

RESEARCH ARTICLE

OPEN ACCESS

A FRACTAL CODING AND TDES-DIMA BASED FRAMEWORK FOR PRIVACY-PRESERVING BIG DATA STORAGE IN CLOUD ENVIRONMENTS

Rajakumar Ponnumani*¹, Sunilkumar Chellaswamy Santhakumari², Jaydipkumar H Trivedi³,
Tadikonda Venkata Durga Prasad⁴

^{1,2,3,4}Department of Computer Science and Engineering, Parul Institute of Technology, Parul University, Vadodara, Gujarat, India

¹<http://orcid.org/0000-0002-2903-2757>, ²<http://orcid.org/0009-0005-5872-3839>

³<http://orcid.org/0009-0004-5701-5550>, ⁴<http://orcid.org/0009-0001-8468-6366>

E-mail: *prajakumarkk@gmail.com, lsunilmca1981@gmail.com, jaydipkumar.trivedi40927@paruluniversity.ac.in, prasad.tvd@gmail.com

ARTICLE INFO

Article History

Received on: February 26, 2026

Reviewed on: March 26, 2026

Accepted on: July 01, 2026

Published on: July 31, 2026

Keywords:

Fractal Coding,
Dynamic Trusted Authentication,
Structural Information,
Sensitivity Allocation,
Confidentiality.

ABSTRACT

The rapid expansion of the big data in cloud facilities has posed immense challenges in terms of storage efficiency, privacy as well as security. To address these issues, this study proposes a Fractal Coding-based Privacy-Preserving Framework with a dynamic trusted authentication architecture to handle big data in cloud systems safely. First the framework divides raw information into blocks and encodes them to small fractal codes minimizing overhead of storage and transmission yet preserving the structure of the information. Then the privacy levels of each blocks are determined by the sensitivity allocation mechanism and much protection is guaranteed on highly sensitive data through assigning the privacy level to the respective divided block. Finally, to strengthen confidentiality, Triple Data Encryption Standard (TDES)-Dynamic Intelligent Multiple Access (DIMA) module added which engages extended TDES encryption together with dynamic authentication to withhold brute force attack and unauthenticated access. The combination of these design balances compression efficiency, the preservation of privacy, and cloud security. Experimental analysis shows that the proposed framework provides a faster decryption, a high transmission efficiency, high data confidentiality of 95% and minimum CPU consumption of 24%.



Copyright ©2026 by authors and Galileo Institute of Technology and Education of the Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

I. INTRODUCTION

Cloud storage has become the default storage and processing location to hold and operate large datasets due to its easy scaling and reduced initial cost of infrastructure. With the shift of the routine reporting to sophisticated analytics, Machine Learning (ML), and real-time decision systems, the amount and diversity of information stored on clouds had to increase manifold as sensor streams and medical records gave way to multimedia and transaction logs [1], [2]. These expansions put new strains on the manner in which data are stored, accessed and secured necessitating to reconsider the efficiency of storage, privacy guarantees and end to end protection in the clouds [3], [4]. To efficiently store large scale data in the cloud, it is needed to deploy algorithmic compression, smart redundancy, and storage aware data placement together [5]. Conventional ways like deduplication, erasure coding and sharding are predominant in minimizing physical footprint and enhancing durability, but it interacts badly with privacy controls and introduces nontrivial computational and network overhead in the event it operates at web scale. The recent work thus explored more adaptive redundancy mechanisms and erasure-code optimizations to reduce storage overheads without impacting on availability and repairability [6]. Privacy and security are now first-class issues instead of subsequently considered. End-to-end data encryption preserves confidentiality but complicates analytics and deduplication; fully homomorphic cryptographic techniques have been proposed to perform computations on ciphertexts, yet are too expensive to scale with large deployments [7]; and lightweight or certificateless cryptographic designs have been suggested to provide a secure IoT-to-cloud pipeline, but are too expensive to scale to large deployments.

The literature of recent times [8] provides a clear picture of these trade-offs between confidentiality, computability, and the usage of resources and inspire hybrid solutions that decouple data utility and secrecy. The latest optimization and adaptive partitioning techniques have rendered the fractal and self-similarity-based schemes feasible in large heterogeneous data sets [9], and exhibited encouraging enhancement in the ratio of compression and encode/decode performance on real data. These advances indicate that fractal coding can be considered an effective part of a privacy-aware storage pipeline together with an effective parameter encoding and targeted fidelity control. Several recent studies tackled secure cloud storage holistically by some combined homomorphic or searchable encryption with storage verification primitives; others proposed dynamic access control or lightweight authenticators to support data dynamics and audits with low preprocessing costs [10]. These works highlighted the need for storage designs that are simultaneously compression-aware, privacy-driven, and verifiable at scale. Despite these advances, important gaps remain. Many storage schemes either prioritize compression and efficiency at the expense of adaptive privacy controls. Recent redundancy and erasure-code optimizations addressed availability, but did not simultaneously solve sensitivity-aware compression and efficient key management across distributed cloud nodes. Motivated by this gap, the proposed model combines three major concepts: (1) fractal compression (2) a sensitivity allocation layer and (3) TDES-DIMA security layer that provides robust, session-bound access control and layered encryption for high-sensitivity content. At the time of data request dynamic authentication, preventing brute-force attacks and unauthorized access. The contribution of the proposed framework was:

- To remove anomalies, redundancy, and noises in the data, the proposed model uses the partitioning and preprocessing step that maximizes the uniformity of the data, which increases the efficiency of compression and general computational performance.
- To improve storage efficiency without reducing the reconstruction quality the proposed model combines the fractal geometric transformations to achieve the efficient data compression the use of the self-similarity.
- To offer the adaptive privacy preservation, the model under consideration allocates varying levels of sensitivity to data that are based on the contextual and content-specific features of data.
- To resolve both confidentiality of data and user authentication, a new privacy-preserving model based on TDES-DIMA was created to allow big data storage and retrieval in clouds securely.
- To improve scalability, redundancy, and consistency without compromising access controls and efficient retrieval, the proposed model uses distributed cloud storage.

The rest of this paper was arranged in the following way: Section 2 is a review of the privacy preservation methods existing in big data in cloud infrastructures. Section 3 explains the proposed TDES-DIMA framework in great detail and is followed by Section 4 which explains the performance evaluation of the proposed model. Lastly, the study is concluded and the future research directions are outlined in the Section 5.

II. LITERATURE REVIEW

In [11], a partition-based distributed big data model to store personnel information in a cloud environment was proposed, which incorporates ChaCha20 encryption, Dual Stage Data Partitioning (DSDP), and k-anonymization to increase the privacy and confidentiality of the information. The model spreads data through several cloud nodes by using the hash of SHA-512 which is more secure, efficient and fault tolerant than the traditional methods. Nevertheless, it did not have dynamic trust negotiation and adaptive anonymization protocols to deal with various data sensitivity and decentralized key management. The Hierarchical Federated Learning (HFL) approach is suggested by the author of [12] to enhance the scalability, computational efficiency, and privacy of healthcare data analysis. Tests on real and artificial medical data revealed that the HFL framework attained an accuracy of 97% of the model with the minimized privacy leakage. However, limitations variations in performance with data characteristics, and increased computational costs were identified. A SecPri-BGMPOP (Security and Privacy of BoostGraph Convolutional Network-Pinpointing-Optimization Performance), a multi-step architecture that was constructed in [13] to improve the security and privacy of cloud-based big data systems. The model integrated BG Convolutional Network Clustering (BGCNC) to minimize computational complexity and employed pinpointing-based encryption to mitigate key leakage risks. Also, the optimization algorithm was used to optimize the generation of encryption keys to enhance better accuracy and efficiency. The model, however, should be able to speed encryption and decryption to enable its use in large applications. One of the studies [14] suggested a scalable method of big data multidimensional anonymization through MapReduce. The main concept was to break down the dataset recursively into small portions using MapReduce until all the partitions had fit in the memory of a computing node. It came up with a tree indexing structure so that it could compute recursively. Moreover, the analysis revealed the relevance of the method to differential privacy.

The experimental findings demonstrated that the suggested approach enhanced the scalability significantly. The study [15] suggested the lightweight matrix factorization framework or personalized recommendations, in which shard gradient training on user IoT device facilitated the local learning of the model without data exposure. A two-phase design guaranteed the security of the data and dimensionality reduction and a stabilization mechanism was used to reduce the gradient perturbation. In experimental assessments, there was greater accuracy, efficiency and privacy than the existing measures. On the same note, the authors of [16] presented a user-customized model of differential privacy based on the level of trust among the users, which is measured by the density of the community and implemented by controlled randomized noise. A noise correlation decoupling mechanism based on Markov and a blockchain-powered community modeling further avoided linkage and poisoning attacks and ensured a better privacy protection. The research [17] offered the TDES technique to ensure high-security levels in the analysis of extensive data. Implementation of TDES improved security of data transmission and storage to avoid the unauthorized access and data leakage. It was able to trade off the strength of encryption with the complexity of computations by using simulation-based tests on large datasets. The study [18], has also put forward an effective and dynamic verification scheme on cloud based Industrial Internet platform (IIP). It was designed using a homomorphic authenticator that was very lightweight and an authenticator accumulator and reduced the computational overhead during preprocessing of the data.

The accumulator of the authenticator was enhanced with dynamic information so that it can make safe changes and updates in the data. In [19], a Two-Step Authentication Inner Product Functional Encryption (TSA-IPFE) scheme was suggested to a three-tier cloud-edge-smart device architecture. The model allowed the dynamic management of authorizations by a two-step mechanism of identity-matching and the efficient revocation of the computational authorizations of migration during device migration. The semantic security, anonymity, and increased efficiency of the simulation proved the superiority over the current algorithms. In [20] presented a TDES-based approach to improve the confidentiality and integrity of the large healthcare information stored in the cloud. The above cloud storage was offered a secured read-write operation and was not susceptible to unauthorized access because it encrypted patient-related data before cloud storage. Its effectiveness in protecting healthcare data was validated with the help of the experiment. The model however had a problem of long computational time.

II.1 RESEARCH GAP

The literature review indicates that there are still unresolved issues in guaranteeing efficient, secure and privacy-conscious big data management in cloud setups. The currently available models including partition-based storage [11], HFL [12], and multi-stage encryptions frameworks [13] have enhanced confidentiality, scale and computation efficiency but remain to be characterized with significant weaknesses. These are lack of dynamically negotiated trust, adaptive anonymization and decentralized key management schemes alongside larger computational and encryption costs of large-scale or heterogeneous data. Further, the conventional encryption schemes [20] are still computationally demanding which implies that there is a necessity of a less heavyweight, more adaptable and distributed security system to balance privacy, efficiency and scalability of the various cloud-based big data uses.

III. PROPOSED FRAMEWORK OVERVIEW

In this section the proposed model Fractal Coding, TDES-DIMA based framework of the big data in cloud is presented. The first step of the model splits the raw heterogeneous data into smaller blocks in order to be preprocessed and normalized. The blocks are then coded using fractal coding method, which takes advantage of the self-similarity of data in order to obtain a high compression with minimum loss. The sensitivity allocation mechanism is built in to allocate levels of privacy depending on the importance of the data. In order to maintain confidentiality, the TDES-DIMA module uses extended TDES encryption and dynamic authentication in order to ensure access and avoid illegal use. Lastly, the stored and encrypted information is retrieved and stored on distributed cloud systems.

Phase 1: Data Ingestion

In order to make the simulation closer to real-world a huge heterogeneous data were simulated in the form of a dataset, which recreates multi source data properties such as time-series sensor measurements, medical imaging data fragments and enterprise transaction data. This is a simulated data that makes it scalable and also variable so as to accommodate both structured and unstructured data. The ingested dataset is represented as:

$$D = \{d_1, d_2, d_3, \dots d_i\} \quad (1)$$

Here d_i is a raw data that has values obtained by a specific source. The heterogeneous data is prepared to further handle the framework.

Phase 2: Partitioning and Preprocessing

After the ingestion of the dataset into the cloud gateway, it passes through the stage of partitioning and preprocessing to be analyzed further. The large data D is broken into smaller data blocks that can be managed in terms of source type, data format, or sampling rate. The metadata linked to every block is Source ID S_i , Timestamp T_i , and Consent Flag C_i as presented in the Figure 1. This association is expressed as follows:

$$M_i = \{S_i, T_i, C_i\} \quad (2)$$

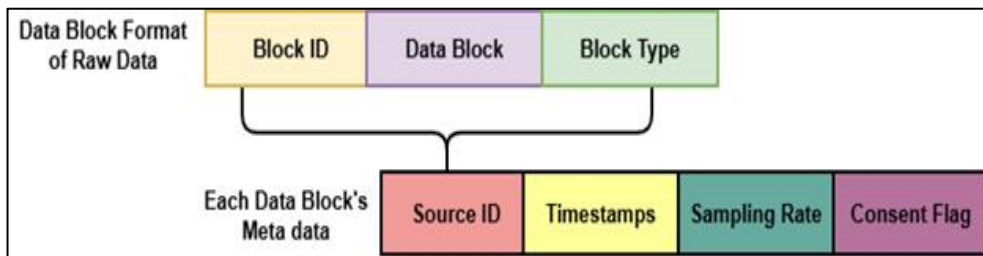


Figure 1: Data Block Format.
Source: Authors, (2026).

The cleaning, normalization and noise removal processes of each block b_i are performed:

$$b_i = f_{norm}(f_{clean}(d_i)) \quad (3)$$

In order to ensure the consistency of the heterogeneous data, z-score normalization is used:

$$x' = \frac{x - \mu}{\sigma} \quad (4)$$

In this case, x is the raw value, μ is the mean, and σ is the standard deviation of the data set. In preprocessing, unnecessary, corrupt, or missing records may be eliminated and the data are normalized so that, it is consistent across various modalities including images, waveforms, tabular records and text blocks. This makes each block of data structurally coherent, and also makes metadata integrity.

Phase 3: Fractal Coding

Here, the processed and segmented data blocks are broken down into fractal coded blocks, which is a mathematical compression algorithm which exploits the self-similarity in data to reduce redundancy and realize high compression efficiency. The main aim of this stage is to symbolize every data block in terms of fractal transformations, which result in a small size of storage, but do not compromise any important structural and statistical data. All preprocessed blocks b_i are subdivided into two categories, range blocks (r) and domain blocks (db). The process identifies a transformation T_i that best maps a domain block db_j to a range block r_i through geometric and intensity adjustments. This mapping relation is stated to be:

$$r_i = T_i(db_j) \quad (5)$$

The transformation T_i uses an affine transformation, scale and offset of the intensity levels of the data block to provide optimal similarity. Mathematically, this is expressed as:

$$T_i(x) = \alpha_i x + \beta_i \quad (6)$$

In this case, the scaling factor of contrast, α_i , and the offset of a data block, β_i . These two values are computed by minimizing the reconstruction error between the original and transformed block in terms of Mean Squared Error (MSE) criterion:

$$E = \min_{T_i} \sum_{x \in r_i} (r_i(x) - T_i(db_j(x)))^2 \quad (7)$$

When the optimum transformation parameters have been generated, it gets stored as a fractal code that represents the block of data. The parameters in each of the codes are: α_i, β_i the domain block ID db_j and E the corresponding error metric.

$$f_i = \{\alpha_i, \beta_i, db_j, E\} \quad (8)$$

These space-efficient fractal codes are stored instead of the original data, and this helps to save a lot of storage space since only these compact codes are stored.

Phase 4: Sensitivity Allocation

At this stage, the fractal-coded blocks that were extracted in the step above are processed and categorized according to their sensitivity level in order to create a privacy-conscious dataset. The objective of this stage is to make sure that data blocks that entail sensitive data are given more robust protection mechanisms. The f_i fractal-coded block is rated according to its sensitivity level (sl_i) which, depending on the metadata and sensitivity of the content of the context (e.g., medical identifiers, financial information):

$$sl_i = f_{priv}(S_i, C_i, D_i) \quad (9)$$

Depending on the calculated sl_i , the fractal-coded block is classified into one of three categories:

- ❖ High-sensitive (H)- Blocks of data that include critical or personally identifiable information (e.g. medical identifiers, financial attributes). These blocks need high level of encryption and high-fidelity reconstruction to maintain the integrity of data.
- ❖ Medium-sensitive (M) – Medium significance blocks that are the subject of balanced protection with mild compression that can be accepted without intrusion to privacy.
- ❖ Low-sensitive (L) – Blocks that hold non-crucial data that can be stored or recreated roughly, so that lossy compression can be used to achieve best storage and computational efficiency.

After identifying the sensitivity class of a given block, all the fractal-coded blocks are labeled with the sensitivity level, creating privacy-aware representation of:

$$tag_i = \{f_i, sl_i\} \quad (10)$$

The sensitivity-based tagging tag_i brings about a change in encryption and reconstruction fidelity during later stages.

Phase 5: Security with TDES-DIMA and Storing in Cloud

This stage is used to make sure that data confidentiality and integrity is upheld especially when dealing with high-sensitive fractal-coded blocks detected in the former stage. The stage involves combination of Triple DES and DIMA mechanism to ensure protection of sensitive data against unauthorized access and identity spoofing at the same time maintaining efficient retrieval and verification. All highly sensitive block f_i are encrypted by TDES algorithm that uses three consecutive encryptions and decryptions with different keys (k_1, k_2, k_3). The ciphertext block c_i is obtained by encrypting the plaintext block p_i using the following format:

$$c_i = e_{k_3}(d_{k_2}(e_{k_1}(p_i))) \quad (11)$$

Here e and d denote encryption and decryption operations, respectively. This multiple level encryption provides high levels of cryptographic robustness because exploiting one key will not give the whole dataset. The process of decryption restores the transformation to retrieve the original plaintext which is defined as:

$$p_i = d_{k_1}(e_{k_2}(d_{k_3}(c_i))) \quad (12)$$

The DIMA mechanism presents a dynamic authentication mechanism which identifies users and devices on a session basis. Each access attempt is authenticated with a representation of the formulation based on hash: a session-specific authentication token is produced.

$$A_{token} = h(Id_u \oplus T_s \oplus R_n) \quad (13)$$

In this case, the identity of the unique user or device, a timestamp to verify freshness of the session, and a random nonce to prevent predictability are denoted as Id_u , T_s and R_n respectively. The $h(.)$ is a safe hash function that generates a special token on each access instance, and therefore offers time-limited and identity-based access control. This is done to stop replaying and impersonation attacks through this mechanism of continuously updating the session tokens with time and random variations. The general structure of the proposed model is presented in figure 2.

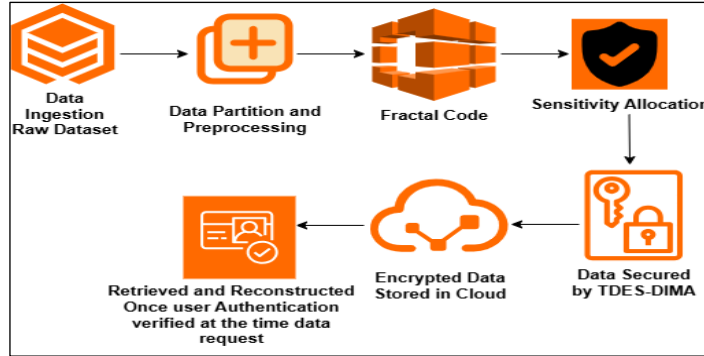


Figure 2: Overall Architecture Diagram of Suggested TDES-DIMA.

Source: Authors, (2026).

The privacy-tagged and securely encrypted data is sent to the distributed cloud storage facility of scalable hosting, high availability, as well as fault tolerance. The distributed architecture also provides greater reliability and redundancy of the data stored in the system such that, when a node fails, the data can be retrieved out of the replicas. The redundancy and reliability (R_d) of the stored dataset is given as:

$$R_d = \frac{n_{rep}}{n_{tot}} \quad (14)$$

Here n_{rep} is the replication of data blocks and n_{tot} is the sum of all the stored blocks. This makes sure that an adequate data replication feature is replicated at the cloud infrastructure. As a result, the distributed storage system has a secure, scalable, and fault-tolerant data management.

Phase 6: Retrieval & Reconstruction

At this last stage, the retrieval and reconstruction procedures will start with a legitimate user placing a data access request to the cloud system. First, DIMA mechanism authenticates the user by session based authentication tokens to confirm authorized access. The system then verifies that the content is correct and then decrypts the sensitive blocks of data with the help of the TDES decryption routine as previously described and gets the original encrypted data. The blocks of the fractal-coded data are then decrypted and reassembled using inverse fractal transformations, restoring the dataset with fidelity levels matching the sensitivity tag of each block, a high sensitive block is reassembled with its full precision whereas a low sensitive block can be reassembled with approximations to achieve more efficiency. The general reconstruction procedure can be formulated as:

$$\hat{d} = f_{recon}(d_{dec}, sl_i) \quad (15)$$

Here $\hat{d}$ denotes the reconstructed dataset, d_{dec} represents the decrypted fractal-coded blocks, and sl_i indicates the sensitivity level of each block guiding the reconstruction fidelity. This phase ensures that the final output is a securely reconstructed dataset, preserving both data privacy and analytical integrity. Where $\hat{d}$ signifies the reconstructed data, d_{dec} signifies decrypted fractal-coded blocks, and sl_i signifies the sensitivity degree of every block that determines the reconstruction fidelity. In this stage, the data is securely reconstructed and both data privacy and data analytical integrity are maintained in the final output.

IV. RESULTS AND DISCUSSIONS

The proposed model enhances security of big data on a cloud environment with Python software in a system with 6GB of a graphics card, 16GB of RAM, i7 core processor, and windows 10 OS. This section describes the comparison of the proposed method and the existing methods. The proposed TDES-DIMA model was compared with throughput of the current SecPri-BGMPOP [13] and Standard TDES as Figure 3. The proposed model showed that in 75 blocks, it had 105 ms, which was above the SecPri-BGMPOP (45 ms), and Standard TDES (80 ms).

With the block count of 300, the throughput was 118 ms, which was 84% and 36% higher than the SecPri-BGMPOP (64 ms) and Standard TDES (87 ms), respectively. The proposed model had a better performance at the highest block size, 525, with 140 ms, which is 72.8% better than SecPri-BGMPOP (81 ms) and 12 times better than Standard TDES (125 ms).

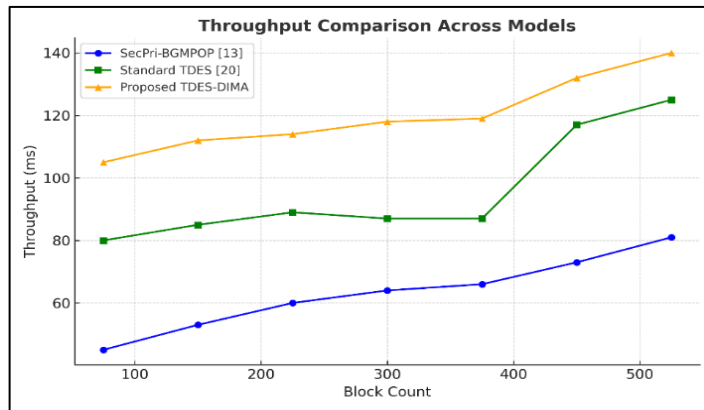


Figure 3: Throughput Analysis.

Source: Authors, (2026).

SecPri-BGMPOP model despite being effective in preserving privacy had disadvantages in that it was slow when encrypting and decrypting large data thus its application was limited in scaling. On the same note, the Standard TDES method had considerable delays because of its abundance of computational time. On the contrary, the proposed model of TDES-DIMA has addressed these disadvantages by incorporating DIMA which made it simpler to access and minimize key usage when encrypting. This mechanism minimized redundant calculations and thus enhance data processing efficiency and hence increase in throughput especially during heavy work loads.

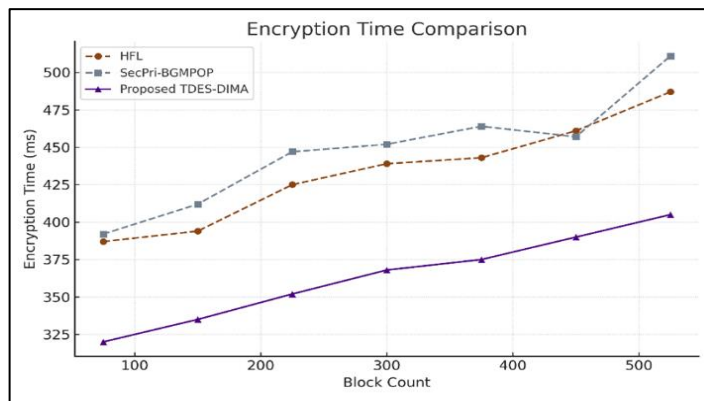


Figure 4: Encryption Time.

Source: Authors, (2026).

It is evident in the comparative analysis between encryption and decryption time that HFL [12], SecPri-BGMPOP, and the proposed model of TDES-DIMA have different differences in computational efficiency. Based on the Figure 4 the HFL framework showed moderate encryption performance, with a base of 387 ms and 487 ms at 75 and 525 blocks, respectively, and is mainly caused by the hierarchical data aggregation process which, although enhances scalability and privacy, introduces latency as a result of the multiple communication layers.

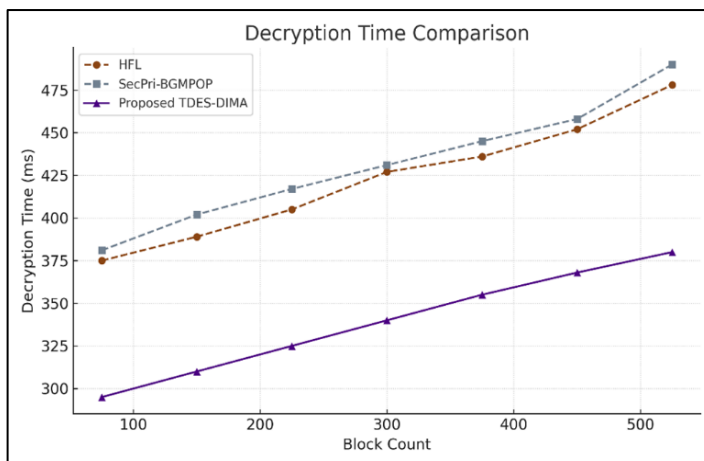


Figure 5: Decryption Time.

Source: Authors, (2026).

Equally, the SecPri-BGMPOP model had a slightly longer encryption time, with a range of 392 ms to 511 ms since the multi-stage encryption events combined with pinpointing-based key generation incurred some extra overhead. This is consistent with its well-established weakness of taking acceleration to encrypt and decrypt messages. By contrast, the proposed TDES-DIMA had a more stable encryption time, ranging between 318 and 409 ms, and decryption time, between 298 and 379, as shown in the Figure 5. This was achieved because of the effective management of sensitivity-based data prioritization that allows faster encryption and decryption of large data without reducing the security and proves the suggested model to be a good balance in the computational workload and privacy conservation where the previous framework had a struggle to balance the increased latency and resource utilization. Figure 6 shows the latency test of the proposed model when compared to the current frameworks. The latency was found to be between 260 ms and 350ms with increasing block count in the HFL model, largely due to its data aggregation layering and federation coordination schemes where frequent synchronization across the distributed nodes was necessary leading to communication bottlenecks. Equally, the model SecPri-BGMPOP had latency levels of between 245 ms and 340 ms which is largely attributed to its sequential encryption and clustering algorithm and pinpointing based optimization; which added more computational burden to the process of intermediate data exchanges.

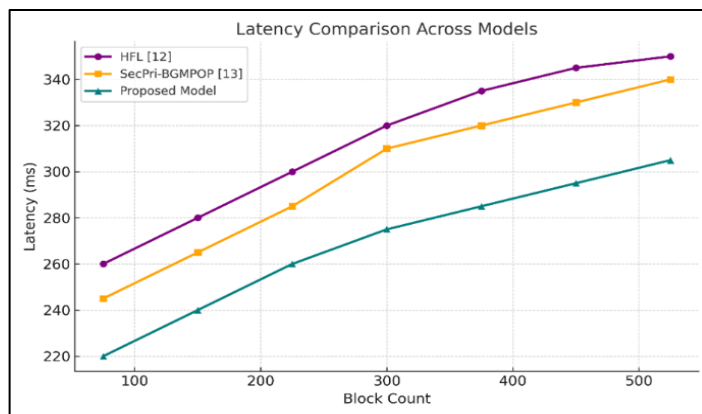


Figure 6: Latency Analysis.
Source: Authors, (2026).

Conversely, the proposed model recorded lower latency of 220 ms to 305 ms with the encapsulated data fractal-based encoding and streamlined block handling system which reduced the overhead of repetitive computation and data transfer. This integration also created a faster response time as the adaptive processing flow was integrated and capable of supporting the high-volume operations of data in cloud-based systems and with minimal delays. Figure 7 presents the relative comparison of data confidentiality and CPU use of DSDP [11], HFL, and the proposed model. The DSDP model had a confidentiality rate of 81% and a CPU usage of 32 and this was mainly attributed to its fixed anonymization and lack of dynamically negotiated trust which restricted its versatility in negotiating heterogeneous and sensitivity-varying data. HFL framework increased the confidentiality to 89% with a minor decrease of CPU usage to 29% but the performance of HFL varied with changes in data distribution and feature complexity resulting in varying privacy levels between datasets.

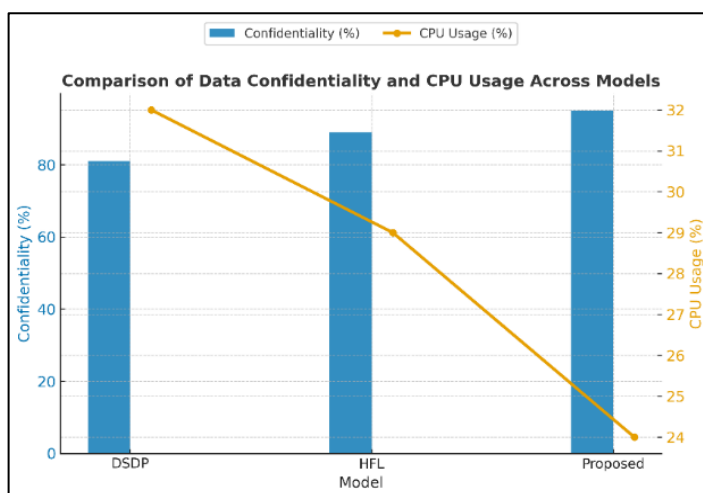


Figure 7: Confidentiality Analysis.
Source: Authors, (2026).

By contrast, the proposed TDES-DIMA framework reached the highest confidentiality of 95% and the lowest CPU consumption of 24% which is 17.3% improvement in confidentiality and 25% decrease in CPU consumption as compared to DSDP. The suggested model had minimal CPU consumption because of optimized fractal coding mechanism that compresses and encodes blocks of data efficiently which minimizes redundant calculations when encryption and retrieval is taking place. The secrecy was due to the dynamic trusted authentication process which highly verifies user identities prior to providing access so that only authorized users can decrypt and manipulate sensitive data.

V. CONCLUSIONS

This study proposed a Fractal Coding based Privacy- Preserving Framework to manage big data safely and efficiently in cloud computing. The framework was useful to minimize the computational cost, in addition to enhancing data confidentiality and security, through fractal coding of compact data representation, sensitivity-based privacy distribution, and TDES-DIMA module of dynamic encryption and authentication. The experimental findings attested that the proposed model attained the best confidentiality of 95% and lowest CPU consumption of 24% indicating the best performance among existing methods. Fractal based compression strategies and sensitivity-based encryption strategies also helped to enhance throughput, latency and better use of storage. To work with hybrid cloud-edge environments in the future, the framework could be expanded to address the latency reduction even more and increase the scalability. The inclusion of quantum-resistant cryptography and AI-assisted adaptive trust should also result in increased resilience to new cyber-threats.

VI. AUTHOR'S CONTRIBUTION

Conceptualization: Rajakumar Ponnumani, Sunilkumar Chellaswamy Santhakumari, Jaydipkumar H Trivedi and Tadikonda Venkata Durga Prasad.

Methodology: Jaydipkumar H Trivedi and Tadikonda Venkata Durga Prasad.

Investigation: Rajakumar Ponnumani, Sunilkumar Chellaswamy Santhakumari, Jaydipkumar H Trivedi.

Discussion of results: Rajakumar Ponnumani, Sunilkumar Chellaswamy Santhakumari, Jaydipkumar H Trivedi.

Writing – Original Draft: Rajakumar Ponnumani.

Writing – Review and Editing: Rajakumar Ponnumani, Sunilkumar Chellaswamy Santhakumari.

Resources: Rajakumar Ponnumani.

Supervision: Rajakumar Ponnumani and Sunilkumar Chellaswamy Santhakumari.

Approval of the final text: Rajakumar Ponnumani, Sunilkumar Chellaswamy Santhakumari, Jaydipkumar H Trivedi and Tadikonda Venkata Durga Prasad.

VII. ACKNOWLEDGMENTS

The author would like to appreciate the effort of the editors and reviewers.

VIII. REFERENCES

- [1] M. El Moudni and E. Ziyati, "Advances and challenges in cloud data storage security: A systematic review," *Int. J. Saf. Secur. Eng.*, vol. 15, no. 4, pp. 653–675, 2025. doi: 10.18280/ijssse.150403.
- [2] M. Pathak, K. N. Mishra, and S. P. Singh, "Securing data and preserving privacy in cloud IoT-based technologies: An analysis of assessing threats and developing effective safeguard," *Artif. Intell. Rev.*, vol. 57, p. 269, 2024. doi: 10.1007/s10462-024-10908-x.
- [3] M. Z. Khan, M. Shoaib, M. S. Husain, et al., "Enhanced mechanism to prioritize cloud data privacy factors using AHP and TOPSIS: A hybrid approach," *J. Cloud Comput.*, vol. 13, p. 42, 2024. doi: 10.1186/s13677-024-00606-y.
- [4] J. Makadia and P. Patel, "Review on cryptographic techniques for privacy-preserving data sharing in cloud," 2025. doi: 10.2139/ssrn.5360584.
- [5] C. Ramesh, B. Srinivasulu, M. S. Babu, et al., "An effective data replication strategy and the improvement of the storage environment," in *Intelligent Systems and Sustainable Computing, ICISSC 2024, Smart Innovation, Systems and Technologies*, vol. 417, Singapore: Springer, 2025. doi: 10.1007/978-981-97-8355-7_32.
- [6] M. Cui and Y. Wang, "EC-Kad: An efficient data redundancy scheme for cloud storage," *Electronics*, vol. 14, no. 9, p. 1700, 2025. doi: 10.3390/electronics14091700.
- [7] R. Hu, C. He, S. Liu, D. Yao, and X. Li, "A fully homomorphic encryption scheme suitable for ciphertext retrieval," *Comput. Mater. Continua*, vol. 84, no. 1, pp. 937–956, 2025. doi: 10.32604/cmc.2025.062542.
- [8] H. Guo, X. Li, and H. Qin, "A lightweight and secure certificateless encryption scheme for smart city IoT security," *Discov. Comput.*, vol. 28, p. 104, 2025. doi: 10.1007/s10791-025-09626-x.
- [9] M. Li and U. K. Tak, "An enhanced fractal image compression algorithm based on adaptive non-uniform rectangular partition," *Electronics*, vol. 14, no. 13, p. 2550, 2025. doi: 10.3390/electronics14132550.
- [10] M. S., K. S. K., T. V., G. S., and D. S., "Dynamic access control through cryptography in cloud," *ITM Web Conf.*, vol. 56, p. 06001, 2023. doi: 10.1051/itmconf/20235606001.
- [11] T. Ting and M. Li, "Enhanced secure storage and data privacy management system for big data based on multilayer model," *Sci. Rep.*, vol. 15, p. 32285, 2025. doi: 10.1038/s41598-025-16624-y.
- [12] J. P. Singh, A. Aqsa, I. Ghani, R. Sonani, and V. Govindarajan, "Privacy-aware hierarchical federated learning in healthcare: Integrating differential privacy and secure multi-party computation," *Future Internet*, vol. 17, no. 8, p. 345, 2025. doi: 10.3390/fi17080345.
- [13] M. Kuttiyappan, J. P. Appadurai, B. P. Kavim, et al., "Big data privacy protection and security provisions of the healthcare SecPri-BGMPOP method in a cloud environment," *Mathematics*, vol. 12, no. 13, p. 1969, 2024. doi: 10.3390/math12131969.
- [14] X. Zhang, et al., "MRMondrian: Scalable multidimensional anonymisation for big data privacy preservation," *IEEE Trans. Big Data*, vol. 8, no. 1, pp. 125–139, 2022. doi: 10.1109/TBDATA.2017.2787661.
- [15] H. Zhou, G. Yang, Y. Xiang, Y. Bai, and W. Wang, "A lightweight matrix factorization for recommendation with local differential privacy in big data," *IEEE Trans. Big Data*, vol. 9, no. 1, pp. 160–173, 2023. doi: 10.1109/TBDATA.2021.3139125.
- [16] Y. Qu, et al., "Towards privacy-aware and trustworthy data sharing using blockchain for edge intelligence," *Big Data Min. Anal.*, vol. 6, no. 4, pp. 443–464, 2023. doi: 10.26599/BDMA.2023.9020012.

- [17] H. Byeon, M. Evin, A. U. Mirza, and S. Murugesan, "Triple data encryption standard based secured and privacy preserved big data analytic," AIP Conf. Proc., vol. 3306, p. 050009, 2025. doi: 10.1063/5.0275945.
- [18] H. Yu, H. Zhang, Z. Yang, and S. Yu, "Edasvic: Enabling efficient and dynamic storage verification for clouds of industrial Internet platforms," IEEE Trans. Inf. Forensics Secur., vol. 19, pp. 6896–6909, 2024. doi: 10.1109/TIFS.2024.3422790.
- [19] H. Yang and C. Peng, "TSA-IPFE: A secure solution for authorization revocation and dynamic identity assignment in IoT," IEEE Internet Things J., vol. 12, no. 12, pp. 20288–20300, 2025. doi: 10.1109/JIOT.2025.3544641.
- [20] M. N. Ramachandra, M. S. Rao, W. C. Lai, B. D. Parameshachari, J. A. Babu, and K. L. Hemalatha, "An efficient and secure big data storage in cloud environment by using triple data encryption standard," Big Data Cogn. Comput., vol. 6, no. 4, p. 101, 2022. doi: 10.3390/bdcc6040101.