is presented and shown to be secure when using CDH's random oracle model.

Future Work: Future Work: Last but not least, the performance analysis demonstrates that this approach will be more effective in real-world settings. The architecture model for the fog computing layer can be enhanced in subsequent research to increase its effectiveness.

REFERENCES

1. W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge computing: Vision and challenges", IEEE Internet Things J., vol. 3, no. 5, pp. 637-646, Oct. 2016.
2. J. Li, Y. Zhang, X. Chen and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing", Compute. Secur., vol. 72, pp. 1-12, Jan. 2018.
3. Y. Deswarte, J.-J. Quisquater and A. Saidane, "Remote integrity checking", Proc. Working Conf. Integrity Internal Control Inf. Syst., pp. 1-11, 2003.
4. H. Wang, D. He, A. Fu, Q. Li and Q. Wang, "Provable data possession with outsourced data transfer", IEEE Trans. Services Compute., vol. 14, no. 6, pp. 1929-1939, Nov. 2021.
5. C. C. Erway, A. Kupçu and C. Papamanthou, "Dynamic provable data possession", ACM Trans. Inf. Syst. Secur., vol. 17, no. 4, pp. 1-29, 2009.
6. A. Acar, H. Aksu, A. S. Uluagac and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation", ACM Compute. Surv., vol. 51, no. 4, pp. 1-35, 2018.
7. H. Wang, L. Feng, Y. Ji, B. Shao and R. Xue, "Toward usable cloud storage auditing revisited", IEEE Syst. J., vol. 16, no. 1, pp. 693-700, Mar. 2022.
8. J. Chang, B. Shao, Y. Ji, M. Xu and R. Xue, "Secure network coding from secure proof of retrievability", Sci. China Inf. Sci., vol. 64, no. 12, Dec. 2021.
9. A. Alrawais, A. Alhothaily, C. Hu and X. Cheng, "Fog computing for the Internet of Things: Security and privacy issues", IEEE Internet Compute., vol. 21, no. 2, pp. 34-42, Mar./Apr. 2017.

10. F. Bonomi, R. Milito, P. Natarajan and J. Zhu, "Fog computing: A platform for Internet of Things and analytics" in Big Data and Internet of Things: A Roadmap for Smart Environments, Cham, Switzerland: Springer, pp. 169-186, 2014.
11. M. Ma, D. He, D. Kumar, K.-K. R. Choo and J. Chen, "Certificateless searchable public key encryption scheme for industrial Internet of Things", IEEE Trans. Ind. Informant., vol. 14, no. 2, pp. 759-767, May 2017.
12. J. Zhou, T. Wang, M. Z. A. Bhuiyan and A. Liu, "A hierarchic secure cloud storage scheme based on fog computing", Proc. IEEE 15th Int. Conf. Dependable Auton. Secure Compute. 15th Int. Conf. Pervasive Intell. Compute. 3rd Int. Conf. Big Data Intell. Compute. Cyber Sci. Technol. Congr. (DASC/PiCom/DataCom/CyberSciTech), pp. 470-477, Nov. 2017.
13. P. Hu, H. Ning, T. Qiu, H. Song, Y. Wang and X. Yao, "Security and privacy preservation scheme of face identification and resolution framework using fog computing in Internet of Things", IEEE Internet Things J., vol. 4, no. 5, pp. 1143-1155, Oct. 2017.
14. H. Yan, J. Li and Y. Zhang, "Remote data checking with a designated verifier in cloud storage", IEEE Syst. J., vol. 14, no. 2, pp. 1788-1797, Jun. 2020.
15. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, et al., "Provable data possession at untrusted stores", Proc. 14th ACM Conf. Comput. Commun. Secure. (CCS, pp. 598-609, 2007.
16. A. Juels and B. S. Kaliski, "Pors: Proofs of retrievability for large files", Proc. 14th ACM Conf. Compute. Commun. Secur. (CCS), pp. 584-597, 2007.
17. G. Ateniese, R. Di Pietro, L. V. Mancini and G. Tsudik, "Scalable and efficient provable data possession", Proc. 4th Int. Conf. Secure. Privacy Commun. Netw. (SecureComm), pp. 1-10, 2008.
18. H. Shacham and B. Waters, "Compact proofs of retrievability", J. Cryptol., vol. 26, no. 3, pp. 442-483, Jul. 2013.
19. H. Wang, "Proxy provable data possession in public clouds", IEEE Trans. Services Compute., vol. 6, no. 4, pp. 551-559, Oct./Dec. 2013.
20. Y. Ren, J. Xu, J. Wang and J.-U. Kim, "Designated-verifier provable data possession in public cloud storage", Int. J. Secur. Appl., vol. 7, no. 6, pp. 11-20, Nov. 2013.
21. KashifMunir and L. A. Mohammed, "Secure third party auditor (TPA) for ensuring data integrity in fog computing", Int. J. Netw. Secure. Appl., vol. 10, no. 6, pp. 13-24, Nov. 2018.
22. J. A. Alzubi, O. A. Alzubi, G. Suseendran and D. Akila, "A novel chaotic map encryption methodology for image cryptography and secret communication with steganography", Int. J. Recent Technol. Eng., vol. 8, no. 1C2, pp. 1122-1128, 2019.
23. J.-F. Tian and H.-N. Wang, "An efficient and secure data auditing scheme based on fog-to-cloud computing for Internet of Things scenarios", Int. J. Distrib. Sensor Netw., vol. 16, no. 5, pp. 1-15, 2020.
24. K. Gu, N. Wu, B. Yin and W. Jia, "Secure data query framework for cloud and fog computing", IEEE Trans. Netw. Service Manage., vol. 17, no. 1, pp. 332-345, Mar. 2020.
25. S. Xu, J. Ning, Y. Li, Y. Zhang, G. Xu, X. Huang, et al., "Match in my way: Fine-grained bilateral access control for secure cloud-fog computing", IEEE Trans. Dependable Secure Compute., vol. 19, no. 2, pp. 1064-1077, Mar./Apr. 2020.
26. O. A. Alzubi, J. A. Alzubi, O. Dorgham and M. Alsayyed, "Cryptosystem design based on Hermitian curves for IoT security", J. Supercomput., vol. 76, no. 11, pp. 8566-8589, Nov. 2020.
27. J. A. Alzubi, R. Manikandan, O. A. Alzubi, I. Qiqieh, R. Rahim, D. Gupta, et al., "Hashed needham schroeder industrial IoT based cost optimized deep secured data transmission in cloud", Measurement, vol. 150, Jan. 2020.
28. H. Noura, O. Salman, A. Chehab and R. Couturier, "Preserving data security in distributed fog computing", Ad Hoc Netw., vol. 94, Nov. 2019.
29. N. Döttling and S. Garg, "Identity-based encryption from the Diffie–Hellman assumption", J. ACM, vol. 68, no. 3, pp. 1-46, Mar. 2021.
30. J. Chang, H. Wang, F. Wang, A. Zhang and Y. Ji, "RKA security for identity-based signature scheme", IEEE Access, vol. 8, pp. 17833-17841, 2020.
31. Y. Liu, L. Y. Zhang, J. Wang, Y. Zhang and K.-W. Wong, "Chosen-plaintext attack of an image encryption scheme based on modified permutation–diffusion structure", Nonlinear Dyn., vol. 84, no. 4, pp. 2241-2250, 2016.
32. H. A. Al Hamid, S. M. M. Rahman, M. S. Hossain, A. Almogren and A. Alamri, "A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing-based cryptography", IEEE Access, vol. 5, pp. 22313-22328, 2017.