Privacy-aware enhanced homomorphic mechanism for group data sharing

Jayalakshmi Karemallaiah, Prabha Revaiah
Department of Computer Science and Engineering, Dr. Ambedkar Institute of Technology, Bangalore, India

Article Info

Article history:

Received Jan 31, 2024
Revised Aug 7, 2024
Accepted Aug 14, 2024

Keywords:

Cloud computing security
Data sharing
Homomorphic encryption
Privacy
Verifiable privacy-aware enhanced homomorphic

ABSTRACT

Cloud-based group data sharing has gained huge popularity in recent years. Accomplishing the efficacy and security of the data in a cloud-computing framework is challenging. Sharing data in a cloud environment is quite challenging and needs to be resolved. Furthermore, while exchanging data on the cloud, it is challenging to achieve both anonymity and traceability. The main aim of this research work is to make it easier for the same group to share and store anonymous data on the cloud securely and effectively. This research work presents verifiable privacy-aware enhanced homomorphic (VPEH) encryption for multiple participants; moreover, the enhanced homomorphic encryption mechanism provides end-to-end encryption and allows the secure computation of data without revealing any data in the cloud. The proposed algorithm uses homomorphic multiplication to compute the hashes product of challenges blocks that makes it more efficient Furthermore, an additional security model is incorporated to verify the shared data integrity. The VPEH mechanism is evaluated considering parameters such as tag generation, proof generation, and verification; model efficiency is proved by observing the marginal improvisation over the other existing model by varying the number of blocks and several challenge blocks.

Corresponding Author:

Jayalakshmi Karemallaiah
Department of Computer Science and Engineering, Dr. Ambedkar Institute of Technology
Bangalore, India
Email: jayalakshmi_112@rediffmail.com

1. INTRODUCTION

The data generated via each organization is distinct in its form, which entails the importance of each organization, it forms the basis of groundwork incorporating information, and knowledge, and eventually, this forms the basis of wisdom to take accurate decisions and activities. This might involve a range of activities, including treating viruses, boosting an organization's general growth and, therefore, revenues, building a well-organized structure, achieving goals, and subsequently improving performance [1]. Henceforth, data sharing, storage, and investigation are the key services essential for an organization to elevate its performance [2]. Although the data generated by each of these organizations in enormous amounts results in explosive development, pressure arises on storing these huge volumes of data locally [3], [4]. This is quite challenging as each passing day discovers the data for its limited resources. Several businesses have switched to the cloud framework for these services due to the many advantages it provides, including on-demand services, scalability, dependability, flexibility, quantifiable services, disaster recovery, accessibility, and many other significant advantages [5]. A platform called cloud-computing enables us to store enormous amounts of memory space and enormous amounts of processing capability at a good price. This becomes a reason for several users to obtain these services through various platforms despite the location and time consequently transferring widespread accessibility to the cloud users [6]. Cloud users can save by transferring data management systems into the cloud for storage purposes based on cloud services, and enhance the management of production to accomplish projects and establish collaborations [7]. Individuals and their collaborations are consequently migrating to the cloud platform to utilize its services [8]. With the expansion of cloud computing techniques, it will be difficult to facilitate the migration of all enterprises to cloud platforms shortly [9]. Figure 1 shows the general framework for sharing environment.

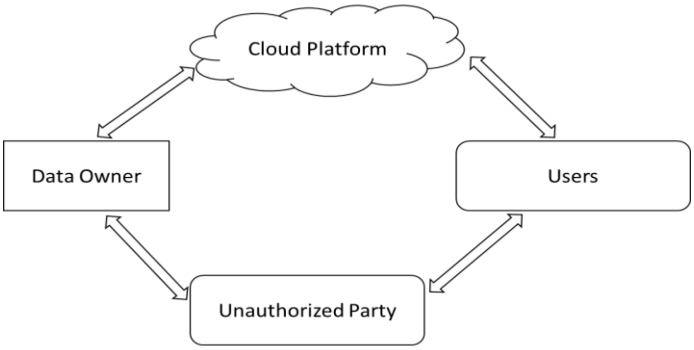


Figure 1. General framework for sharing environment

Figure 1 here depicts a shared environment, where the organization's crucial information needed to be shared on the cloud paradigm by the owners, because of the restricted storage and computational ability of the organizations and several other benefits provided by the cloud platform. Moreover, the data shared on the cloud is accessed by multiple users to accommodate various requirements based on its effectiveness. Nevertheless, the data may be leaked by the recipient upon receiving it. The data is leaked by the third party or the recipients; they may steal the data by unauthorized access and illegal means. The data that is disclosed or lost during the process may pose a serious threat to the confidentiality of the organization. The organization's reputation could be affected by the disclosure of confidential information, such as the value of its shareholders and its rank and standing [8]. The data of the enterprise is an essential asset, it is necessary to keep this asset confidential and secure. This leads to various solutions to preserve the integrity of data effectively in a shared environment.

Besides several features, provided by the cloud-computing paradigm, this results in various obstructions that may hinder its growth if not handled properly [10]. By implementing a viable solution, the organization permits its employees and departments to store and share data via the cloud platform. By utilizing the cloud platform, the organization may be relieved of the maintenance and storage responsibilities [11], [12]. Consequently, this includes a variety of hazards that result in a range of concerns for cloud consumers [13]. Outsourcing data through cloud servers indicates that the data is not under the control of the user, which may cause distress and lead to the loss of sensitive and valuable information. In an open-source environment, the server becomes vulnerable to attack due to the constant dissemination of shared data across the operation. In the worst-case scenario, the cloud server may disclose user information for unlawful purposes [14], [15]. To improve the performance of the business, the data had to be shared with various stakeholders, such as business partners, employees, and consumers, via the interior or exterior of the enterprise's environment. The receiving end mishandles the data and intentionally or unintentionally discloses sensitive information to unauthorized third parties [16], [17].

Data authentication has become a challenging mechanism in the fields of information security and cloud computing recently. There is a need for robust methods that can effectively find solutions to the existing problems, by prohibiting data leakage and detecting the malicious activity that causes data leakage, this emerging challenge can be overcome to a significant extent. To protect and authenticate the data on a cloud platform, numerous approaches have been developed. Even though several solutions are presented to address existing problems, there is always room for a comprehensive examination of the existing solutions pertinent to these applications. In the field of cloud computing, the significance of a greater comprehension of the current trends for securely sharing cloud-based data is discussed. This research designs and develops verifiable privacy-aware enhanced homomorphic (VPEH) for secure group data sharing in the cloud, further contribution of research work is as follows: i) VPEH is a privacy-aware enhanced homomorphic-based security framework in the cloud that aims to verify the users and data; ii) Developed enhanced homomorphic mechanism uses homomorphic multiplication to compute the hash product of challenged blocks to be more efficient; and iii) The additional security layer is developed for privacy, verification, and trust; VPEH is evaluated considering the different challenged blocks over the different parameters *i.e.* tag generation, proof generation, and verification.

This research is organized as follows: the first section of the research starts with the background of cloud computing, its security, and the importance of group data-sharing protocol. Further, this section concludes with the motivation and contribution of research work. The second section discusses various existing security protocols along with their shortcoming. The third section presents the framework of VPEH along with the algorithm and mathematical model, VPEH is evaluated in the fourth section of the research.

2. RELATED WORK

Security has been a major concern in the cloud since the development of computing models as they are vulnerable due to access from anywhere characteristics, moreover, there have been several aspects of security including confidentiality, integrity, and participants' privacy, this section of the research discusses some of the efficient model developed in recent past to conquer the security concern. In study [11], the confidentiality, integrity, and authenticity of data are proposed via cryptographic-based algorithms. To guarantee the integrity of the data, symmetric keys, and hash codes are utilized as a part of a cryptographic function. The validity and integrity of the data are guaranteed by the elliptic curve digital signature technique. In addition, the complex encryption standard-Galois counter mode and the powerful whirlpool hash function are integrated to offer authenticity and secrecy. Liang *et al.* [12] suggested to use a ciphertext-policy attribute-based proxy re-encryption system for the secure sharing of cloud data. Re-encryption is being improved, along with the key-generation stages that go with it, to lower computation costs and enhance communication. The data owner encrypts the data access rights given to others on a cloud-based platform. A file hierarchy attribute-based encryption approach for cloud-based data authentication and security is proposed by Sreedhara *et al.* [13].

Assuming decisional bilinear Diffie-Hellman (DBDH), this mechanism acts as an access structure to show the security of the file hierarchy ciphertext policy-based encryption (FH-CP-ABE) approach, which tends to successfully block some plaintext assaults. The results show that cipher text policy attribute-based encryption (CP-ABE) requires substantially more computational complexity and storage space for encryption than it does for decryption. The disadvantage of this approach is that the calculation cost increases considerably whenever a data owner wants to estimate common attributes and an integrated ciphertext just once. Liu *et al.* [14] provided a proposal for the fair regulation of cloud-based data access. The system employs a fair key generation mechanism to prevent unauthorized access to shared data, and none of the users relocated their shares. The recommended method for concealing the decryption key for shared data results in the creation of a large number of fictitious keys. A theoretical analysis of this method reveals that some shares are consistently offered by their respective users, allowing them to consistently recover the correct decryption key. The performance study also revealed that the authentication system was still ineffective despite reductions in transmission costs and computation time.

Liu *et al.* [15] proposed a CP-ABE strategy to reduce the computational cost of intensive decryption at the user end, which increases with the complexity of the access policy. This system allowed for the outsourcing of decryption, the revocation of attributes, and the updating of rules when user attributes changed. While the efficiency of the recommended approach has been rigorously evaluated in terms of processing and storage overhead, privacy protection is one area where it falls short. Li *et al.* [16] present a lightweight data-sharing technique (LDSS) for mobile cloud computing. LDSS enhanced the structure of the access control tree to enable the mobile cloud-relevant procedure by employing the CP-ABE technique. Using this approach, a sizable portion of mobile devices' calculations are delegated to outside proxy servers. As users exchange data under mobile cloud settings, LDSS reduces the load on the mobile device. An approach for privilege-based multilevel organizational data sharing in [17] is proposed, privilege-based multilevel organizational data-sharing scheme (P-MOD). P-MOD extends the attribute-based encryption method by introducing a privilege-based access structure, making it simpler to distribute and manage massive data sets. Tests demonstrate that for implementing encryption and decryption as well as generating keys in a hierarchical system with many layers, the P-MOD technique may be superior to CP-ABE [18] and FH-CP-ABE [19]. The P-MOD scheme also has fewer operations overall when compared to the multilevel systems hierarchy attribute-based encryption (HABE) [20], [21], and FH-CP-ABE [19].

In a cloud-based setting, Li *et al.* [22] introduced a linear secret sharing technique (LSSS) matrix access structure based on a successful CP-ABE approach to dynamically upgrade the file and increase the efficacy of the policy. The plan's objectives are to defend against chosen-plaintext attack (CPA) and reduce the computation, connectivity, and storage expenses for both the data owner and the proxy cloud service provider (PCSP). Theoretical analysis and practical simulation show that the proposed approach outperforms policy update CP-ABE [23] in terms of efficiently administering policy changes and file updates. Zhang *et al.* [24] provide a privacy-preserving hidden access policy CP-ABE, hidden access policy cipher text policy attribute-based encryption (HP-CP-ABE) system with an efficient authority verification to ensure data security and protect user privacy. Zhong *et al.* [25] developed a novel migration model among the cloud provider to develop a key agreement and mutual authentication model on the elliptic curve cryptographic (ECC) scheme for peer-to-peer cloud, this scheme aims at developing trust among the participants. Shen *et al.* [26] developed a novel paradigm of data integrity without using private key storage, in here biometric-based data is used as the user's private key. Furthermore, a linear sketch with error correction and coding is utilized for user identification confirmation. A novel signature was designed for supporting block-less verifiability.

3. PROPOSED METHOD

In cloud computing, security possesses various aspects including data integrity, development of trust among the participants, and preserving the privacy of participants, cloud security model needs to be both secure and efficient considering the above aspect, this research work designs and develops a VPEH mechanism for overall secure data transmission among the multiple participants in groups. At first, we design and develop an enhanced homomorphic encryption model that utilizes the homomorphic encryption for three approaches *i.e.* tag generation, verification, and proof generation, unlike the existing protocol. A further additional level of security framework is incorporated to develop trust as a well-verifiable model.

3.1. Enhanced homomorphic encryption

This approach is novel as it uses homomorphic multiplication to compute the hash product of the challenge blocks, which increases security and efficiency. Moreover, the formation and verification of the signature are very secure because of elliptic curve encryption. A further additional level of security framework is incorporated to develop trust as a well-verifiable model. Below algorithm shows the enhanced homomorphic encryption.

Algorithm 1. Enhanced homomorphic encryption
Step 1: Enter the verification file and the frequency of challenge blocks (K).
Step 2: Divide the file into blocks of fixed size D, where D is the $block\_size$.
Step 3: Select a random number l that is comparatively prime irrespective of the elliptic curve group that is denoted as β.
Step 4: Generate β $(\text{private key}, \text{public key})$
Step 5: Select RN such that, $1 < RN < K - 1$
Step 6: Evaluate the product of the challenge blocks through the homomorphic multiplication:
$P = P1 \times P2 \times \ldots \times PK$
Step 7: A tag for the file, $\mathrm{J} = \mathrm{G} \times \mathrm{RN}^{\mathrm{l}}\ mod\ c$, where c is the order of β
Step 8: Evaluate the significance of the tag, Sign=$\mathrm{l}^{-1} \times (\mathrm{J} + \mathrm{a} \times \mathrm{G})\ mod\ c$, where a denotes the private key
Step 9: Transfer the tag, $file,$ and $sign$ to the recipient.
Step 10: On the recipient side, divide the file into blocks of size D
Step 11: Calculate the product of the hash (K), where $\mathrm{P} = \mathrm{P1} \times \mathrm{P2} \times \ldots . \times \mathrm{PK}$, is the homomorphic multiplier.
Step 12: Verification of signature upon computing $\mathrm{J}' = \mathrm{V}^{\mathrm{Sign}} \times \mathrm{z}^{\mathrm{P}}\ mod\ c$, here V is the generator of $\beta,\ z$ is the public key, P is the product of hash, and (K)
Step 13: If $\mathrm{J}' = \mathrm{J}, \text{file} = \text{authentic, else not authentic}$

3.2. Generation and updation

In this section, the data file is generated which is encrypted using symmetric encryption. Upon encryption a cipher text is generated the user amongst the group submits the data that is encrypted and further transferred to the μ. The μ is further provided with the public key and the private key, the public key is responsible to μ encrypts the re-encrypted message.

3.2.1. Data encryption

The following tasks are performed when a data file is to be uploaded by the user. The user here encrypts the data file ε for symmetric encryption $Encrypt_{\chi}()$ which is the common conference key χ. The cipher text $Cipher\ _{GM}$ is determined by $Cipher\ _{GM} = Encrypt_{\chi}()$, here the user amongst the group submits the data that is encrypted to the μ that consists of given in (1).

$$(\mathrm{AD}_{\mathrm{GM}}, \mathrm{AD}_{\mathrm{data,}}\ \mathrm{CT}\ _{\mathrm{MG}}, \nu_{\mathrm{data}},\ \mu) \quad (1)$$

Here μ is the signature extracted from the group member by $GS(\ )$ v_{data} estimates the current time. Here the μ checks the validity of the member by $ValidateSignal(\ )$ and $ValidateRV$. Once successful validation is done, the μ picks two parameters x and y that determines $= \mathrm{xy}$. Based on RSA public-key encryption [27]. The μ here picks a large integer u estimates the adjacent s, that satisfies $us \equiv 1\, mod\, (x-1)(y-1)$. The public key for the μis given by (u, x) whereas the private key is determined as (s, x). The public key for this μ encrypts the re-encrypted message $\mathrm{CT_G}$ as given here in (2). However, $\mathrm{CT_{cloud}}$ is uploaded to the cloud to the μ. Henceforth to validate a legal user. To authorize a legal user, the μ determines in (3). Where ai is the random number given and selected by the μ and $\mathrm{W} = s(\mathbb{R}, \mathbb{R})$ that necessarily generates a group sign for the row known as $(\mathrm{CT_{cloud}}, \mathrm{G_s})$. Along the secret key given as $(\mathrm{H_C}, \mathrm{j_C})$ issued by the cloud, which supports authentication of the messages uploaded. The tuple is given as $(\mathrm{CT_{cloud}}, \mathrm{G_s})$ with the sign μ is uploaded to the cloud.

$$\mathrm{CT_{cloud}} = (\mathrm{CT_G})^{\mathrm{t}} \tag{2}$$

$$\mathrm{G_s} = (\mathbb{R}^{\mathrm{s.a}}, \mathrm{s}, \mathrm{F^a}) \tag{3}$$

3.2.2. Group member repudiation

In this section, the repudiation of the group users is achieved by the μ who manages the PO. The PO consists of a tuple $(\mathrm{H_{i,}}\ \mathrm{j_i},\ \mathrm{T_i})$ depicts the member i along the private key $(\mathrm{H_{i,}}\ \mathrm{j_i})$ repudiated by time $\mathrm{T_i}$. In addition to this PO is bound with AD_{GM} and $sign(PO)$ to identify PO. Based on ValR(), a repudiate member is not validated to the cloud.

3.2.3. Key updation

The process of upgrading a key consists of two steps: first, updating the μ common conference key denoted as χ and private key (s, x). Noting down the encryption mechanism once more, along with the access control, and outsourced data, this prevents the collision performed on the cloud and repudiated randomly. In this proposed scheme the χ is updated in a specific period. Here PO is transferred and updated to the random χ. In contrast, the group's private key (s, x) is updated whenever the number of group members changes. From the below functions considered here, the update of the μ is determined. The repudiated user does not pose any threat due to the reason for access control and the μ.

a. The μ produces new public/private key pairs $(\mathrm{s}^*, \mathrm{x})$ and $(\mathrm{u}^*, \mathrm{x})$ and selects a new random a^*.
b. The μ here estimates the $\mathrm{G_s^*} = (\ \mathbb{R}^{\mathrm{s.a}}, \mathrm{s}, \mathrm{F^a})$ generates a group signature on the message through $\mathrm{GS().G_s^*}$ and the corresponding signature is uploaded to the cloud.
c. The cloud platform replaces $\mathrm{G_s}$ by $\mathrm{G_s^*}$ upon successful validation of signature. It then estimates.

$$\mathrm{CT_{cloud}^*} = (\mathrm{CT_{cloud}})^{\mathrm{u^*}/\mathrm{u}} \tag{4}$$

3.3. File access and traceability

3.3.1. Access file

To retrieve the data stored on the cloud, the following processes are performed. In addition, responds to the data requested as req to the user. The authenticated user obtains the required group data for the encryption secret key for the μ and the common conference key for the group.

a. The user here transmits the data request that consists of$(\mathrm{AD_{GM}}, \mathrm{AD_{data,}}, \mathrm{v},\ \mu)$ for the μ, where $\mathrm{AD_{data}}$ denotes the shared group data. AD Here depicts the current time and μ is the group signature on the message $((\mathrm{AD_{GM}}, \mathrm{AD_{data,}}, \mathrm{v}\ ).$
b. The μ transmits an authorization function determined as $\mathrm{s_{GM\rightarrow L}} = \ \mathbb{R}^{\mathrm{k_e}/\mathrm{u_p}}$ to the cloud after the successful validation of $\mathrm{ValS()}$ and $\mathrm{ValR()}$. $\mathrm{s_{GM\rightarrow L}}$ depicts the authenticated data from the μto group user. $\mathrm{k_e}$ depicts the secret key of the group member and $\mathrm{u_p}$ is the present private key of the μ.
c. Upon receiving the authenticated data from the μ the cloud platform estimates:

$$\begin{aligned} \mathrm{req} &= (s(\mathbb{R}^{\mathrm{k_e}/\mathrm{u_p}}, \mathrm{R}^{\mathrm{u_p.a_p}}), \mathrm{u_p}.\,\mathrm{M^{a_p}}) \\ &= (s(\mathbb{R}, \mathbb{R})^{\mathrm{k_e.a_p}}, \mathrm{u_p}.\,\mathrm{M^{a_p}}) \end{aligned} \tag{5}$$

d. Upon receiving the data requested and req from the cloud, the user with the secret key performs encryption $\mathrm{k_e}$ obtains the encryption secret key by the μ estimated as:

$$\mathrm{u_p} = (\mathrm{u_p}.\ \mathrm{M^{a_p}})/(\ s(\mathbb{R}, \mathbb{R})^{\mathrm{k_e.a_p}})^{1/\mathrm{k_e}} \tag{6}$$

3.3.2. Tracking

In this section, the μ is responsible to track the actual identity of the data owner when an argument occurs. The argument generated for the data file is given as $\mathrm{AD}_{\mathrm{data}}$, the μ obtains a signature β_{data} on the file. After validation of the signature's correctness and a repudiation validation, the μ performs the specific operations.

a. To compute $N_a = O_3 - (\zeta 1. O_1 + \zeta 2. O_2)$ by the master key $(\zeta 1, \zeta 2)$.

b. By analyzing the user list to determine the real identity of the data owner.

3.4. Security analysis

Here in this section, the security of the system is determined via data confidentiality, fault-tolerant property, anonymity, traceability, and access control. In this scheme where a users, the user, and the volunteer in the model are probabilistic polynomial time Turing machine as the opponent. A passive opponent is a person that attempts to learn the information about the outsourced data overheard on the communication bandwidth to capture the common conference key.

3.4.1. Security analysis for data confidentiality

In this scheme where a users, the user, and the volunteer in the model are probabilistic polynomial time Turing machine as the opponent. A passive opponent is a person that attempts to learn the information about the outsourced data overheard on the communication bandwidth to capture the common conference key. An opponent has access to the system parameters like that $\{\mathbb{R}, D, v_2(KD_a) | 0 \le a \le x-1\}$, thereby the session key i_a for the user, a is protected from the opponent. If $S \approx pl^Y$ in the model depicted as secure against a passive attack. $S \approx pl^Y$ shows two tuples of random variables. $C = \{\mathbb{R}, \alpha, v_2(KG_a), s(\mathbb{R}, \sum_{a=1}^{x} i_a\, \kappa_a) | 0 \le a \le x-1\}$, and $D = \left\{\mathbb{R}, \alpha, v_2(KG_a), \frac{b}{0} \le a \le x-1, b \in M_h^*\right\}$, cannot be distinguished. Here $b \in M_h^*$ is a randomly selected number. If $S \approx pl^Y$ for all the polynomial parameters the probability to distinguish C and D is smaller when compared to $\frac{1}{2} + \frac{1}{O(v)}$ for all $O(v)$. Here $v \in Z^+$ A security parameter in the key agreement model that determines the size of ρ, all the polynomials function in probabilistic time with v as the input. If the condition for $S_a \approx pl_a^Y$ holds for all the users_a , then $S \approx pl^Y$. For the reason of a discrete logarithmic problem on elliptic curves when V has a high value of more than 512-bits long we get $S_a \approx pl_a^Y$. Hence, we prove that $\Lambda_{a=0}^{x-1} C_a \approx_{pl} \Lambda_{a=0}^{x-1} D_a$, ($C \approx_{pl}$ D).

$$C_a = \{ \mathbb{R}, \alpha_{PB}, V_2(KG_a), s(\mathbb{R}, \textstyle\sum_{a=1}^{x} i_a\, \kappa_a) | 0 \le a \le x-1\} \tag{7}$$

$$D_a = \{ \mathbb{R}, \alpha_{PB}, V_2(KG_a), D_a\} \tag{8}$$

$$C = (\mathbb{R}, \alpha_{PB}, V_2(KG_a), s(\mathbb{R}, \textstyle\sum_{a=1}^{x} i_a\, \kappa_a) | 0 \le a \le x-1) \tag{9}$$

$$\begin{aligned} C &= (\mathbb{R}, \alpha_{PB}, V_2(KG_a), s(\mathbb{R}, \textstyle\sum_{a=1}^{x} i_a\, \kappa_a) | 0 \le a \le x-1) \\ &= (\mathbb{R}, \alpha_{PB}, V_2(KG_0), V_2(KG_1), \ldots, , V_2(KG_{x-1}), \\ &\quad s(\mathbb{R}, i_0\kappa_0). s(\mathbb{R}, i_1\kappa_1) \ldots \ldots . . s(\mathbb{R}, i_{x-1}\kappa_{x-1}) \\ &= \Lambda_{a=0}^{x-1} C_a \end{aligned} \tag{10}$$

$$\begin{aligned} D &= (\mathbb{R}, \alpha_{PB}, V_2(KG_a), d | 0 \le a \le x-1,\ b \in M_h^*) \\ &= (\mathbb{R}, \alpha_{PB}, V_2(KG_0), V_2(KG_1), \ldots, , V_2(KG_{x-1}), d) \\ &= \Lambda_{a=0}^{x-1} D_a \end{aligned} \tag{11}$$

3.4.2. Security framework for unknown

Here the original identity of the user who performs the signature is preserved, which implies the anonymity of this scheme. The group signature μ an entity acquires constraints such asβ_1, β_2 and β_3. The entire constraint cannot reveal the user who signs identity, given as;

$$R_i = \beta_3 - (\zeta 1\, \beta_1 + \zeta 2\, \beta_2).$$

3.4.3. Security framework for verification

The original identity of the user who signs can be tracked via the μ that involves tracking the scheme. The μ acquires R_i through the master key $(\zeta 1, \zeta 2)$ in an efficient manner. To reveal the original identity of the KG_a of the user who signs by examining the user list maintained.

3.4.4. Validation of the group signature

It is not possible for the attacker or intrudes to generate a group signature, a polynomial-dependent time-algorithm δ occurs that is capable of forging a group signature without any probability. In a random oracle, the algorithm ξ generates two valid signatures $(\mathrm{H}, \mu_0, \mathrm{f}, \mu_1)$ and $(\mathrm{H}, \mu_0, \mathrm{f}', \mu_1')$. Consequently, this algorithm obtains a secret key (y', C') by estimating $y' = \frac{\Delta c_h}{\Delta v}$ and$C' = \beta_3 - \frac{\Delta c_m + \Delta c_n}{\Delta v}$, here *q* is used to forge a valid group signature.

$$\begin{cases} \mu_0 = (\beta_1, \beta_{2,}\beta_3, \mathrm{f}, \mathcal{M}_{1,}\mathcal{M}_{2,}\mathcal{M}_3, \mathcal{M}_4, \mathcal{M}_5) \\ \mathrm{f} = \mathrm{j}(\mathscr{g}, \beta_1, \beta_{2,}\beta_3, \mathcal{M}_{1,}\mathcal{M}_{2,}\mathcal{M}_3, \mathcal{M}_4, \mathcal{M}_5) \\ \mathrm{f}' = \mathrm{j}'(\beta_1, \beta_{2,}\beta_3, \mathrm{f}, \mathcal{M}_{1,}\mathcal{M}_{2,}\mathcal{M}_3, \mathcal{M}_4, \mathcal{M}_5) \\ \qquad \mu_1 = (c_m, c_n, c_k, c_{\rho_1}, c_{\rho_2}) \\ \qquad \mu_1' = (c_m', c_{n,}' c_{h,}' c_{\rho_1}' c_{\rho_2}') \end{cases} \tag{12}$$

$$\begin{cases} c_m = g_m + km & c_m' = g_m + c'm \\ c_n = g_n + kn & c_n' = g_n + c'n \\ c_h = g_h + kh & c_h' = g_h + c'h \\ c_{\rho_1} = g_{\rho_1} + k\rho_1 & c_{\rho_1}' = g_{\rho_1} + c'\rho_1 \\ c_{\rho_2} = g_{\rho_2} + k\rho_2 & c_{\rho_2}' = g_{\rho_2} + c'\rho_2 \end{cases} \tag{13}$$

3.4.5. Security framework for access control

The proposed model achieves effective access control from the above we can conclude that a successfully registered user who has not been repudiated is capable of accessing the cloud and the repudiated user is not able to access the cloud after repudiation. Users, which have not been able to access the cloud after repudiation. The algorithm $ValidateSignal()$ for users who are not repudiated the following equations are established: $\mathcal{M}_1 = \mathcal{M}_1$, $\mathcal{M}_2 = \mathcal{M}_{2,}$ $\mathcal{M}_3 = \mathcal{M}_3$,$\mathcal{M}_4 = \mathcal{M}_4$, $\mathcal{M}_5 = \mathcal{M}_5$. Henceforth the hash value *c* is equal to $\mathrm{q}_1(\mathscr{g}, \beta_1, \beta_{2,}\beta_3, \mathcal{M}_{1,}\mathcal{M}_{2,}\mathcal{M}_3, \mathcal{M}_4, \mathcal{M}_5)$, the algorithm returns 'true'. We can prove that $\mathcal{M}_1 = \mathcal{M}_1$ by the given in (14). Similarly, to this we have $\mathcal{M}_2 = \mathcal{M}_{2,}$,$\mathcal{M}_4 = \mathcal{M}_4$, $\mathcal{M}_5 = \mathcal{M}_5$. Although $\mathcal{M}_3 = \mathcal{M}_3$ for the following reason. This is mathematically represented in (15). The users here are repudiated by the μ who does not access the cloud after repudiation. Accordingly, the signature of the group is developed by a repudiated user, there exists an l_g that shows the equation $\mathrm{s}(\mathcal{M}_3 - \mathrm{l_g}, \mathrm{Q_Z}) = \mathrm{t}$, as mention in (16).

$$\begin{aligned} &\mathcal{M}_1 = c_m . \mathrm{G} - \mathrm{c}.\ \beta_1 \\ &= (g_m + km). \mathrm{G} - \mathrm{c.m.G} \\ &= g_m . \mathrm{G} \\ &= \mathcal{M}_1 \end{aligned} \tag{14}$$

$$\begin{aligned} \varpi_3 &= \left(\frac{\mathrm{s}(\mathcal{M}_3, \mathrm{P})}{\mathrm{s}(\mathrm{K}, \mathrm{K})}\right)^{\mathrm{C}} . \mathrm{s}(\mathcal{M}_3, \mathrm{K})^{c_h} . \mathrm{s}(\mathrm{Q}, \mathrm{P})^{-c_m - c_n} \\ &= \mathrm{s}(\mathrm{Q}, \mathrm{K})^{-c_{\rho_1} - c_{\rho_1}} \\ &= \left(\frac{\mathrm{s}(\mathcal{M}_3, \mathrm{P})}{\mathrm{s}(\mathrm{K}, \mathrm{K})}\right)^{\mathrm{C}} . \mathrm{s}(\mathcal{M}_3, \mathrm{K})^{r_{h+kh}} . \mathrm{s}(\mathrm{Q}, \mathrm{P})^{-r_m - cm - c_n - cn} \\ &= \mathrm{s}(\mathrm{Q}, \mathrm{K})^{-r_{\rho_1} - khm - r_{\rho_1} - khn} \\ &= \left(\frac{\mathrm{s}(\mathcal{M}_3, \mathrm{P})}{\mathrm{s}(\mathrm{K}, \mathrm{K})}\right)^{\mathrm{C}} . \mathrm{s}(\mathcal{M}_3, \mathrm{hK})^{\mathrm{c}} . \mathrm{s}(-(\mathrm{m} + \mathrm{n})\mathrm{Q}, \mathrm{P} + \mathrm{hK})^{\mathrm{C}} \\ &= \mathrm{s}((\mathcal{M}_3, \mathrm{P})^{r_h} . \mathrm{s}(\mathrm{Q}, \mathrm{P})^{-r_m - r_n} . \mathrm{s}(\mathrm{Q}, \mathrm{K})^{-c_{\rho_1} - c_{\rho_1}} \\ &= \left(\frac{\mathrm{s}(\mathcal{M}_3, \mathrm{P})}{\mathrm{s}(\mathrm{K}, \mathrm{K})}\right)^{\mathrm{C}} . \mathrm{s}(\mathcal{M}_3, \mathrm{hK})^{\mathrm{c}} . \mathrm{s}(-(\mathrm{m} + \mathrm{n})\mathrm{Q}, \mathrm{P} + \mathrm{hK})^{\mathrm{C}} . \varpi_3 \\ &= \left(\frac{\mathrm{s}(\mathcal{M}_3, \mathrm{P})}{\mathrm{s}(\mathrm{K}, \mathrm{K})}\right)^{\mathrm{C}} . \mathrm{s}(\mathcal{M}_3 - (\mathrm{m} + \mathrm{n})\mathrm{Q}, \mathrm{P} + \mathrm{hK})^{\mathrm{C}} \\ &= \mathrm{s}(\mathcal{M}_3, \mathrm{P})^{-\mathrm{c}} . \varpi_3 \\ &= \left(\frac{\mathrm{s}(\mathcal{M}_3, \mathrm{P}) + \mathrm{hK}}{\mathrm{s}(\mathrm{K}, \mathrm{K})}\right)^{\mathrm{C}} . \varpi_3 \end{aligned}$$

$$
\begin{aligned}
&= \left(\frac{s\left(\frac{1}{(h+o)K},(O+h)K\right)}{s(K,K)}\right)^{C} . \varpi_3 \\
&= \varpi_3 \qquad (15)
\end{aligned}
$$

$$
\begin{aligned}
&s(\mathcal{M}_3 - l_g, Q_Z\ ) \\
&= s\ (l_g + (\ m + n). Q - l_g, Q_Z) \\
&= s(m. Q. Q_Z). s(n. Q. Q_Z) \\
&= s(m. A. \zeta_1. Q_Z). s(n\ . B, \zeta_2. Q_Z) \\
&s(\mathcal{M}_1, Q_1). s((\mathcal{M}_2, Q_2) \qquad (16)
\end{aligned}
$$

4. PERFORMANCE EVALUATION

In this section, the cost comparison is carried out along the proposed system with the existing system. The performance is evaluated through a series of experiments carried out. In the first graph, a comparison is carried out between the proposed and the existing methods. For performance comparison, the method consisting of the same function as the proposed system (PS) is evaluated. The proposed method is compared with three methods, one is the research development and innovation center (RDIC) scheme with privacy protection, and the other is used in dynamic updation.

4.1. Experimental setup

VPEH is evaluated on computing cost, there are three major computations i.e. tag generation, Proof generation, and Proof verification; The no of data blocks considered in this approach are 200, 400, 600, 800, and 1,000. The PS is evaluated on the methods that implement the signature operation. The cost is high and the time linear increases as the number of blocks is increased. The cost of data integrity is exponentially increased in comparison to the PS with the existing system. The timer is increased for the blocks from 200 to 1,000 upon each iteration of each analysis. The performance is evaluated for each data-integrity proof mechanism and the graphs are plotted for each.

4.2. Computation cost

The following notations are considered to denote operations in the proposed method. Assume C_{ef}, C_{mf}, C_{po} for the first group irrespectively. z denotes the total no of blocks, j denotes the challenge blocks, C_{gen} and C_{ver} for generation and verification of signatures. In the PS considered a block is segmented into various sections, henceforth resulting in a reduction of storage cost. The methods by [27]–[29] divide the file into z blocks into h sectors. Segmentation and exponential blocks are expensive and mainly estimates the cost. Functions such as segmentation and exponentiation estimate the PS's cost. The other methods such as hashing and addition have the least costs, which upon further analyses are neglected. The cost is compared for three methods [27]–[29]. Further, the existing model [30] is also considered for comparison with VPEH.

4.3. Experimental results

4.3.1. Tag generation

Figure 2 presents the comparison of tag generation over the various number of blocks considering the different existing models. The tag generation is evaluated wherein RDIC, identity-based RDIC, data deduplication, and Identity-based remote data possession checking existing system (RDPC-ES) are compared with the proposed system, and the graph is plotted for 200, 400, 600, 800, and 1,000 blocks. In RDIC, for 200 blocks the value observed is 2.3, identity-based RDIC for 200 blocks observes a value of 2.5, data deduplication observes a value of 3.2, identity-based remote data possession checking (RDPC) observes a value of 3.4 and PS observes a value of 0.000101. In RDIC for 400 blocks the value observed is 4.7, in identity-based RDIC for 400 blocks observes a value of 5, data deduplication observes a value of 5.2, identity-based RDPC observes a value of 5.7 and PS observes a value of 0.001672. In the RDIC, method for 600 blocks the value observed is 6.8, identity-based RDIC for 600 blocks observes a value of 7.2, data deduplication observes a value of 7.6, identity-based RDPC observes a value of 8 and PS observes a value of 0.0008. In RDIC for 800 blocks, the value observed is 8.3, identity-based RDIC for 800 blocks observes a value of 9, data deduplication observes a value of 9.5, identity based RDIC observes a value of 10.1 and PS gives a value of 0.000344. In RDIC for 1,000 blocks the value observed is 12, in identity-based RDIC for 200 blocks the value observed is 12.3, Wu *et al.* [29] observes a value of 13, Bian *et al.* [30] observes a value of 13.8 and PS observes a value of 0.000234.

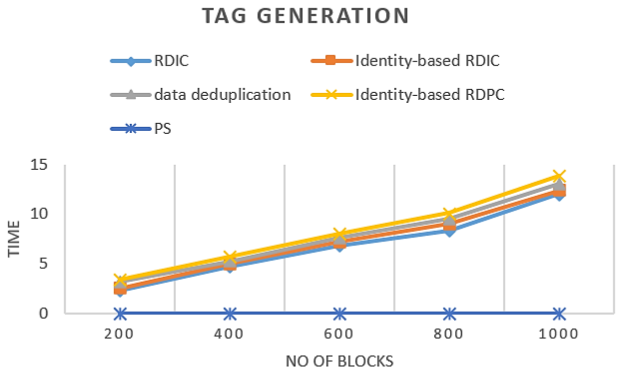
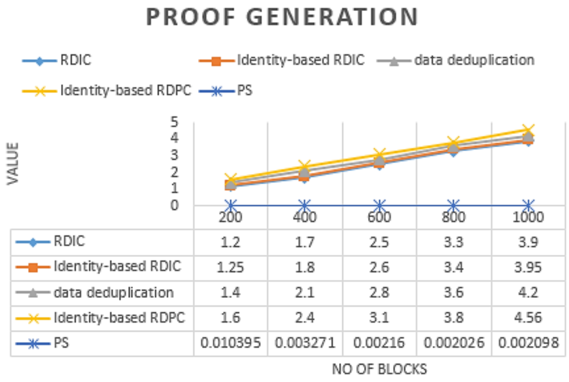


Figure 2. Tag generation comparison

4.3.2. Proof generation

Figure 3 present the proof generation evaluation is compared with the proposed system and the graph is plotted for 200, 400, 600, 800, and 1,000 blocks. In RDIC for 200 blocks, the value observed is 1.2, in identity-based RDIC for 200 blocks, the value observed is 1.25, data deduplication for 200 blocks observes a value of 1.4, identity-based RDPC observes a value of 1.6 and PS observes a value of 0.010395. In RDIC for 400 blocks, the value observed is 1.7, identity-based RDIC for 400 blocks observes a value of 1.8, data deduplication observes a value of 2.1, identity-based RDPC observes a value of 2.4 and PS observes a value of 0.003271. In RDIC for 600 blocks, the value observed is 2.5, identity-based RDIC for 600 blocks observes a value of 2.6, data deduplication observes a value of 2.8, identity-based RDPC observes a value of 3.1 and PS observes a value of 0.002026. In RDIC for 800 blocks, the value observed is 3.3, identity-based RDIC for 800 blocks observes a value of 3.4, data deduplication observes a value of 3.6, identity-based RDPC observes a value of 3.8 and PS gives a value of 0.002026. In RDIC for 1,000 blocks, the value observed is 3.9, identity-based RDIC for 200 blocks observes a value of 3.95, data deduplication gives a value of 4.2, identity-based RDPC observes a value of 4.56 and PS observes a value of 0.002098. Figure 3 shows the proof generation comparison.



	200	400	600	800	1000
RDIC	1.2	1.7	2.5	3.3	3.9
Identity-based RDIC	1.25	1.8	2.6	3.4	3.95
data deduplication	1.4	2.1	2.8	3.6	4.2
Identity-based RDPC	1.6	2.4	3.1	3.8	4.56
PS	0.010395	0.003271	0.00216	0.002026	0.002098

Figure 3. Proof generation comparison

4.3.3. Verification

Figure 4 depicts the verification comparison, the verification mechanism is evaluated here wherein the existing system is compared with the proposed system, and the graph is plotted for 200, 400, 600, 800, and 1,000 blocks. In RDIC for 200 blocks, the value observed is 2.2, identity-based RDIC for 200 blocks observes a value of 2.5, data deduplication observes a value of 2.8, identity-based RDPC observes a value of 4.1 and PS observes a value of 0.057744. In RDIC for 400 blocks, the value observed is 3.1, identity-based RDIC for 400 blocks observes a value of 3.2, data deduplication observes a value of 4, identity-based RDPC observes a value of 7.5 and PS observes a value of 0.002558. The RDIC for 600 blocks observes a value of 4.3, identity-based RDIC for 600 blocks observes a value of 5, data deduplication observes a value of 5.8, identity-based RDPC observes a value of 11.2 and PS observes a value of 0.001421. In RDIC for 800 blocks, the value observed is 7.2, identity-based RDIC for 800 blocks observes a value of 7.4, data deduplication observes a value of 8, identity-based RDPC observes a value of 13 and PS observes a value of 0.001257. In RDIC for 1,000 blocks, the value observed is 8.8, identity-based RDPC for 200 blocks observes a value of 9, data deduplication observes a value of 9.5, identity-based RDPC observes a value of 15.2 and PS observes a value of 0.001319.

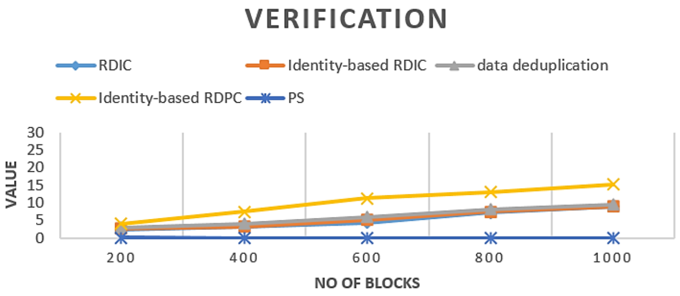


Figure 4. Verification comparison

4.4. Comparative analysis

The comparative analysis is done here for cost computation purposes. The evaluation is carried out by comparing the existing system with the proposed system for target generation, proof generation, and verification. The detail of each is explained in the below section which depicts the comparative study for each.

4.4.1. Tag generation

The comparative analysis is carried out for the tag generation, the PS is compared with the existing system, for 200 blocks the existing system (ES) value observed is 3.4 whereas the PS value is observed at 0.000101 For 400 blocks the ES value is observed at 5.7 and PS value is observed as 0.001672. For 600 blocks, the ES value observed is 8 and the PS value is observed as 0.0008. For 800 blocks, the ES value observed is 10.1 and the PS value observed is 0.000344. For 1,000 blocks, the ES value observed is 13.8 and the PS value observed is 0.000234. In the end to conclude that our PS performs better than the existing system, concerning cost computation the time utilized is less in comparison with the existing system resulting in an overall improvisation of 200%. Table 1 gives the comparative analysis for tag generation for the no of blocks considered.

Table 1. Tag generation comparative analysis

No of blocks	ES	PS
200	3.4	0.000101
400	5.7	0.001672
600	8	0.0008
800	10.1	0.000344
1,000	13.8	0.000234

4.4.2. Proof generation

The comparative analysis is carried out for the proof generation, the PS is compared with the existing system, for 200 blocks the ES value observed is 1.6 and the PS value observed is 0.010395. For 400 blocks, the ES value observed is 2.4 and the PS value observed is 0.003271. For 600 blocks, the ES value observed is 3.1 and the PS value observed is 0.00216. For 800 blocks, the ES value observed is 3.8 and the PS value observed is 0.002026. For 1000 blocks, the ES value observed is 4.56 and the PS value is observed as 0.002098. In the end to conclude that our PS performs better than the existing system, concerning cost computation the time utilized is less in comparison with the existing system resulting in an overall improvisation of 200%. Table 2 shows the proof generation comparative analysis.

Table 2. Proof generation comparative analysis

No of blocks	ES	PS
200	1.6	0.010395
400	2.4	0.003271
600	3.1	0.00216
800	3.8	0.002026
1,000	4.56	0.002098

4.4.3. Verification

The comparative analysis is carried out for verification, the PS is compared with the existing system, for 200 blocks the ES value observed is 4.1 and the PS value is observed as 0.057744. For 400 blocks, the ES value observed is 7.5 and the PS value is observed as 0.002558. For 600 blocks, the ES value observed is 11.2 and the PS value observed is 0.001421. For 800 blocks, the ES value observed is 13 and the PS value is observed as 0.001257. For 1,000 blocks, the ES value observed is 15.2 and the PS value is observed as 0.001319. In the end to conclude that our PS performs better than the existing system, concerning cost computation the time utilized is less in comparison with the existing system resulting in an overall improvisation of 200%. Table 3 shows the verification comparative analysis.

Table 3. Verification comparative analysis

No of blocks	ES	PS
200	4.1	0.057744
400	7.5	0.002558
600	11.2	0.001421
800	13	0.001257
1,000	15.2	0.001319

5. CONCLUSION

A secure cloud-computing model has been in demand since the development of a particular computing model, recent development in technologies and high computation power has made security more vulnerable especially when there are multiple participants involved. Furthermore, considering the framework of the group-based model, privacy becomes even major along with verification and access control. This research work designs and develop a novel VPEH encryption model for secured group data sharing; at first novel homomorphic, encryption is designed for block computations, and further additional security framework is incorporated for group-based verification and access control. Verifiable PEH is evaluated by considering parameters such as tag generation cost, proof generation cost, and proof verification cost by varying the number of blocks and challenge blocks. The comparative analysis carried out ensures that the proposed approach ensures better performance in comparison with the existing system with an overall improvisation of 20% for target generation, proof generation, and verification. Although VPEH provides marginal improvisation over the existing model, the recent development of blockchain and deep learning architecture can be incorporated for further efficiency in terms of the integrity of the data.

REFERENCES

[1] T. Dillon, C. Wu, and E. Chang, "Cloud computing: Issues and challenges," in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, 2010, pp. 27–33, doi: 10.1109/AINA.2010.187.

[2] I. Gupta, A. K. Singh, C.-N. Lee, and R. Buyya, "Secure data storage and sharing techniques for data protection in cloud environments: A systematic review, analysis, and future directions," *IEEE Access*, vol. 10, pp. 71247–71277, 2022, doi: 10.1109/ACCESS.2022.3188110.

[3] Q. Huang, Y. Yang, W. Yue, and Y. He, "Secure data group sharing and conditional dissemination with multi-owner in cloud computing," *IEEE Transactions on Cloud Computing*, vol. 9, no. 4, pp. 1607–1618, Oct. 2021, doi: 10.1109/TCC.2019.2908163.

[4] B. Alouffi, M. Hasnain, A. Alharbi, W. Alosaimi, H. Alyami, and M. Ayaz, "A systematic literature review on cloud computing security: Threats and mitigation strategies," *IEEE Access*, vol. 9, pp. 57792–57807, 2021, doi: 10.1109/ACCESS.2021.3073203.

[5] T. Shang, F. Zhang, X. Chen, J. Liu, and X. Lu, "Identity-based dynamic data auditing for big data storage," *IEEE Transactions on Big Data*, vol. 7, no. 6, pp. 913–921, Dec. 2021, doi: 10.1109/TBDATA.2019.2941882.

[6] S. Li, Y. Zhang, C. Xu, and K. Chen, "Cryptoanalysis of an authenticated data structure scheme with public privacy-preserving auditing," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2564–2565, 2021, doi: 10.1109/TIFS.2021.3059270.

[7] R. Gupta, I. Gupta, A. K. Singh, D. Saxena, and C.-N. Lee, "An IoT-centric data protection method for preserving security and privacy in cloud," *IEEE Systems Journal*, vol. 17, no. 2, pp. 2445–2454, Jun. 2023, doi: 10.1109/JSYST.2022.3218894.

[8] X. Yang, M. Wang, X. Wang, G. Chen, and C. Wang, "Stateless cloud auditing scheme for non-manager dynamic group data with privacy preservation," *IEEE Access*, vol. 8, pp. 212888–212903, 2020, doi: 10.1109/ACCESS.2020.3039981.

[9] P. Pujar, A. Kumar, and V. Kumar, "Efficient plant leaf detection through machine learning approach based on corn leaf image classification," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 1, pp. 1139–1148, Mar. 2024, doi: 10.11591/ijai.v13.i1.pp1139-1148.

[10] Z. Wen, J. Cala, P. Watson, and A. Romanovsky, "Cost effective, reliable, and secure workflow deployment over federated clouds," *IEEE Transactions on Services Computing*, vol. 10, no. 6, pp. 929–941, Nov. 2017, doi: 10.1109/TSC.2016.2543719.

[11] A. Al-Haj, G. Abandah, and N. Hussein, "Crypto-based algorithms for secured medical image transmission," *IET Information Security*, vol. 9, no. 6, pp. 365–373, Nov. 2015, doi: 10.1049/iet-ifs.2014.0245.

[12] K. Liang *et al.*, "A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing," *Future Generation Computer Systems*, vol. 52, pp. 95–108, Nov. 2015, doi: 10.1016/j.future.2014.11.016.

[13] S. H. Sreedhara, V. Kumar, and S. Salma, "Efficient big data clustering using adhoc fuzzy C means and auto-encoder CNN," in *Inventive Computation and Information Technologies- Lecture Notes in Networks and Systems, vol 563*, Springer, Singapore, 2023, pp. 353–368, doi: 10.1007/978-981-19-7402-1_25.

[14] H. Liu, X. Li, M. Xu, R. Mo, and J. Ma, “A fair data access control towards rational users in cloud storage,” *Information Sciences*, vol. 418–419, pp. 258–271, Dec. 2017, doi: 10.1016/j.ins.2017.07.023.

[15] Z. Liu, Z. L. Jiang, X. Wang, and S. M. Yiu, “Practical attribute-based encryption: Outsourcing decryption, attribute revocation and policy updating,” *Journal of Network and Computer Applications*, vol. 108, pp. 112–123, Apr. 2018, doi: 10.1016/j.jnca.2018.01.016.

[16] R. Li, C. Shen, H. He, X. Gu, Z. Xu, and C.-Z. Xu, “A lightweight secure data sharing scheme for mobile cloud computing,” *IEEE Transactions on Cloud Computing*, vol. 6, no. 2, pp. 344–357, Apr. 2018, doi: 10.1109/TCC.2017.2649685.

[17] E. Zaghloul, K. Zhou, and J. Ren, “P-MOD: Secure privilege-based multilevel organizational data-sharing in cloud computing,” *IEEE Transactions on Big Data*, vol. 6, no. 4, pp. 804–815, Dec. 2020, doi: 10.1109/TBDATA.2019.2907133.

[18] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in *2007 IEEE Symposium on Security and Privacy (SP ’07)*, May 2007, pp. 321–334, doi: 10.1109/SP.2007.11.

[19] S. Wang, J. Zhou, J. K. Liu, J. Yu, J. Chen, and W. Xie, “An Efficient file hierarchy attribute-based encryption scheme in cloud computing,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1265–1277, Jun. 2016, doi: 10.1109/TIFS.2016.2523941.

[20] G. Wang, Q. Liu, and J. Wu, “Hierarchical attribute-based encryption for fine-grained access control in cloud storage services,” in *Proceedings of the 17th ACM conference on Computer and communications security*, Oct. 2010, pp. 735–737, doi: 10.1145/1866307.1866414.

[21] G. Wang, Q. Liu, J. Wu, and M. Guo, “Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers,” *Computers & Security*, vol. 30, no. 5, pp. 320–331, Jul. 2011, doi: 10.1016/j.cose.2011.05.006.

[22] J. Li *et al.*, “An efficient attribute-based encryption scheme with policy update and file update in cloud computing,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6500–6509, Dec. 2019, doi: 10.1109/TII.2019.2931156.

[23] K. Yang, X. Jia, and K. Ren, “Secure and verifiable policy update outsourcing for big data access control in the cloud,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3461–3470, Dec. 2015, doi: 10.1109/TPDS.2014.2380373.

[24] L. Zhang, Y. Cui, and Y. Mu, “Improving security and privacy attribute based data sharing in cloud computing,” *IEEE Systems Journal*, vol. 14, no. 1, pp. 387–397, Mar. 2020, doi: 10.1109/JSYST.2019.2911391.

[25] H. Zhong, C. Zhang, J. Cui, Y. Xu, and L. Liu, “Authentication and Key agreement based on anonymous identity for peer-to-peer cloud,” *IEEE Transactions on Cloud Computing*, vol. 10, no. 3, pp. 1592–1603, Jul. 2022, doi: 10.1109/TCC.2020.3004334.

[26] W. Shen, J. Qin, J. Yu, R. Hao, J. Hu, and J. Ma, “Data integrity auditing without private key storage for secure cloud storage,” *IEEE Transactions on Cloud Computing*, vol. 9, no. 4, pp. 1408–1421, Oct. 2021, doi: 10.1109/TCC.2019.2921553.

[27] Y. Yu *et al.*, “Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 767–778, Apr. 2017, doi: 10.1109/TIFS.2016.2615853.

[28] J. Li, H. Yan, and Y. Zhang, “Identity-based privacy preserving remote data integrity checking for cloud storage,” *IEEE Systems Journal*, vol. 15, no. 1, pp. 577–585, Mar. 2021, doi: 10.1109/JSYST.2020.2978146.

[29] Y. Wu, Z. L. Jiang, X. Wang, S. M. Yiu, and P. Zhang, “Dynamic data operations with deduplication in privacy-preserving public auditing for secure cloud storage,” in *22017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, Jul. 2017, pp. 562–567, doi: 10.1109/CSE-EUC.2017.104.

[30] G. Bian, R. Zhang, and B. Shao, “Identity-based privacy preserving remote data integrity checking with a designated verifier,” *IEEE Access*, vol. 10, pp. 40556–40570, 2022, doi: 10.1109/ACCESS.2022.3166920.

BIOGRAPHIES OF AUTHORS

Jayalakshmi Karemallaiah is currently working as an assistant professor in the Department of Computer Science and Engineering, Dr. Ambedkar Institute of Technology, Bangalore, India. She has obtained Bachelor's of Engineering (BE) degree in computer science and engineering from Mysore University, Master’s degree (M.Tech.) computer network engineering from VTU in 2009. And currently she is a research scholar at Dr. Ambedkar Institute of Technology doing her Ph.D. in computer science and engineering. She has attended many workshops and induction programs conducted by various universities. Her areas of interest are cloud computing and computer networks. She can be contacted at email: jayalakshmi_112@rediffmail.com.

Prabha Revaiah is currently working as a professor in the Department of Computer Science and Engineering, Dr. Ambedkar Institute of Technology, Bangalore, India. She obtained her Bachelor of Engineering degree in computer science and engineering branch from Mysore University, M.E. in computer science and engineering from Computer Science Department, UVCE, Bangalore University in the year 2003. She has 30 years of teaching experience. She was awarded Ph.D. in Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bangalore. Her research interest is in the area of wireless sensor networks and IoT. She can be contacted at email: prabha.cs@drait.edu.in.