

Homomorphic technique for group data sharing in cloud computing environment

Jayalakshmi Karemallaiah, Prabha Revaiah

Department of Computer Science and Engineering, Dr. Ambedkar Institute of Technology, Bangalore, India

Article Info

Article history:

Received Jan 6, 2024

Revised Aug 7, 2024

Accepted Aug 14, 2024

Keywords:

Cloud computing security

Data sharing

Homomorphic encryption

Privacy

Verifiable privacy-aware

enhanced homomorphic

ABSTRACT

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) 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 make 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 models by varying the number of blocks and number of challenge blocks.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Jayalakshmi Karemallaiah

Department of Computer Science and Engineering, Dr. Ambedkar Institute of Technology

Bangalore, India

Email: jayalakshmi_112@rediffmail.com

1. INTRODUCTION

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 [1]. The 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 [2]. Individuals and their collaborations are consequently migrating to the cloud platform to utilize its services [3]. With the expansion of cloud computing techniques, it will be difficult to facilitate the migration of all enterprises to cloud platforms shortly [4]. Figure 1 shows the general framework for sharing environment.

Figure 1 here depicts a shared environment, where the organization's crucial information needs 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 [5]. 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.

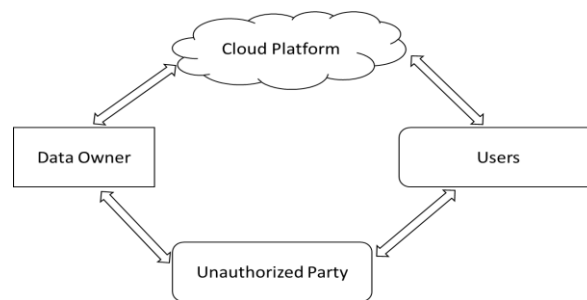


Figure 1. General framework for sharing environment

Liu *et al.* [6] proposed a ciphertext-policy attribute-based encryption (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.* [7] presents a lightweight data sharing scheme (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 (P-MOD). Sharing in [8] is proposed, 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 [9] and file hierarchy-based ciphertext-policy attribute-based encryption (FH-CP-ABE) [10].

The P-MOD scheme also has fewer operations overall when compared to the multilevel systems [11], [12], and FH-CP-ABE. In a cloud-based setting, studies [13]–[15] 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 attacks (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 [16] in terms of efficiently administering policy changes and file updates. The studies [17]–[19] provide a hidden policy ciphertext-policy attribute-based encryption (HP-CP-ABE) system with an efficient authority verification. To ensure data security and protect user privacy. Zhong *et al.* [16] developed a novel migration model among the cloud providers to develop a key agreement and mutual authentication model on the elliptic curve cryptography (ECC) scheme for peer-to-peer cloud, this scheme aims at developing trust among the participants [20]. Developed a novel paradigm of data integrity without using the 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 [21]. Novel signatures were designed to support block less verifiability. 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:

- VPEH is a privacy-aware enhanced homomorphic-based security framework in the cloud that aims to verify the users and data.
- Developed enhanced homomorphic mechanism uses homomorphic multiplication to compute the hash product of challenged blocks to be more efficient.
- An 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 presents the framework of VPEH along with the algorithm and mathematical model, VPEH is evaluated in the third section of the research.

2. PROPOSED METHOD

This approach is novel as it uses homomorphic multiplication to compute the hash product of the challenge blocks. This increases security and efficiency. Moreover, the formation and verification of the signature are very secure because of elliptic curve 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 (private key, public key).

Step 5: Select RN such that, $1 < RN < K - 1$.

Step 6: Evaluate the product of challenge blocks through the homomorphic multiplication $P = P1 \times P2 \times \dots \times PK$.

Step 7: A tag for the file, $J = G \times RN^l \text{ mod } c$, where c is the order of β .

Step 8: Evaluate the signature of the tag, $Sign = l^{-1} \times (J + a \times G) \text{ mod } c$, 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 $P = P1 \times P2 \times \dots \times PK$, is the homomorphic multiplier.

Step 12: Verification of signature upon computing $J' = V^{\text{Sign}} \times z^P \text{ mod } c$, where V is the generator of β , z is the public key, P is the product of hash, and (K).

Step 13: If $J' = J$, *file = authentic*, else not authentic.

2.1. Security framework for unknown

Here the original identity of the user who performs the signature is preserved. That implies the anonymity of this scheme. The group signature μ an entity acquires constraints such as β_1, β_2 and β_3 . The entire constraint cannot reveal the user who signs identity, given as $R_i = \beta_3 - (\zeta_1 \beta_1 + \zeta_2 \beta_2)$.

2.2. 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 (ζ_1, ζ_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.

2.3. Validation of the group signature

It is not possible for the attacker or intruder 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 (H, μ_0, f, μ_1) and (H, μ_0', f', μ_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.

$$\begin{cases} \mu_0 = (\beta_1, \beta_2, \beta_3, f, \mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4, \mathcal{M}_5) \\ f = j(\mathcal{G}, \beta_1, \beta_2, \beta_3, \mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4, \mathcal{M}_5) \\ f' = j'(\beta_1, \beta_2, \beta_3, f, \mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4, \mathcal{M}_5) \\ \mu_1 = (c_m, c_n, c_k, c_{\rho_1}, c_{\rho_2}) \\ \mu_1' = (c'_m, c'_n, c'_h, c'_{\rho_1}, c'_{\rho_2}) \end{cases} \quad (1)$$

$$\begin{cases} c_m = g_m + km & c'_m = g_m + c'm \\ c_n = g_n + kn & c'_n = g_n + c'n \\ c_h = g_h + kh & c'_h = g_h + c'h \\ c_{\rho_1} = g_{\rho_1} + k\rho_1 & c'_{\rho_1} = g_{\rho_1} + c'\rho_1 \\ c_{\rho_2} = g_{\rho_2} + k\rho_2 & c'_{\rho_2} = g_{\rho_2} + c'\rho_2 \end{cases} \quad (2)$$

2.4. 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 $q_1(\mathcal{g}, \beta_1, \beta_2, \beta_3, \mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4, \mathcal{M}_5)$, the algorithm returns 'true'. We can prove that $\mathcal{M}_1 = \mathcal{M}_1$ by (3). 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 mentioned in (4). 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 $s(\mathcal{M}_3 - l_g, Q_Z) = t$, as shown in (5).

$$\begin{aligned} \mathcal{M}_1 &= c_m \cdot G - c \cdot \beta_1 \\ &= (g_m + km) \cdot G - c \cdot m \cdot G \\ &= g_m \cdot G \\ &= \mathcal{M}_1 \end{aligned} \quad (3)$$

$$\begin{aligned} \omega_3 &= \left(\frac{s(\mathcal{M}_3, P)}{s(K, K)} \right)^c \cdot s(\mathcal{M}_3, K)^{c_h} \cdot s(Q, P)^{-c_m - c_n} \\ &= s(Q, K)^{-c_{\rho_1} - c_{\rho_1}} \\ &= \left(\frac{s(\mathcal{M}_3, P)}{s(K, K)} \right)^c \cdot s(\mathcal{M}_3, K)^{r_h + k_h} \cdot s(Q, P)^{-r_m - c_m - c_n - c_n} \\ &= s(Q, K)^{-r_{\rho_1} - k_{hm} - r_{\rho_1} - k_{hn}} \\ &= \left(\frac{s(\mathcal{M}_3, P)}{s(K, K)} \right)^c \cdot s(\mathcal{M}_3, hK)^c \cdot s(-(m+n)Q, P + hK)^c \\ &= s((\mathcal{M}_3, P)^{r_h} \cdot s(Q, P)^{-r_m - r_n} \cdot s(Q, K)^{-c_{\rho_1} - c_{\rho_1}} \\ &= \left(\frac{s(\mathcal{M}_3, P)}{s(K, K)} \right)^c \cdot s(\mathcal{M}_3, hK)^c \cdot s(-(m+n)Q, P + hK)^c \cdot \omega_3 \\ &= \left(\frac{s(\mathcal{M}_3, P)}{s(K, K)} \right)^c \cdot s(\mathcal{M}_3 - (m+n)Q, P + hK)^c \\ &= s(\mathcal{M}_3, P)^{-c} \cdot \omega_3 \\ &= \left(\frac{s(\mathcal{M}_3, P) + hK}{s(K, K)} \right)^c \cdot \omega_3 \\ &= \left(\frac{s\left(\frac{1}{(h+o)K}, (O+h)K\right)}{s(K, K)} \right)^c \cdot \omega_3 \\ &= \omega_3 \end{aligned} \quad (4)$$

$$\begin{aligned} s(\mathcal{M}_3 - l_g, Q_Z) &= s(l_g + (m+n) \cdot Q - l_g, Q_Z) \\ &= s(m \cdot Q, Q_Z) \cdot s(n \cdot Q, Q_Z) \\ &= s(m \cdot A \cdot \zeta_1, Q_Z) \cdot s(n \cdot B \cdot \zeta_2, Q_Z) \\ &= s(\mathcal{M}_1, Q_1) \cdot s(\mathcal{M}_2, Q_2) \end{aligned} \quad (5)$$

3. PERFORMANCE EVALUATION

The proof generation is evaluated through comparison with the proposed system (PS) and the graph is plotted for 200, 400, 600, 800, and 1,000 blocks. In study [22], for 200 blocks the value is 1.2, study [23] for 200 blocks gives a value of 1.25, study [24] gives a value of 1.4, study [25] gives a value of 1.6, and PS gives a value of 0.010395. In study [22], for 400 blocks the value is 1.7, study [23] for 400 blocks gives a value of 1.8, study [24] gives a value of 2.1, study [25] gives a value of 2.4, and PS gives a value of 0.003271. In [22], for 600 blocks the value is 2.5, study [23] for 600 blocks gives a value of 2.6, study [24] gives a value of 2.8, study [25] gives a value of 3.1, and PS gives a value of 0.002026. In [22], for 800 blocks the value is 3.3, study [23] for 800 blocks gives a value of 3.4, study [24] gives a value of 3.6, study [25] gives a value of 3.8, and PS gives a value of 0.002026. In study [22], for 1,000 blocks the value is 3.9, study [23] for 200 blocks gives a value of 3.95, study [24] gives a value of 4.2, study [25] gives a value of 4.56, and PS gives a value of 0.002098. Figure 2 shows the proof generation comparison.

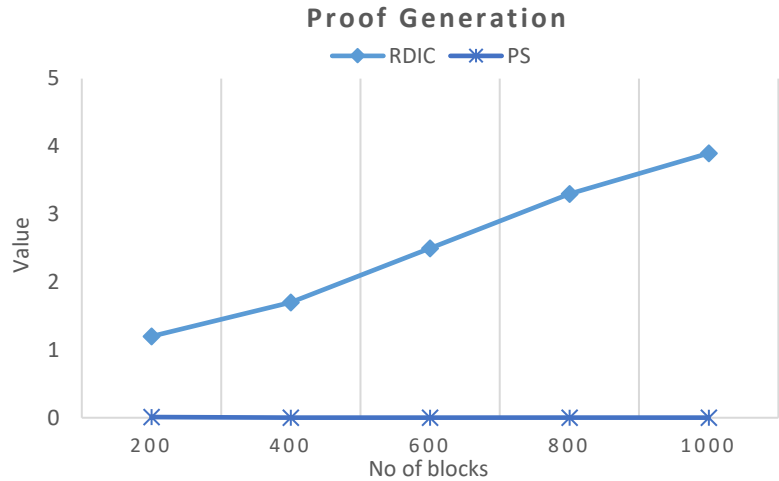


Figure 2. Proof generation comparison

3.1. Verification

Figure 3 presents the verification comparison. The verification mechanism is evaluated here wherein is compared with the proposed system and the graph is plotted for 200, 400, 600, 800, and 1,000 blocks. In study [22], for 200 blocks the value is 2.2, study [23] for 200 blocks gives a value of 2.5, study [24] gives a value of 2.8, study [25] gives a value of 4.1, and PS gives a value of 0.057744. In study [22], for 400 blocks the value is 3.1, study [23] for 400 blocks gives a value of 3.2, study [24] gives a value of 4, study [25] gives a value of 7.5, and PS gives a value of 0.002558. In study [22], for 600 blocks the value is 4.3, study [23] for 600 blocks gives a value of 5, study [24] gives a value of 5.8, study [25] gives a value of 11.2, and PS gives a value of 0.001421. In [22], for 800 blocks the value is 7.2, study [23] for 800 blocks gives a value of 7.4, study [24] gives a value of 8, study [25] gives a value of 13, and PS gives a value of 0.001257. In study [22], for 1,000 blocks the value is 8.8, study [23] for 200 blocks gives a value of 9, study [24] gives a value of 9.5, study [25] gives a value of 15.2, and PS gives a value of 0.001319. Figure 3 shows the verification comparison.

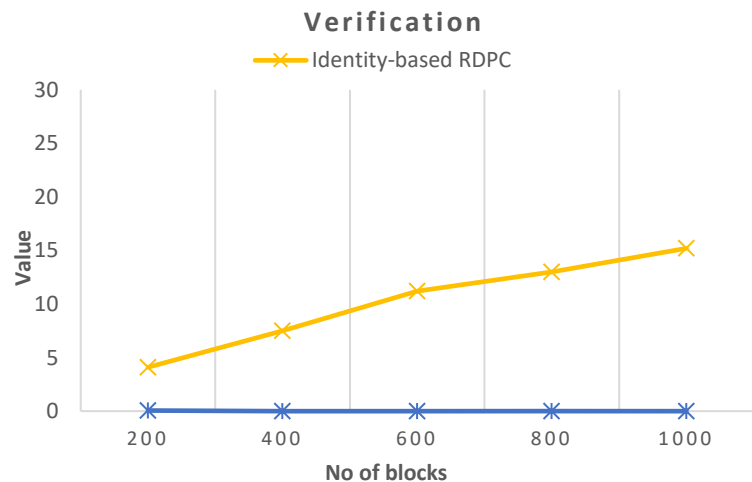


Figure 3. Verification comparison

3.2. Comparative analysis

3.2.1. Proof generation

The comparative analysis is carried out for the proof generation, the PS is compared with the existing system, for 200 blocks the existing system (ES) value is 1.6 and the PS value is given as 0.010395 the improvisation is 19.7422%. For 400 blocks, the ES value is 2.4 and the PS value is given as 0.003271 the

improvisation is 19.9456%. For 600 blocks, the ES value is 3.1 and the PS value is given as 0.00216 the improvisation is 19.9721%. For 800 blocks, the ES value is 3.8 and the PS value is given as 0.002026 the improvisation is 19.9787%. For 1000 blocks, the ES value is 4.56 and the PS value is given as 0.002098 the improvisation is 19.9816%. In the end to conclude that, our PS performs better than the existing system.

3.2.2. Verification

The comparative analysis is carried out for verification and the PS is compared with the existing system, for 200 blocks the ES value is 4.1 and PS value is given as 0.057744 the improvisation is 19.445%. For 400 blocks, the ES value is 7.5 and the PS value is given as 0.002558 the improvisation is 19.9864%. For 600 blocks, the ES value is 11.2 and the PS value is given as 0.001421 the improvisation is 19.949%. For 800 blocks, the ES value is 13, and the PS value is given as 0.001257 the improvisation is 19.961%. For 1,000 blocks, the ES value is 15.2 and the PS value is given as 0.001319 the improvisation is 19.993%. In the end to conclude that, our PS performs better than the existing system.

4. CONCLUSION

A secure cloud computing model has been in demand since the development of a particular computing model, recent developments in technologies and high computation power have 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 develops 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. VPEH 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, recent developments of blockchain and deep learning architecture can be incorporated for further efficiency in terms of the integrity of the data.




REFERENCES

- [1] 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.
- [2] 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, 2023, doi: 10.1109/JSYST.2022.3218894.
- [3] 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.
- [4] L. Xiong, D. Dong, Z. Xia, and X. Chen, "High-capacity reversible data hiding for encrypted multimedia data with somewhat homomorphic encryption," *IEEE Access*, vol. 6, pp. 60635–60644, 2018, doi: 10.1109/ACCESS.2018.2876036.
- [5] 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.
- [6] 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, 2018, doi: 10.1016/j.jnca.2018.01.016.
- [7] 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, 2018, doi: 10.1109/TCC.2017.2649685.
- [8] E. Zaghoul, 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.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.




- [17] 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.
- [18] J. Shen, T. Zhou, D. He, Y. Zhang, X. Sun, and Y. Xiang, "Block design-based key agreement for group data sharing in cloud computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 6, pp. 996–1010, Nov. 2019, doi: 10.1109/TDSC.2017.2725953.
- [19] K. Lee, "Comments on "secure data sharing in cloud computing using revocable-storage identity-based encryption"," *IEEE Transactions on Cloud Computing*, vol. 8, no. 4, pp. 1299–1300, Oct. 2020, doi: 10.1109/TCC.2020.2973623.
- [20] 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.
- [21] J. Wei, W. Liu, and X. Hu, "Secure data sharing in cloud computing using revocable-storage identity-based encryption," *IEEE Transactions on Cloud Computing*, vol. 6, no. 4, pp. 1136–1148, Oct. 2018, doi: 10.1109/TCC.2016.2545668.
- [22] 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.
- [23] 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.
- [24] 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 *2017 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.
- [25] 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 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.