A Multi-User Blockchain Framework Integrated With Disentangled Graph Variational Autoencoder for Data Integrity Assurance in Heterogeneous Cloud Environments

Mohd Anwar Ali[1], Nagesh Vadaparthi[2], L Sumalatha[3]

[1]Research Scholar, JNTUK Kakinada, Kakinada, Andhra Pradesh, India, dr.mdanwarali123@gmail.com
[2]Professor, MVGR College of Engineering, Vizianagaram, Andhra Pradesh, India, itsnageshv@gmail.com
[3]Professor, JNTUK Kakinada, Kakinada, Andhra Pradesh, India, lsumalatha@jntucek.ac.in

Abstract
Heterogeneous cloud environments integrate a variety of platforms, services and data formats for multiple applications and multiple user access. In these systems, maintaining data integrity is crucial for accuracy, consistency, and preventing unauthorized modifications, particularly when data is accessed and stored across various locations. Challenges include detecting data tampering, enforcing access control, and managing heterogeneous data types, highlighting the need for scalable solutions to ensure cloud data integrity. This paper proposes a Multi-User Blockchain Framework with a Disentangled Graph Variational Autoencoder for Enhancing Integrity in Heterogeneous Cloud Data (MUBF-DGVAE). The framework has three phases. In data acquisition, heterogeneous cloud data is processed using a Polynomial Non-Linear Chaotic Function (PNLCF) to create unique hash fingerprints for integrity verification. During encryption and storage, Quantum Hash-Based Attribute-Based Encryption (QHABE) is applied per a defined access policy; encrypted data and the quantum hash are stored in the cloud. A blockchain transaction with the encrypted data URL, initial hash key, timestamp, and access policy is validated and recorded using Non-Interactive Practical Proof-of-Storage (nPPoS). In verification, when a user requests access, Multi-Qubit Quantum Key Distribution Ciphertext-Policy Attribute-Based Encryption (QKDCPABE) generates decryption keys by validating user attributes against the access policy. After decryption, PNLCF regenerates a hash from decrypted data, compared to the original blockchain-stored hash by the Disentangled Graph Variational Autoencoder (DGVAE) to detect tampering. Experimental evaluations show MUBF-DGVAE achieves a stable hash bit change of 11.9–14.8 ms, a hash time of 0.011–0.080 ms, a cloud encryption runtime of 124.47 ms at 100 users, and a decryption runtime of 83.54 ms at 100 users, outperforming methods like EI-MUBF-HSCM, EPQKG-MUAED-EHR, and ARA-MUCIC-ME across all metrics..

Keyword: Blockchain, Cloud Data Integrity, Disentangled Graph Variational Autoencoder, Heterogeneous Cloud, Multi-User Access Control, Non-Interactive Proof-of-Storage, Quantum Attribute-Based Encryption, Quantum Key Distribution.

I. Introduction

Cloud computing has changed the way IT resources are delivered and consumed by providing access to shared computing resources on demand via the Internet [1]. Cloud services, working under a pay-per-use pricing model as provided by the major service providers have sped up the move of real time applications and enterprise data to cloud services, dramatically cutting the capital expenditures [2]. But with the changes in cloud network architecture and communication technologies, the traditional security apparatus has become increasingly ineffective, especially in multi-tenant environments that have diverse architectures and services [3]. The classical cryptographic methods are based on mathematical foundations and presumed computational intractability and are thus fundamentally susceptible to quantum computer advances and are not well suited to the changing access patterns of today's cloud environments [4], [5].

Data integrity and secure multi-user access management are significant security problems in heterogeneous cloud environments. Concurrently, the interaction of multiple data sources, multiple distributed storage nodes, and multiple users significantly increases the potential for data tampering, data integrity and data access issues. Fine-grained access control was also proposed to be resolved using Attribute Based Encryption (ABE) frameworks, but most of the existing solutions do not meet the requirements of complex, location-aware, and cross-platform cloud attributes [8]. While traditional security solutions may be limited by their inability to respond comprehensively to these challenges, new paradigms like blockchain-based immutable record keeping and intelligent graph-based anomaly detection offer a more comprehensive solution [9, 10].

In view of these challenges, this paper presents a new Multi-User Blockchain Framework integrated with a Disentangled Graph Variational Autoencoder (MUBF-DGVAE) to tackle the issue of data-integrity in heterogeneous cloud environments. The framework is designed for data storage that is accessible only through data owners and for fine-grained access control using attribute-based cryptographic primitives that are resistant to quantum computers.The

framework is engineered to allow for fine-grained access control, using attribute-based cryptographic primitives resistant to quantum computers, and for data storage that can only be accessed by data owners, with data accessible after the data owner has added it, but not before. By combining DGVAE, the disentangled latent representations of re-hashed decrypted data can be compared to the original hash stored on the blockchain, which allows for accurate identification of tampering and corruption in the data. The result is a scalable and intelligent solution that keeps data integrity in distributed and heterogenous cloud storage infrastructures.

The primary novelty of this work lies in the synergistic integration of DGVAE in a multi-user blockchain architecture with quantum secure cryptographic modules (QHABE and QKDCPABE) to provide end-to-end data integrity assurance in heterogeneous cloud environments. It is the first framework that uses disentangled graph-structured latent representations to do integrity verification after decryption, which makes it able to detect even minor data manipulations that traditional hash-comparison methods cannot.

The principal contributions of this paper are summarized as follows:

- **MUBF-DGVAE Framework:** A novel end-to-end framework is proposed that integrates DGVAE within a multi-user blockchain architecture, supported by advanced quantum encryption, to achieve reliable tamper detection and secure multi-user data access in heterogeneous cloud environments.
- **Heterogeneous Cloud Dataset Utilization:** A heterogeneous cloud dataset is used to train and validate the framework, with a total of 5,000 records across multiple cloud providers, closely simulating the complexity of distributed, multi-source cloud deployments.
- **QHABE-Based Encryption:** Quantum Hash-Based Attribute-Based Encryption is introduced to encrypt input data according to defined access policies, ensuring that only authorized users whose attributes satisfy the policy can decrypt the data, while simultaneously embedding a quantum-resistant hash for integrity verification.
- **QKDCPABE for Secure Key Distribution:** Multi-Qubit Quantum Key Distribution Ciphertext-Policy Attribute-Based Encryption is employed for private key generation via quantum-level attribute validation, enabling fine-grained and quantum-resilient access control in multi-user settings.
- **Blockchain-Backed Immutable Ledger:** A blockchain ledger stores transaction records comprising encrypted data references, access policies, timestamps, and initial hash keys, ensuring tamper-resistance, transparency, and auditable traceability of all data access events.
- **DGVAE-Based Integrity Verification:** The Disentangled Graph Variational Autoencoder is employed to compare graph-structured representations of re-hashed decrypted data against the original blockchain-stored hash, enabling robust and intelligent detection of tampering or data corruption following decryption.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the proposed methodology. Section IV presents experimental results and analysis. Section V concludes the paper and outlines directions for future work.

II. Related Work

This area briefly summarizes recent studies on multi-user frameworks based on blockchain, quantum secure encryption techniques, and integrity checking in cloud computing via deep learning approaches.

[11] Karumanchi et al. (2023) introduced a multi-user blockchain model for specific supply chain management involving heterogeneous supply chains. The design adopted by their approach is a hybrid data-integrity scheme with variable sizes for supply chain files stored in the cloud, which integrates advanced heterogeneous integrity computation, and attribute encryption and decryption based on integrity policies to provide high security for cloud data. But the framework has scalability constraints and high transaction latency for large-scale transaction loads.

[12] Garigipati et al. (2025) proposed a poly-quantum integrity key generation scheme for multi-user access control in homogeneous and heterogeneous cloud Electronic Health Record (EHR) databases. The framework uses chaotic poly-quantum keys to create integrity values of different sizes for encryption and decryption, which boosts the security of quantum-safe systems. However, the method is very complex and can be problematic for real-time deployments.

[13] Li et al. (2024) proposed a Real-time Adaptive Partition (RAP) framework for end-cloud inference collaboration among multiple users in mobile environments. The framework includes a split-point decision algorithm for the adaptive partitioning of the DNN and a Joint Multi-user Model Partition and Resource Allocation (JM-MPRA) algorithm. Although it's resource-efficient, the solution necessitates a stable network connection and doesn't cover data integrity within cloud storage.

[14] Li et al. (2024) proposed a revocable and verifiable weighted Attribute-Based Encryption scheme with collaborative access (RVWABE-CA) to secure EHR sharing in public cloud environments. While providing fine-grained revocable multi-user collaborative access and data integrity verification, the scheme has significant computational overhead as a result of the complexity of the policies and revocation rules.

[15] Zhang et al. (2024) proposed a distributed cloud computing privacy-preserving multi-user image search system. The system is based on an SGX architecture to ensure that the similarity of images is protected from cloud servers and offers the ability to obfuscate access by re-encrypting the index. The method is efficient in keeping search and access pattern private, but has significant processing cost, resulting in low scalability when applied to large distributed environments.

A summary of the methods, aims, advantages and limitations of related works are provided in Table I. A cross-cutting analysis shows that individual methods tackle each aspect of the cloud security puzzle, such as integrity, access

control, privacy, but none at the same time provide quantum-resilient encryption, immutability with blockchain, and intelligent verification of integrity on a graph-structured format across a multi-user setting. The proposed MUBF-DGVAE fills these gaps with comprehensive and scalable integrity assurance for heterogeneous cloud environments, based on blockchain technology, quantum cryptographic primitives, and disentangled deep graph learning.

TABLE I. Comparative Summary of Related Works

Reference	Method	Objective	Advantages	Limitations
Karumanchi et al. [11] (2023)	Hybrid integrity + Blockchain	Secure multi-user supply chain	Supports heterogeneity; data integrity	Scalability concerns; high latency
Garigipati et al. [12] (2025)	Poly-quantum key gen + ABE	Secure cloud EHR access	Quantum-safe; strong access control	High computational complexity
Li et al. [13] (2024)	Adaptive DNN partitioning + resource allocation	Efficient mobile inference	Real-time; resource-efficient	Requires stable connectivity
Li et al. [14] (2024)	RVWABE-CA	Secure EHR sharing with revocation	Fine-grained; revocable access	High computational overhead
Zhang et al. [15] (2024)	SGX-based privacy-preserving search	Secure image search in distributed cloud	Protects search and access patterns	Significant processing overhead

III. Proposed Methodology

This section introduces the MUBF-DGVAE approach for improving the integrity of data in a heterogeneous cloud environment by providing secure access management and tamper detection by optimizing the access management. This section introduces the MUBF-DGVAE approach for improving data integrity in a heterogeneous cloud environment, through optimized secure access management and tamper detection. The framework consists of three main stages: (1) Data Acquisition and Hash Generation, (2) Encryption and Blockchain Storage, and (3) Verification and Integrity Checking. The overall system architecture is shown in Fig. 1.

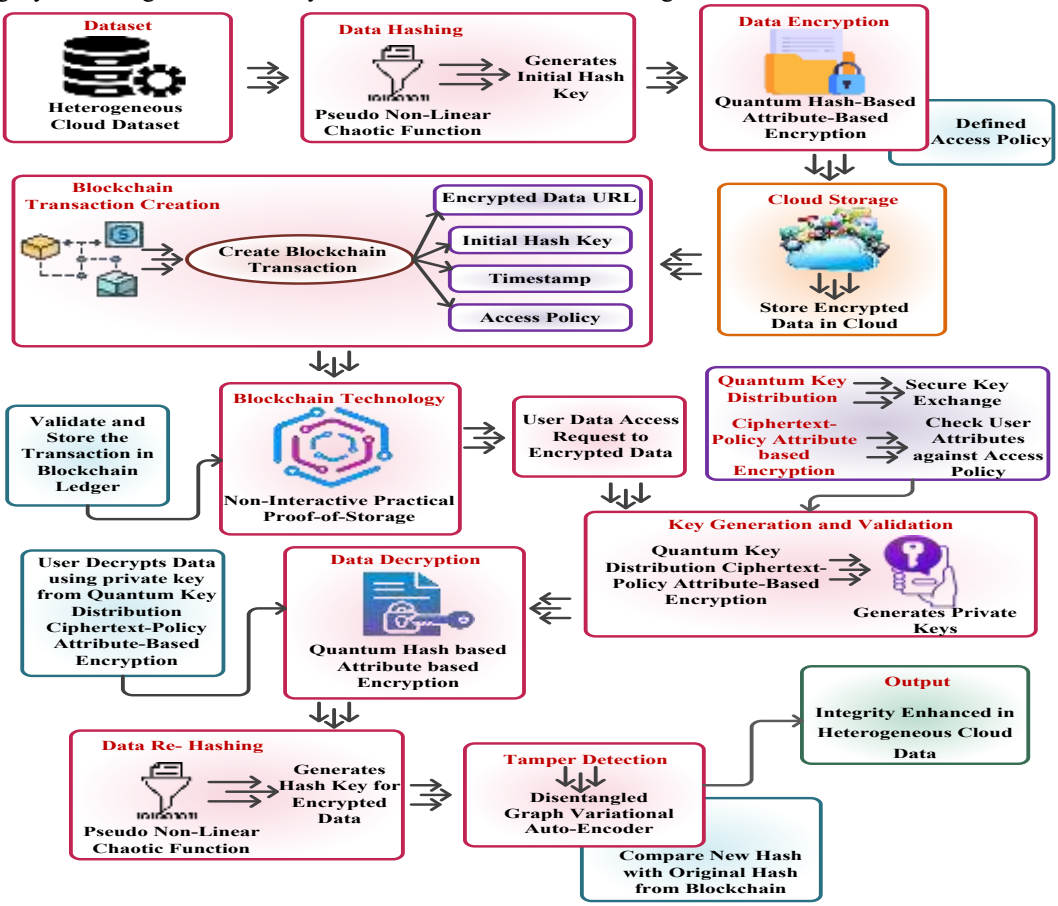

Fig. 1: Block Diagram of the Proposed MUBF-DGVAE Framework

A. Data Acquisition

The framework uses a Heterogeneous Cloud Dataset [16] which consists of 5000 records, each with a unique text document and metadata. These records contain eight pieces of information: file name, user role, cloud provider, access level, a time stamp, file path, file size, and the original file text content. All of these features provide a representation of the document content and its context in the cloud environment. This dataset is split into training (80%), validation (10%) and test (10%) sets to ensure strong model training and assessment.

The dataset is evenly split by user role (around 1600 files per auditor, viewer, and admin) by access level (1696 files public, 1699 files confidential, and 1695 files internal) by cloud provider (Azure: 34%, Google Drive: 33.2%, AWS S3: 32.8%). Most of the files are in the range 200 to 300 characters, with the highest concentration of files in the range 240 to 260 characters. Over time, the distribution of file creation activity (from May 2024 through May 2025) is reasonably stable, but it has spikes as high as 25 files/day, indicating the dataset is appropriate for use when evaluating real-world multi-cloud scenarios.

B. Non-Linear Chaotic Hash Generation (PNLCF)

A Polynomial Non-Linear Chaotic Function (PNLCF) is used to create a hash fingerprint with data record, which is used as the prior baseline to verify integrity in the future. The PNLCF works as follows: It transforms user attributes into a byte array and then splits the input data M into k fixed size blocks of 8 bit each. If the length of the message is greater than the block length, it is extended to the length of the block by adding the specified bit pattern (0000001). Every block also has 32-bit sub-blocks that go through a sequence of non-linear mathematical transformations to derive intermediate hash values. The hash values of the sub-blocks are then combined together to generate a hash fingerprint. The avalanche properties of PNLCF are strong and the Hamming distance is about 128 bits when slight alteration of the data, regardless of the size of the input.

C. Data Encryption via Quantum Hash-Based Attribute-Based Encryption (QHABE)

Quantum Hash-Based Attribute-Based Encryption (QHABE) [17] is used to encrypt the input data based on a predetermined access policy. QHABE combines an ABE scheme and quantum resistant hash functions to offer strong protection against classical and quantum attacks, and fine-grained, policy-driven access control.

The quantum hash is computed from the Recursive Non-Linear Polynomial Graph-Centered Integrity Algorithm (RNLPIA), which uses a non-linear polynomial map with chaotic property defined in a prime cyclic group of elements, using a multiplicative structure. A hash function is given by:

$$H(i) = f(D, H(i-1), H(i-2), \lambda \cdot i) \quad (1)$$

where H(i) is the polynomial hash value at iteration i; D is the input data; H(i−1) and H(i−2) are hash values resulting from the previous two iterations; and λ·i is a scaling factor that increases as a function of the iteration index. Summarizing the QHABE encryption procedure is shown in Algorithm 1.

Algorithm 1: QHABE Encryption Procedure

Input: Plaintext data D; Access policy P (specifying required user attributes)
Output: Ciphertext CT (encrypted data, embedded policy, quantum hash)

Step 1: Generate quantum-resistant hash using RNLPIA:
    QH ← RNLPIA(D) [Eq. (1)]
Step 2: Encrypt data under attribute-based access policy:
    CT_data ← ABE_Encrypt(D, P)
Step 3: Bundle ciphertext, hash, and policy:
    CT ← {CT_data, QH, P}
Return CT

The QHABE algorithm produces a quantum resistant hash QH as a unique digital fingerprint of the plaintext data, then encrypts the data with ABE with respect to the access policy P. The encrypted data, integrity hash and policy are combined into a ciphertext CT, which can only be decrypted by users whose attributes match P. The encrypted data and embedded Quantum hash is then sent to the cloud for storage.

D. Blockchain Transaction Creation

When QHABE encryption is completed, a blockchain transaction will be generated to record and validate the provenance information of the data. The four essential elements in each transaction are: (i) the Encrypted Data URL containing a pointer to the encrypted data that are stored in the cloud; (ii) the Initial Hash Key, a hash key generated by the PNLCF which acts as the integrity baseline; (iii) the Timestamp, a record of the precise time of the data upload; and (iv) the Access Policy, which is the policy that determines who can decrypt the stored data. By capturing all these elements on the blockchain ledger, the data is made immutable, transparent, and auditable, offering a reliable way to manage access to sensitive data in heterogeneous cloud environments.

E. Non-Interactive Practical Proof-of-Storage (nPPoS) for Blockchain Validation

The new Non-Interactive Practical Proof-of-Storage (nPPoS) [18] is used to validate and record blockchain transactions. nPPoS allows a data owner (client) to create an efficient cryptographic proof of data stored correctly and completely without the need for iterative client-server communication, and this proof can be independently verified by a verifier. This design is significantly communication overhead reduced, yet verifiable, and with data integrity assurance.

The generation of the proof is formalised as:

$\pi \leftarrow \text{Prove}(D, sk\_client)$ (2)
where π is the proof that the client generates without interacting with the server, D is the data stored on the server and sk_client is the secret key stored on the client. The proof independently verifies the correct storage of the data without needing additional client interaction. The verifier performs its own verification of π and the verified proof is transferred to the blockchain ledger in an unchangeable way as follows:
$\text{Ledger} \leftarrow \text{Store}(\text{Verify}(\pi), \text{Blockchain})$ (3)
ensuring transparency, decentralized accessibility, and tamper-resistance of the stored transaction records.

F. Private Key Generation via Multi-Qubit QKDCPABE

If a user wants to access the data in the cloud, the Multi-Qubit Quantum Key Distribution Ciphertext-Policy Attribute-Based Encryption (QKDCPABE) [19] is called to obtain the private decryption key(s). QKDCPABE is based on quantum properties such as photon entanglement and the no-cloning theorem, which guarantee that any eavesdropping attempt in the quantum process to exchange keys results in an irreversible alteration of the quantum state, and that this alteration is noticed in the very act.
The key generation process is formalized in Algorithm 2: The process starts with a Quantum Key Distribution (QKD) protocol in order to generate a secure shared key. Access is refused if the QKD process detects that eavesdropping has taken place or the process fails. Otherwise, the user attributes are checked against the pre-stored access policy which is embedded in the ciphertext. If the attribute set meets the policy, then a private attribute key is created by the CP-ABE component along with the QKD key.

Algorithm 2: Multi-Qubit QKDCPABE Private Key Generation
Input: User attributes A; Access policy P
Output: PrivateKey SK, or AccessDenied

```
function KeyGen(A, P):
  // Step 1: Quantum Key Distribution
  sk_qkd ← QKD_Exchange(A)
  if Eavesdropping_Detected(sk_qkd) then
    return AccessDenied
  end if
  // Step 2: Attribute validation against access policy
  if not Satisfies(A, P) then
    return AccessDenied
  end if
  // Step 3: Private key generation
  SK ← CPABE_KeyGen(A, P, sk_qkd)
  return SK
end function
```

The hybrid quantum-classical method provides security for sensitive cloud data in multi-user settings with confidentiality, fine-grained access control, and resistance to both quantum and classical cryptographic attacks.

G. Data Decryption and Re-Hashing

If QKDCPABE can successfully generate its private key, the user decrypts the ciphertext by using QHABE and the private key SK generated by QKDCPABE. The decryption is granted only when the user attributes are matched with the access policy defined. The decrypted data is then fed into the PNLCF to compute hash. This regenerated hash is used as a fresh fingerprint of the decrypted data, and then passed on to the DGVAE module to compare the data with the original hash saved on the blockchain.

H. Hash Validation via Disentangled Graph Variational Autoencoder (DGVAE)

The integrity verification module of the proposed framework is the Disentangled Graph Variational Autoencoder (DGVAE) [20]. DGVAE can learn different and independent latent representations of graph-structured data, which allows for the robust and fine-grained comparison of the hash representations for the purpose of tamper detection.
The DGVAE architecture consists of: (i) a feature extractor based on a Transformer, which generates an item-item graph from the data; (ii) a Graph Encoder to generate disentangled latent variables from the graph; (iii) a Mutual Information Maximization (MIM) module to improve the quality and discriminability of latent representations; and (iv) a Decoder to reconstruct the input from the latent variables, where a reconstruction loss is utilized to ensure accurate representation learning. The architecture of DGVAE is shown in Fig. 2.

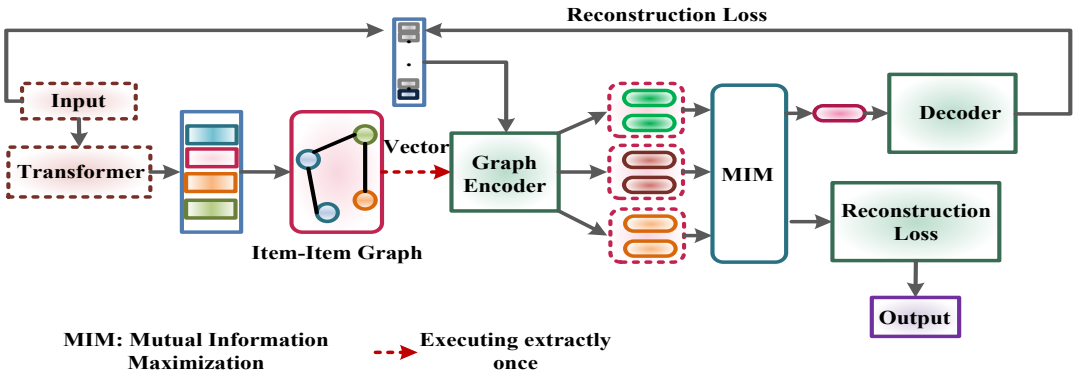

Fig. 2: DGVAE Architecture Diagram

The DGVAE encodes the original graph data G into disentangled latent variables Z and generates an integrity hash H_G as:

$Z = \text{Encoder}(G), \quad H\_G = \text{Hash}(Z) \quad (4)$

The original hash H_G derived from the graph data is then stored immutably on the blockchain ledger:

$\text{Ledger} \leftarrow \text{Store}(H\_G) \quad (5)$

Upon decryption, the DGVAE regenerates a hash H'_G from the decrypted graph data and performs the integrity comparison:

$\text{Integrity} \leftarrow (H\_G == H'\_G) \quad (6)$

If H_G and H'_G are identical, the data is certified as intact. Any discrepancy indicates unauthorized modification or corruption, triggering an integrity violation alert. The use of disentangled latent representations ensures that even subtle structural manipulations in the data are reflected in the hash comparison, enabling precise and reliable tamper detection beyond the capabilities of conventional hash-matching approaches.

IV. Experimental Results And Discussion

In this section, the experimental evaluation of the proposed MUBF-DGVAE framework is presented. All simulations were performed using Python and evaluated with five performance criteria namely Hash Bit Change, Hash Time, Cloud Encryption Runtime, Aggregation Time and Cloud Decryption Runtime. The proposed approach is compared with three existing approaches: EI-MUBF-HSCM [21], EPQKG-MUAED-EHR [22] and ARA-MUCIC-ME [23].

A. Performance Metrics

Hash Bit Change: Calculates the Hamming Distance between the hash before encryption and the hash after decryption. The statistical significance and consistency of the bit change suggest that the hash function is sensitive to changes in the data, which confirms the collision resistance

Hash Time: Measures the amount of time it takes PNLCF to compute a hash of the input data. Faster hash times means that this process is more efficient, and it will allow for the rapid integrity fingerprinting of data while uploading it and verifying it after the data has been decrypted.

Cloud Encryption Runtime: The amount of time the data takes to be encrypted by the QHABE algorithm before it gets stored in the cloud. This number measures how much computing power is required to evaluate attributes and create the ciphertext when there are many concurrent users.

Aggregation Time: The time needed to gather and combine data (such as PNLCF hashes, encrypted data URLs, time stamps, and access policies) into blockchain transactions. A faster aggregation time means efficient throughput of ledger transactions.

Cloud Decryption Runtime: Time to decrypt the ciphertext stored in the cloud and obtain the plaintext that can be recovered with the authorized private key created by QKDCPABE. This measure quantifies the effectiveness of decryption when the number of users connected to the system increases; directly representing system responsiveness in multi-user cloud environment.

B. Hash Performance Analysis

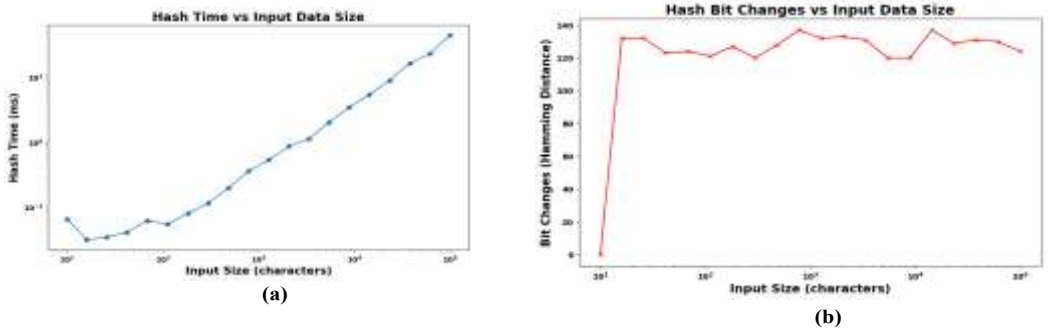

Figure 3: Analysis of (a) Hash Time and (b) Hash Bit Changes vs Input Data Size

The hash time with the input data size (10 to 100,000 characters) is shown in Fig. 3(a). The proposed MUBF-DGVAE consistently yields the lowest hash times ranging from 0.011 ms to 0.080 ms with all the input size while the EI-MUBF-HSCM, EPQKG-MUAED-EHR and ARA-MUCIC-ME have hash times in the range of 0.077 ms to 45.39 ms, 0.629 ms to 4.86 ms and 0.312 ms to 0.89 ms respectively. These results confirm the lightweight scalability of PNLCF, and validate its suitability as a fast integrity fingerprinting mechanism for large scale, heterogeneous cloud deployments.

The Hash Bit Change analysis is shown in Fig. 3(b), which shows that the proposed framework has a stable computation time of the hash function, where the hash computation time is around 11.9 ms to 14.8 ms for all the input sizes, thus the strong and stable avalanche of PNLCF. These times for EI-MUBF-HSCM are much smaller, although they are more variable (2.5–4.5 ms); for ARA-MUCIC-ME these times spike to more than 11 ms, and for EPQKG-MUAED-EHR they exceed 5 ms frequently. The slightly elevated but stable times of the proposed method are justified by the additional security layers and integrity verification features it provides.

C. Cloud Encryption and Decryption Runtime Analysis

The Fig. 4(a) shows the Cloud Encryption Runtime plotted against the number of concurrent users (0-100). The proposed MUBF-DGVAE gives much faster encryption time compared to the EI-MUBF-HSCM (456.62ms), EPQKG-MUAED-EHR (344.38ms) and ARA-MUCIC-ME (224.87ms) schemes, with the encryption time ranging from 24.05ms to 124.47ms for 100 users. The consistent and predictable scaling to 60 users validates the framework for implementing real-time, large scale data encryption in a heterogeneous cloud environment.

Cloud Decryption Runtime for the same user range is shown in Fig. 4(b). The proposed framework exhibits the lowest decryption time of around 19.99 ms for zero users and 83.54 ms for 100 users, while the EI-MUBF-HSCM demonstrated the highest time of 217.18–280.58 ms, followed by EPQKG-MUAED-EHR with a time of 152.39–226.08 ms and ARA-MUCIC-ME with a time of 61.77–155.43 ms. The results confirm that the framework is able to sustain efficient and scalable decryption performance as the number of users grows.

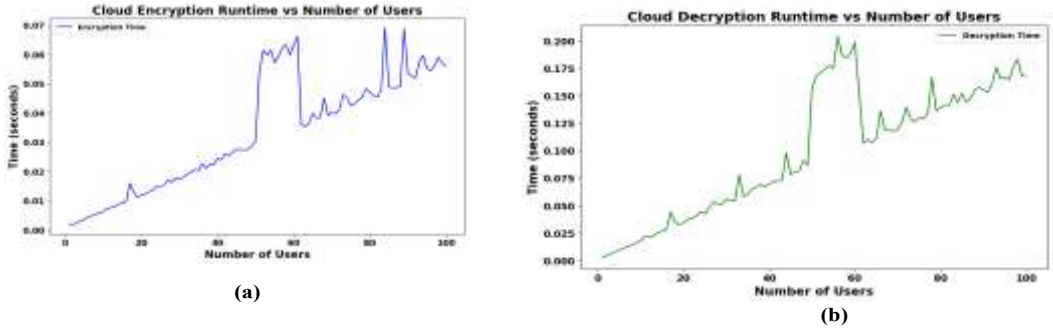

Figure 4: Analysis of (a) Cloud Encryption Runtime and (b) Cloud Decryption Runtime vs Input Data Size

D. Aggregation Time and DGVAE Training Analysis

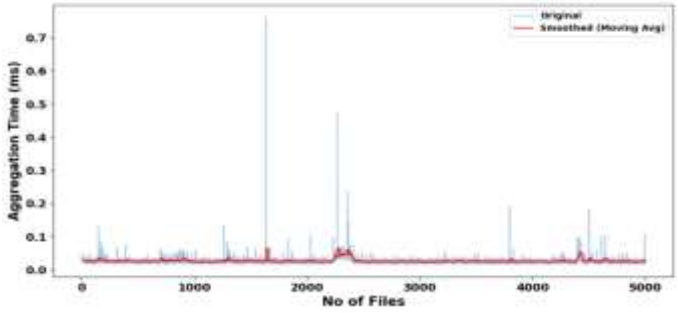

Figure 5: Analysis of Aggregation Time

Fig. 5 shows the Aggregation Time against the number of files (0 - 5,000). The proposed MUBF-DGVAE achieves the aggregation time starting from approximately 64.77 ms and up to 289.75 ms in the case of the largest file volumes, which is significantly better than the EI-MUBF-HSCM (932.69 ms), EPQKG-MUAED-EHR (760.77 ms), and ARA-MUCIC-ME (516.24 ms). The average transaction time is around 0.05ms per transaction, which underlines the low latency blockchain integration achieved due to the nPPoS consensus mechanism.

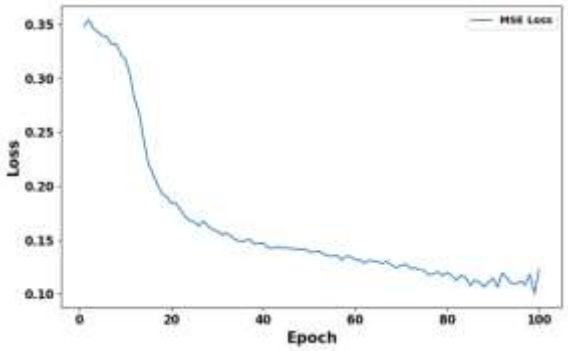

Figure 6: Analysis of Loss

The training convergence of DGVAE is shown as Mean Squared Error (MSE) loss vs epoch in Fig. 6. During 100 epochs, the MSE loss gradually reduces from 0.35 to 0.10, indicating that the DGVAE is able to learn to differentiate between unrepaired and manipulated data representations efficiently. This tight coupling shows that disentangled representation learning can be very effective in identifying integrity violations when comparing the hash after decryption.

TABLE II. Quantitative Performance Comparison

Method	**Hash Time (ms)**	**Encrypt. Runtime @ 100 users (ms)**	**Aggregation Time @ 5000 files (ms)**	**Decrypt. Runtime @ 100 users (ms)**
EI-MUBF-HSCM [21]	Up to 45.39	456.62	932.69	217.18–280.58
EPQKG-MUAED-EHR [22]	Up to 4.86	344.38	760.77	152.39–226.08
ARA-MUCIC-ME [23]	Up to 0.89	224.87	516.24	61.77–155.43
Proposed MUBF-DGVAE	**0.011–0.080**	**124.47**	**289.75**	**83.54**
Improvement (vs. best baseline)	~91% lower	~45% lower	~44% lower	~46% lower

A tabular comparison of the quantitative performance is provided in Table II, which summarises the results of the methods evaluated. The proposed MUBF-DGVAE consistently gives the lowest runtime across the metrics, showing its superior computational efficiency, scalability and its ability to ensure data integrity in large-scale, multi-user, heterogeneous cloud environments in real-time.

V. Conclusion

A novel multi-user blockchain framework, MUBF-DGVAE, has been presented, which is combined with a Disentangled Graph Variational Autoencoder to guarantee the integrity of the data in the heterogeneous cloud environment. This framework tackles multiple problems of tamper detection and fine-grained access control, effectively working in conjunction with the immutability of the blockchain, quantum-resistant cryptographic primitives (QHABE and QKDCPABE), and graph-based intelligent integrity verification (DGVAE). Experimental assessments show that the proposed framework has a hash time of 0.011–0.080 ms, an aggregation time of 289.75 ms with 5,000 files, and a cloud decryption time of 83.54 ms with 100 users, all of which are better results than all the baseline methods evaluated.

Future studies will look into hybrid approaches to integrity verification, leveraging both cryptographic hash functions and behavioral anomaly analysis to offer more robust and sustainable integrity assurance. Future research challenges involve designing cryptographic primitives dedicated to verifying integrity across clouds and extending the framework to federated multi-cloud policies that change over time.

References

[1] S. Khan, Z. Jiangbin, M. Irfan, F. Ullah, and S. Khan, "An expert system for hybrid edge to cloud computational offloading in heterogeneous MEC-MCC environments," J. Netw. Comput. Appl., vol. 225, p. 103867, 2024.

[2] P. Tamilselvi, "Blockchain chain based cloud security using provable partitioned folding encryption for integrity proofing in cloud environment," SN Comput. Sci., vol. 5, no. 8, pp. 1–12, 2024.

[3] T. H. Hoang et al., "Enabling end-to-end secure federated learning in biomedical research on heterogeneous computing environments with AppFLX," Comput. Struct. Biotechnol. J., vol. 28, pp. 29–39, 2025.

[4] S. Chen and W. Jiang, "Online dynamic multi-user computation offloading and resource allocation for HAP-assisted MEC: An energy efficient approach," J. Cloud Comput., vol. 13, no. 1, p. 92, 2024.

[5] H. A. Shafei and C. C. Tan, "A closer look at access control in multi-user voice systems," IEEE Access, vol. 12, pp. 40933–40946, 2024.
[6] X. Li, H. Wang, and S. Ma, "An efficient ciphertext-policy weighted attribute-based encryption with collaborative access for cloud storage," Comput. Stand. Interfaces, vol. 91, p. 103872, 2025.
[7] S. Khan, H. Abbas, and W. Iqbal, "Verifiable privacy-preserving image retrieval in multi-owner multi-user settings," IEEE Trans. Emerg. Top. Comput. Intell., vol. 8, no. 2, pp. 1640–1655, 2024.
[8] Y. Li, C. Xu, L. Xu, L. Mei, and Y. Zhu, "Verifiable searchable encryption scheme with flexible access control in the cloud," J. Parallel Distrib. Comput., vol. 197, p. 105025, 2025.
[9] B. Sonkoly et al., "An edge cloud based coordination platform for multi-user AR applications," J. Netw. Syst. Manage., vol. 32, no. 2, p. 40, 2024.
[10] L. Xu, X. Cheng, W. Tian, H. Wang, and Y. Zhang, "Cloud-assisted privacy-preserving spectral clustering algorithm within a multi-user setting," IEEE Access, 2024.
[11] M. D. Karumanchi, J. I. Sheeba, and S. P. Devaneyan, "An efficient integrity based multi-user blockchain framework for heterogeneous supply chain management applications," Int. J. Comput. Appl., vol. 45, no. 4, pp. 337–351, 2023.
[12] N. Garigipati, S. Srithar, and V. Krishna Reddy, "An efficient poly-quantum integrity key generation based multi-user access control encryption and decryption framework for homogeneous and heterogeneous cloud EHR databases," Inf. Secur. J. Glob. Perspect., pp. 1–21, 2025.
[13] Y. Li, Z. Liu, Z. Kou, Y. Wang, G. Zhang, Y. Li, and Y. Sun, "Real-time adaptive partition and resource allocation for multi-user end-cloud inference collaboration in mobile environment," IEEE Trans. Mobile Comput., 2024.
[14] X. Li, H. Wang, S. Ma, M. Xiao, and Q. Huang, "Revocable and verifiable weighted attribute-based encryption with collaborative access for electronic health record in cloud," Cybersecurity, vol. 7, no. 1, p. 18, 2024.
[15] Z. Zhang, F. Zhou, and Y. Wang, "Multi-user privacy-preserving image search in distributed cloud computing," Comput. Electr. Eng., vol. 118, p. 109434, 2024.
[16] Radicles Info, "Heterogeneous Cloud Dataset," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/radiclesinfo/heterogeneous-cloud-dataset
[17] K. K. Singamaneni, G. Muhammad, and Z. Ali, "A novel quantum hash-based attribute-based encryption approach for secure data integrity and access control in mobile edge computing-enabled customer behavior analysis," IEEE Access, 2024.
[18] J. W. Heo, G. Ramachandran, and R. Jurdak, "nPPoS: Non-interactive practical proof-of-storage for blockchain," Blockchain Res. Appl., vol. 5, no. 4, p. 100221, 2024.
[19] K. K. Singamaneni, G. Muhammad, and Z. Ali, "A novel multi-qubit quantum key distribution ciphertext-policy attribute-based encryption model to improve cloud security for consumers," IEEE Trans. Consum. Electron., vol. 70, no. 1, pp. 1092–1101, 2023.
[20] X. Zhou and C. Miao, "Disentangled graph variational auto-encoder for multimodal recommendation with interpretability," IEEE Trans. Multimedia, 2024.
[21] M. D. Karumanchi, J. I. Sheeba, and S. P. Devaneyan, "An efficient integrity based multi-user blockchain framework for heterogeneous supply chain management applications," Int. J. Comput. Appl., vol. 45, no. 4, pp. 337–351, 2023.
[22] N. Garigipati, S. Srithar, and V. Krishna Reddy, "An efficient poly-quantum integrity key generation based multi-user access control encryption and decryption framework for homogeneous and heterogeneous cloud EHR databases," Inf. Secur. J. Glob. Perspect., pp. 1–21, 2025.
[23] Y. Li et al., "Real-time adaptive partition and resource allocation for multi-user end-cloud inference collaboration in mobile environment," IEEE Trans. Mobile Comput., 2024.