Panamerican Mathematical Journal
ISSN: 1064-9735
Vol 35 No. 3s (2025)

Performance Optimization of Hybrid Encryption Techniques in Oracle Cloud Infrastructure: A Comparative Study

Sourabh Jain[1], Dr. Rajesh K Shukla[2]

[1]Research Scholer, [2]Professor

[1,2]Computer Science and Engineering Department

[1]Oriental University Indore, India

[2]Oriental University Indore, India

Sourabhjain0412@gmail.com , shukladrrajeshk@gmail.com

Article History:

Received: *24-09-2024*

Revised: *12-10-2024*

Accepted: *06-12-2024*

Abstract: Cloud security has become a critical concern due to the increasing volume of sensitive data stored and processed in cloud environments. Traditional encryption methods often fail to balance security, efficiency, and scalability, leading to vulnerabilities in cloud infrastructure. This study explores the performance optimization of hybrid encryption techniques in Oracle Cloud Infrastructure (OCI) by integrating RSA, Blowfish, Homomorphic Encryption, and Blockchain-based key management. The proposed models enhance data confidentiality, secure key exchange, and time-limited access control. Experimental results show that RSA + Blowfish and Homomorphic Encryption + Blockchain achieve higher accuracy (99.86%), lower decryption latency, and reduced error rates (MSE: 0.189, MAE: 0.349) compared to existing techniques. These findings establish hybrid encryption as a robust solution for securing OCI-based data storage and computation while maintaining optimal performance.

Keywords: Hybrid Encryption, Oracle Cloud Security, Blockchain-based Key Management, Homomorphic Encryption, Performance Optimization, Cryptographic Algorithms.

1. Introduction

Cloud computing has revolutionized data storage and processing by offering scalable, cost-effective, and efficient computing resources. Organizations increasingly rely on cloud platforms like Oracle Cloud Infrastructure (OCI) for handling sensitive data, including healthcare records, financial transactions, and enterprise operations. However, with the rise of cloud adoption, security vulnerabilities have also increased, making data confidentiality, integrity, and access control primary concerns. Traditional encryption techniques such as AES, RSA, and Blowfish provide security, but they often face challenges related to computational overhead, key management, and resistance to evolving cyber threats. Hybrid encryption models that combine symmetric and asymmetric cryptography, along with blockchain-based key management and homomorphic encryption, have emerged as a promising solution to optimize cloud security while maintaining computational efficiency.

This research focuses on enhancing security in Oracle Cloud Infrastructure (OCI) using hybrid encryption techniques. The study evaluates the effectiveness of integrating RSA + Blowfish encryption and Homomorphic Encryption + Blockchain-based key management in cloud security. The proposed model aims to improve encryption and decryption speed, key exchange security, and access control policies while minimizing computational overhead. By leveraging homomorphic encryption, this study enables privacy-preserving computations on encrypted data without decryption, making it highly suitable for cloud-based healthcare, financial transactions, and enterprise data security. The inclusion of blockchain technology ensures tamper-resistant key storage and access monitoring, further strengthening cloud security.

This research makes the following key contributions:

1. Development of a Hybrid Encryption Framework: The study proposes an optimized hybrid encryption model that integrates RSA, Blowfish, and Blockchain-based key management to secure cloud data efficiently.

2. Implementation of Homomorphic Encryption: The research incorporates fully homomorphic encryption (FHE) to enable privacy-preserving computations on encrypted data, ensuring enhanced data security in OCI.
3. Blockchain-Based Key Management: The paper introduces a decentralized, tamper-proof key distribution system using blockchain, which eliminates vulnerabilities associated with traditional key exchange mechanisms.
4. Performance Optimization & Comparative Analysis: The study evaluates and compares the performance of the proposed hybrid encryption methods against conventional encryption techniques in terms of encryption time, decryption latency, accuracy, precision, recall, and error metrics (MSE & MAE).
5. Implementation and Real-World Applicability: The research provides a practical implementation framework applicable to healthcare data security, financial transactions, and enterprise cloud storage, ensuring scalability and efficiency.

The remainder of this article is structured as follows, Section 2 provides a summary of related research, highlighting existing encryption methodologies and their limitations in cloud security. Section 3 presents the problem statement, addressing the key challenges in encryption performance and data security in Oracle Cloud Infrastructure. Section 4 discusses the methodology and architecture of the proposed hybrid encryption approach, detailing key components like RSA + Blowfish encryption, Homomorphic Encryption, and Blockchain-based key management. Section 5 presents the experimental findings and discussion, comparing the performance metrics of hybrid encryption with traditional methods. Section 6 concludes the study, summarizing key insights and outlining future research directions for enhancing cloud security.

2. Literature Review

Shivaramakrishna et al. (2023), Cloud computing's rapid expansion demands robust security solutions to protect sensitive data stored remotely. This study introduces a hybrid cryptographic framework incorporating time-limited access control, adaptive key management, and dual encryption methods (RSA and AES-OTP). The framework enhances security through intelligent key creation, distribution, and rotation, while time-limited access control adds an extra layer of privacy protection. Performance evaluation demonstrates high security and efficiency, achieving accuracy, precision, recall, and F1-score values of 99.12%, 98.78%, 98.11%, and 98.56%, respectively [1].

Rakhra et al. (2024), Cloud computing has revolutionized data management but introduces security challenges due to its centralized nature. Traditional cryptographic methods alone may be insufficient, making hybrid cryptography, which combines symmetric and asymmetric encryption, a promising solution. Symmetric encryption ensures fast and efficient data encryption, while asymmetric encryption provides secure key exchange and authentication. This hybrid approach effectively enhances cloud security without compromising computational efficiency, making it ideal for multi-party data access scenarios [2].

Sasikumar et al. (2024), Cloud computing offers cost-effective and scalable solutions, but security risks arise from third-party storage and internet-based access. Cryptography plays a vital role in ensuring data authentication, confidentiality, integrity, and availability. This study comprehensively examines various cryptographic methods, including DNA-based, elliptic curve, homomorphic, hybrid, and lightweight cryptography, analyzing their methodologies and applications. It suggests using elliptic curve cryptography (ECC) for secure communication and lightweight cryptography for IoT devices, highlighting the importance of hybrid cryptography in balancing security and efficiency [3].

Reddy et al. (2024), Cloud computing centralizes processing and storage, enabling efficient handling of large datasets. This study proposes a two-tier cryptographic security system combining symmetric (DES, AES) and asymmetric (Elliptic Curve Cryptography - ECC) encryption. ECC generates secret keys used in layered encryption and decryption, ensuring scalability, adaptability, and reliability for secure cloud storage [4].

Debnath et al. (2024), Securing cloud data transactions is a major challenge, and traditional cryptographic methods may not be sufficient. This research introduces hybrid cryptographic models integrating Modified-RSA (MRSA), which enhances security by using three prime numbers instead of two. A comparative study of RSA and MRSA hybrid models was conducted using Python simulations to evaluate encryption time, decryption time, throughput, and efficiency, identifying the most effective model for cloud security [5].

Ali et al. (2024), The Internet of Things (IoT) connects billions of devices, generating massive data that needs secure transmission and storage. This study proposes a hybrid cryptographic approach combining DNA cryptography, Genetic Algorithm (GA), and Elliptic Curve Cryptography (ECC) to ensure data confidentiality. The algorithm improves key generation, encryption, and decryption performance, making it suitable for resource-constrained IoT devices like WSNs and RFIDs [6].

Shrivastava et al. (2024), Cloud services face security threats due to unauthorized data access. This study proposes a blockchain-based hybrid cryptographic scheme (HLEEE) integrating Elliptic Curve Cryptography (ECC), ElGamal encryption, and SHA-256 hashing. Flamingo Search Optimization (FSO) is used for optimal key selection, while Proof of Authority (PoA) ensures blockchain validation. The model enhances cloud data security by leveraging blockchain's immutability and decentralized nature [7].

Agrawal et al. (2024), Blockchain ensures data integrity and transparency, while fog computing enhances efficiency by distributing cloud capabilities. This study introduces a hybrid encryption model that integrates blockchain, fog computing, and the Hybrid Encryption Algorithm (HEA) for secure data access, distributed storage, and authentication. It employs Deep Adaptive Power Probabilistic Clustering for optimal cluster selection, ECDH for secure key exchange, and SHA-512 for hashing. Ethereum Smart Contracts (SC) further strengthen cross-domain trust, achieving a 95% reliability score in simulations [8].

Abitova et al. (2024), Secure file storage is critical in the digital era. Hybrid cryptography, combining symmetric and asymmetric encryption, offers strong protection against unauthorized access, tampering, and data loss. This paper reviews advancements in post-quantum cryptography, blockchain-based encryption, homomorphic encryption, and secure multi-party computation, addressing emerging security threats. Hybrid cryptography's adaptability makes it a powerful approach for ensuring long-term data security in evolving digital environments [9].

Sengupta et al. (2024), Traditional encryption methods (AES, DES, RC6, LSB steganography) face limitations in cloud-based file sharing. This study explores the transition to multi-party encryption, which improves data security, access control, and integrity in distributed computing. A structured methodology is presented, highlighting implementation strategies and expected impact on efficiency, security, and usability, ultimately enhancing cloud-based file-sharing practices [10].

Abdo et al. (2024), Cloud computing demands efficient and secure data transmission. This research proposes a hybrid encryption-compression approach, integrating multiple encryption layers with LZMA compression to enhance performance and prevent unauthorized access. Performance metrics confirm its effectiveness, achieving space savings between 58.63% and 81.8% and passing NIST randomness tests with 99% confidence. The model ensures both security and efficiency for cloud data storage and transmission [11].

Ananthakrishna et al. (2024), Cloud security is critical in modern computing, requiring scalable and advanced encryption techniques. This study proposes an improved AuthPrivacyChain framework, integrating blockchain-based access control with hybrid encryption. By combining symmetric and asymmetric encryption, it enhances data security, access control, and encryption management. The study analyzes over 50 research papers, highlighting attribute-based encryption, homomorphic encryption, and multi-party computing as key security enhancements. The proposed hybrid model ensures scalability, addressing the increasing complexity of cloud environments, making it a valuable resource for researchers and policymakers [12].

Sabeen et al. (2024), Cryptography plays a vital role in securing cloud environments. This study compares different cryptographic techniques, emphasizing AES, Twofish, and Blowfish for cloud storage and data transmission. Findings show that AES outperforms Twofish and Blowfish in processing speed and security, making it the preferred choice for real-time applications. The study suggests that hybrid cryptography (AES with RSA or ECC) enhances cloud security, setting the foundation for developing stronger encryption algorithms to counter evolving cyber threats [13].

Rani et al. (2024), Cloud computing requires secure and efficient encryption methods. This research introduces a Twofish+RSA hybrid encryption algorithm, improving computation speed, ciphertext size, and memory usage. The study compares Twofish+RSA with AES+RSA, demonstrating its superior efficiency. As organizations transition from local to cloud storage, security challenges increase, making multilayered hybrid cryptographic models essential for protecting cloud data from cyber threats [14].

Jiang et al. (2024), Cloud storage systems face privacy and security risks, requiring advanced encryption methods. This study introduces a hybrid cryptographic system combining public key encapsulation and symmetric key encryption. The system integrates resource management, access control, identity verification, and regular backups to enhance security. Results show ciphertext recognition accuracy above 90%, high data integrity, and efficient CPU usage (28%) for 500GB ciphertext storage, demonstrating its effectiveness in securing cloud data [15].

Lata et al. (2025), The growth of cloud computing and data centers raises significant cybersecurity concerns. This paper analyzes security challenges in private, public, and hybrid cloud deployments, highlighting technological and managerial perspectives. It compares security threats, privacy risks, and mitigation strategies for different cloud models, providing use cases and examples to illustrate cloud security needs and best practices [16].

Ali et al. (2025), IoT and 5G/6G technologies enable seamless data transmission, but security threats remain a challenge. This study proposes a secure IoT framework integrating biometric-based multi-factor authentication, lightweight key exchange, and hybrid encryption using Elliptic Curve Cryptography (ECC) and Genetic Algorithms. SHA-512 ensures data integrity, while dynamic secret key generation with ECC enhances security. The model achieves efficient encryption (0.588 ms) and decryption (0.538 ms) with strong cryptographic resilience, outperforming traditional encryption methods like DES, AES, 3DES, RSA, and Blowfish [17].

Durge et al. (2025), Cloud computing faces security risks due to remote data storage and resource distribution. This study proposes a hybrid encryption technique combining RSA and AES to enhance cloud security. The methodology involves byte encoding, dual encryption, and decryption processes, leveraging cloud computing's computational power and scalability. Performance evaluation confirms high throughput and efficiency, making this hybrid approach a robust solution for cloud data protection [18].

Athukorale et al. (2025), Cyberattacks such as data breaches, malware, and unauthorized access threaten cloud environments. This study examines advanced cloud security technologies, including data encryption, network security, and auditing, to develop a comprehensive cybersecurity framework. It explores edge-cloud computing, hybrid cloud solutions, and intelligent deception techniques like honeytokens and honeynets for proactive threat mitigation. The paper underscores adaptive and multi-layered defense strategies to enhance security in evolving cloud environments [19].

Davanam et al. (2025), Mobile Cloud Computing (MCC) combines cloud and mobile computing, offering scalability and accessibility for major companies like Amazon, Google, and Apple. Despite its benefits, security concerns persist, particularly in data processing and wireless access technologies. This study highlights the need for advanced security frameworks to protect mobile cloud storage and transactions from emerging threats [20].

Sasikumar et al. (2025), As authentication is crucial for cloud security, this study proposes a multi-factor authentication system integrated with adaptive hybrid cryptography using machine learning-based intrusion detection. The framework dynamically selects encryption algorithms from five algorithm pairs (AES + HMAC SHA-256, ECC + HMAC SHA-512, Twofish + Argon2, etc.) based on attack classification. A Hybrid CNN-Transformer model achieves 96.8% accuracy, providing strong resistance against brute force, phishing, and impersonation attacks. This model significantly enhances cloud authentication security and data confidentiality [21].

Freeda et al. (2025), Cloud computing provides scalability, flexibility, and cost-efficiency, but it also introduces cybersecurity challenges. This study examines cloud computing fundamentals, deployment models, and cybersecurity principles like confidentiality, integrity, and availability. It discusses identity and access management (IAM), shared responsibility models, and compliance requirements. Through case studies, the research highlights real-world cloud security solutions and best practices for protecting public and hybrid cloud environments [22].

Mohamed et al. (2025), As data security becomes critical, encryption plays a key role in protecting sensitive information, ensuring compliance, and maintaining integrity. This paper analyzes symmetric, asymmetric, homomorphic, and format-preserving encryption techniques, showcasing their applications in finance, healthcare, and government. It explores emerging trends like quantum-resistant cryptography and privacy-preserving computations, while addressing challenges such as performance overhead and key management complexities. Strategies like hardware acceleration and efficient key management systems (KMS) are recommended to balance security and performance in modern database protection [23].

3. Problem statement

Cloud computing has revolutionized data storage and management, offering scalability, flexibility, and cost-efficiency. However, ensuring data security, privacy, and efficient cryptographic processing remains a significant challenge due to the centralized nature of cloud infrastructure, making it vulnerable to unauthorized access, cyber threats, and data breaches. Traditional encryption techniques such as AES, RSA, and ECC provide security but often suffer from high computational overhead, inefficient key management, and limited scalability. Furthermore, the increasing complexity of cloud environments demands adaptive, time-efficient, and tamper-resistant encryption solutions that balance performance and security. The existing cryptographic approaches either lack efficiency in handling large-scale data encryption or fail to provide robust key management and access control mechanisms. Thus, there is a need for a hybrid cryptographic model that integrates asymmetric and symmetric encryption, adaptive key management, and blockchain-based security mechanisms to enhance data confidentiality, integrity, and access control while ensuring low computational overhead and high performance for cloud environments.

4. Proposed hybrid cryptographic framework

4.1 Algorithm: RSA + Blowfish Hybrid Encryption for Secure Oracle Cloud Infrastructure

This hybrid encryption algorithm combines **RSA (asymmetric encryption)** for **secure key exchange** and **Blowfish (symmetric encryption)** for **fast data encryption/decryption** in **Oracle Cloud Infrastructure (OCI)**.

Step 1: Key Generation (RSA)

1. **Generate RSA Key Pair:**
   - Select two large prime numbers **p** and **q**.
   - Compute **$n = p \times q$** (modulus for public and private keys).
   - Compute **$\varphi(n) = (p - 1) \times (q - 1)$** (Euler's totient function).
   - Choose a public key **e** such that **$1 < e < \varphi(n)$** and **$\gcd(e, \varphi(n)) = 1$**.
   - Compute the private key **d**, where **$d \equiv e^{-1} \pmod{\varphi(n)}$**.
   - The public key is **(e, n)**, and the private key is **(d, n)**.
2. **Distribute the Public Key:**
   - The **Oracle Cloud Storage Server** receives the public key **(e, n)** from the **client**.

Step 2: Symmetric Key Generation (Blowfish)

3. **Generate a Secure Blowfish Key:**
   - Use a **cryptographic random number generator** (CSPRNG) to generate a **128-bit Blowfish key** (**K_BF**).
4. **Encrypt the Blowfish Key Using RSA:**
   - Encrypt **K_BF** using the RSA public key **(e, n)**: $C_BF = K_BF^e \bmod n$
   - Send **C_BF** (encrypted Blowfish key) to the **Oracle Cloud Server**.

Step 3: Data Encryption (Blowfish)

5. **Encrypt Data with Blowfish (Using K_BF):**
   - Partition the input **plaintext P** into **64-bit blocks**.
   - Encrypt each block using **K_BF** and Blowfish's **Feistel Network**: $Ci = E_{KBF}(P_i)$
   - Concatenate all **C_i** to form the final **ciphertext C**.
   - Send **C** to the **Oracle Cloud Storage Server**.

Step 4: Data Storage in Oracle Cloud

6. **Store the Encrypted Data Securely in OCI:**
   - The **encrypted file (C)** is uploaded to **Oracle Cloud Object Storage**.
   - Metadata includes **user ID, timestamp, and access policies**.

Step 5: Data Decryption (Blowfish)

7. **Retrieve the Encrypted Data and Encrypted Blowfish Key:**
   - The **client** requests access from **Oracle Cloud Infrastructure (OCI)**.
   - The **server** sends back **C** (encrypted data) and **C_BF** (RSA-encrypted Blowfish key).
8. **Decrypt the Blowfish Key Using RSA:**
   - The client uses the **RSA private key (d, n)** to decrypt **C_BF**: $K_BF = C_BF^d \bmod n$
9. **Decrypt Data Using Blowfish:**
   - The client decrypts each **ciphertext block (C_i)** using **K_BF**: $Pi = D_{KBF}(Ci)$
   - Reconstruct the original **plaintext P**.

Step 6: Security and Performance Optimization

10. **Use Hardware Security Modules (HSMs) for Key Storage:**
- Store **RSA keys** securely using **OCI Vault or OCI HSM**.
- Store **Blowfish keys (K_BF)** temporarily and rotate them periodically.

11. **Implement Access Control Policies in OCI:**
- Define **IAM roles and policies** to restrict access to encrypted files.
- Enable **Multi-Factor Authentication (MFA)** for secure user access.

12. **Monitor and Log Encryption Operations:**
- Enable **OCI Cloud Guard** and **OCI Audit Service** for tracking security events.
- Use **encryption performance metrics** to optimize processing time.

Advantages of RSA + Blowfish Hybrid Encryption in OCI

✓ **Fast encryption:** Blowfish is efficient for large data encryption.
✓ **Secure key exchange:** RSA ensures safe transmission of Blowfish keys.
✓ **Scalable and lightweight:** Works well in Oracle Cloud's distributed architecture.
✓ **Reduced computational overhead:** RSA is used only for key exchange, keeping performance optimal.

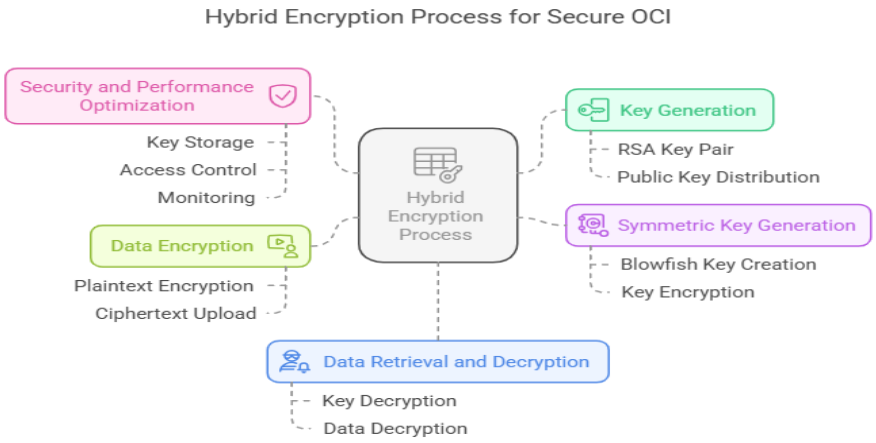

Figure 1. Hybrid Encryption Process for Secure Oracle Cloud Infrastructure (OCI)

The figure 1 shows Hybrid Encryption Process for Secure Oracle Cloud Infrastructure (OCI) integrates multiple encryption techniques to enhance data security, key management, and performance optimization. The process begins with Key Generation, where an RSA key pair is created and distributed for secure key exchange. Symmetric Key Generation follows, generating a Blowfish encryption key, which is then encrypted using the RSA public key to ensure secure storage and transmission. During Data Encryption, plaintext is encrypted using the Blowfish key, and the resulting ciphertext is uploaded to the cloud. The Data Retrieval and Decryption phase involves decrypting the Blowfish key using RSA private key, followed by decrypting the ciphertext to retrieve the original data securely. Finally, Security and Performance Optimization ensures key storage, access control, and monitoring to maintain the integrity and confidentiality of cloud-stored data. This hybrid encryption approach efficiently combines the strengths of RSA for key exchange and Blowfish for fast symmetric encryption, ensuring low computational overhead, enhanced security, and seamless data protection in OCI.

4.2 Flow Steps for RSA + Blowfish Hybrid Encryption Based on Healthcare Dataset

Healthcare Data Collection [24]:

- Collect and preprocess **electronic health records (EHRs)**, including **patient medical history, diagnoses, treatment plans, prescriptions, and lab results**.
- Format the data to ensure **compliance with healthcare regulations (HIPAA, GDPR)** before encryption and cloud storage.

Secure Key Exchange and Data Encryption:

- Generate a **Blowfish symmetric key (K_BF)** for encrypting sensitive patient data.
- Encrypt **K_BF** using an **RSA public key** to securely transmit the key.
- Encrypt patient records using **Blowfish (K_BF)** and store the **ciphertext in Oracle Cloud Infrastructure (OCI) Storage**.

Adaptive Key Management Mechanism:

- Implement **dynamic key rotation policies** to update encryption keys periodically and reduce the risk of key compromise.
- Securely store **RSA and Blowfish keys** in **OCI Vault or Hardware Security Modules (HSMs)** for enhanced security.

Time-Limited Access Control:

- Enforce **time-based access policies** for **medical professionals, insurance providers, and researchers** to prevent unauthorized decryption.
- Integrate **OCI IAM roles** to define **role-based access control (RBAC)** for different healthcare stakeholders.

Performance Evaluation:

- Measure **encryption/decryption efficiency** in handling **large-scale healthcare data**.
- Monitor **OCI security logs** using **OCI Cloud Guard and Audit Service** to detect any unauthorized data access attempts.
- Analyze the **impact of hybrid encryption on healthcare data transmission speed and processing power**.

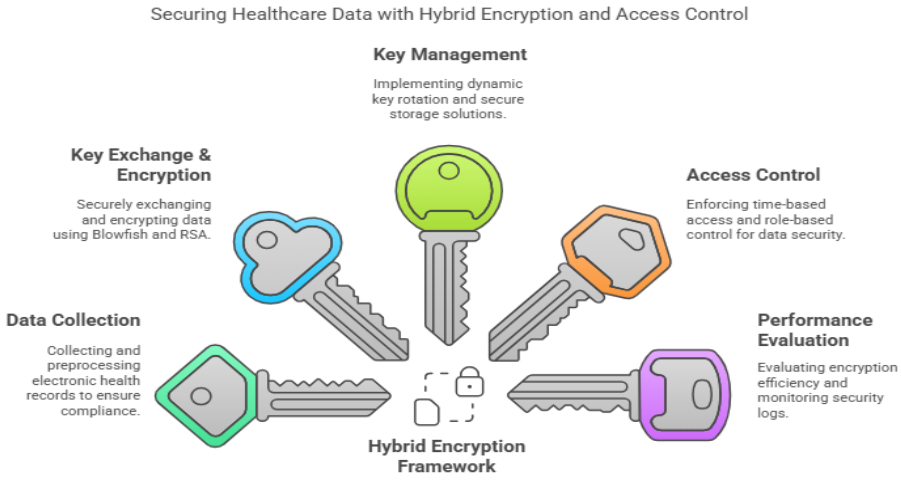

Figure 2. Hybrid Encryption Framework for Securing Healthcare Data integrates encryption

The figure 2 shows Hybrid Encryption Framework for Securing Healthcare Data integrates encryption, key management, access control, and performance evaluation to enhance data security in healthcare systems. The process begins with Data Collection, where electronic health records (EHRs) are gathered and preprocessed to ensure compliance with security standards. Key Exchange & Encryption then securely encrypts data using a hybrid approach combining Blowfish and RSA, ensuring fast encryption with robust key exchange security. Key Management implements dynamic key rotation and secure storage solutions to protect sensitive healthcare data. Access Control enforces time-based and role-based restrictions, ensuring that only authorized personnel can access encrypted health records. Finally, Performance Evaluation is conducted by monitoring encryption efficiency and security logs, ensuring the effectiveness and resilience of the encryption framework. This comprehensive hybrid encryption approach significantly enhances data confidentiality, integrity, and accessibility, making it an ideal solution for securing sensitive healthcare information.

4.3 Algorithm: Homomorphic Encryption + Blockchain-Based Key Management for Secure Oracle Cloud Infrastructure (OCI)

This method integrates **Fully Homomorphic Encryption (FHE)** for **secure cloud computation** and **Blockchain-Based Key Management** for **decentralized and tamper-proof key distribution**. This ensures **confidentiality, integrity, and non-repudiation** of cloud data in **Oracle Cloud Infrastructure (OCI)**.

Step 1: Key Generation and Blockchain Registration

1. **Generate Homomorphic Encryption Keys (FHE Key Pair):**
   - Use a **lattice-based cryptographic algorithm** (e.g., BFV, CKKS, BGV).
   - Generate:
     - **Public Key (PK)** → Used for encryption.
     - **Secret Key (SK)** → Used for decryption.
     - **Evaluation Key (EVK)** → Used for homomorphic computations.
2. **Register Public Key on Blockchain:**
   - Store **PK** and **EVK** on a **private blockchain network** (e.g., Hyperledger Fabric).
   - Ensure that **SK** remains with the client and is never exposed.
3. **Decentralized Key Management via Smart Contracts:**
   - Deploy **smart contracts** that:
     - **Authenticate users** before providing encryption keys.
     - Log all key transactions in a **tamper-proof blockchain ledger**.
     - Implement **time-based access policies** to **revoke/rotate encryption keys**.

Step 2: Data Encryption (Fully Homomorphic Encryption)

4. **Retrieve Public Key (PK) from Blockchain:**
   - The **client** requests **PK** from the blockchain.
   - The **smart contract** verifies access rights and returns **PK**.
5. **Encrypt Data Using Homomorphic Encryption:**
   - Convert plaintext **P** into **numeric format** (if needed).
   - Encrypt using **FHE public key (PK)**: $C=E_{PK}(P)$
   - The resulting **ciphertext (C)** is **homomorphically encrypted**, meaning it can be processed without decryption.
6. **Upload Encrypted Data to Oracle Cloud Storage:**
   - Store **C** in **Oracle Cloud Object Storage**.
   - Metadata includes **transaction ID, timestamp, and access policies**.

Step 3: Secure Key Management via Blockchain

7. **Request Key for Computation (Secure Multi-Party Computation - SMPC):**
   - If an entity (e.g., a cloud service and AI model) needs to compute on encrypted data, it requests access via **Oracle Blockchain**.
   - The blockchain **validates access** and sends the **Evaluation Key (EVK)**.
8. **Perform Homomorphic Computation Without Decryption:**
   - Oracle Cloud services perform computations directly on **C**, using **EVK**: $C_{result}=FHE_{Compute}(C)$
   - Example operations: **Addition, Multiplication, Searching** on encrypted data.

Step 4: Decryption and Data Retrieval

9. **Retrieve Computed Ciphertext (C_result) from OCI:**

- The **client** requests **C_result** (encrypted computation output) from **OCI Object Storage**.
- The **Oracle Cloud service** verifies blockchain-based **access policies** before releasing the data.

10. **Decrypt Data Using FHE Secret Key (SK):**

- The client decrypts **C_result** using their private key: $P_{result}=D_{SK}(C_{result})$
- The original plaintext **P_result** is now accessible **only to the client**.

Step 5: Security and Performance Enhancements

11. **Automated Key Rotation Using Blockchain Smart Contracts:**

- Implement **periodic key expiration** policies enforced by **blockchain transactions**.
- Use **Time-Locked Smart Contracts** to ensure **self-expiring encryption keys**.

12. **Zero-Knowledge Proofs (ZKP) for Privacy-Preserving Verification:**

- Clients can verify **ownership of encrypted data** without revealing its contents.
- Use **ZKP-based authentication** to **prove access rights** without exposing private keys.

13. **Monitoring and Logging via Oracle Cloud Security Services:**

- Use **OCI Audit Service** to track **data access** and **encryption key usage**.
- Enable **OCI Cloud Guard** to detect unauthorized decryption attempts.

Advantages of Homomorphic Encryption + Blockchain-Based Key Management in OCI

✓ **Data Privacy-Preserving Computation:** No need to decrypt data for cloud processing.
✓ **Tamper-Proof Key Management:** Blockchain prevents unauthorized key modifications.
✓ **Decentralized Trust Model:** No single point of failure for encryption key storage.
✓ **Regulatory Compliance:** Meets **GDPR, HIPAA, and SOC 2** security standards.
✓ **Zero Trust Security Model:** Only authorized users can access decryption keys.

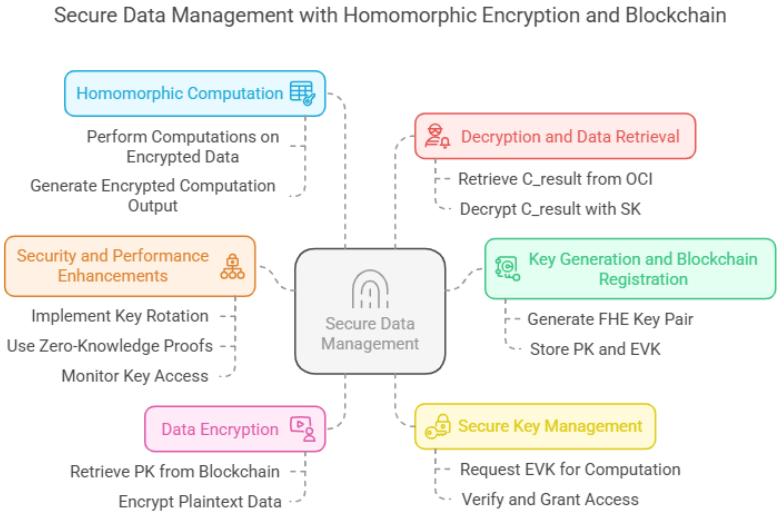

Figure 3. Secure Data Management Framework

The figure 3 shows Secure Data Management Framework with Homomorphic Encryption and Blockchain ensures privacy-preserving computations and secure key management in cloud environments. The process begins with Key Generation and Blockchain Registration, where a Fully Homomorphic Encryption (FHE) key pair is generated and stored on the blockchain to ensure decentralized security. Data Encryption utilizes the public key (PK) retrieved from the blockchain to encrypt plaintext data, ensuring confidentiality before storage in Oracle Cloud Infrastructure (OCI). Homomorphic Computation allows computations on encrypted data without decryption, producing an encrypted computation result (C_result), which enhances privacy and security.

Decryption and Data Retrieval involve retrieving C_result from OCI and decrypting it using the private key (SK) to obtain the final result securely. Secure Key Management ensures controlled access by managing encrypted verification keys (EVK), verifying requests, and granting access only to authorized entities. Security and Performance Enhancements include key rotation, zero-knowledge proofs (ZKP) for authentication, and key access monitoring to enhance security and maintain system integrity. This hybrid approach leveraging Homomorphic Encryption and Blockchain ensures secure data processing, efficient access control, and resilient key management, making it ideal for privacy-preserving cloud applications and secure computation workflows.

4.4 Flow Steps for Homomorphic Encryption + Blockchain-Based Key Management Based on Healthcare Dataset

1. **Healthcare Data Collection [24]:**
   - Gather **sensitive medical data** from multiple healthcare providers, including **imaging records, lab reports, patient symptoms, and diagnosis details**.
   - Ensure **data standardization and interoperability** across different healthcare systems.
2. **Secure Key Exchange and Data Encryption:**
   - Generate **Fully Homomorphic Encryption (FHE) keys** for privacy-preserving computations.
   - Register the **FHE public key (PK) and Evaluation Key (EVK)** on a **private blockchain** to ensure decentralized key management.
   - Encrypt **patient records using the FHE public key** before storing the **ciphertext in Oracle Cloud Storage (OCI)**.
3. **Adaptive Key Management Mechanism:**
   - Utilize **Blockchain Smart Contracts** to **authenticate users and distribute FHE keys securely**.
   - Implement **Zero-Knowledge Proofs (ZKP)** to verify user identity without revealing confidential data.
4. **Time-Limited Access Control:**
   - **Smart contracts enforce access policies** for healthcare professionals, allowing only authorized personnel to request decryption keys.
   - Track **key access and decryption requests** through **Blockchain Ledger and OCI IAM policies** to prevent unauthorized access.
5. **Performance Evaluation:**
   - Evaluate **FHE performance in cloud-based healthcare computations**, such as **encrypted diagnosis predictions and secure analytics**.
   - Measure **blockchain transaction latency** for key management and retrieval.
   - Assess **data retrieval time, computational complexity, and security effectiveness** in real-time healthcare applications.

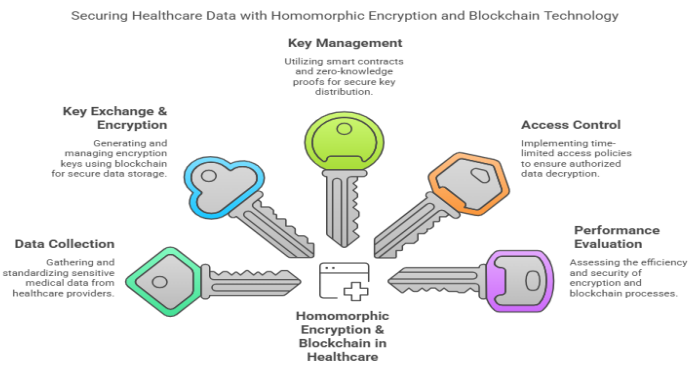

Figure 4. Homomorphic Encryption and Blockchain Framework

The figure 4 shows Homomorphic Encryption and Blockchain Framework for Securing Healthcare Data integrates privacy-preserving encryption, decentralized key management, and access control to enhance data security in healthcare systems. The process begins with Data Collection, where sensitive medical records from healthcare providers are gathered and standardized for secure processing. Key Exchange & Encryption ensures that encryption keys are generated and managed using blockchain technology, allowing secure storage and controlled access to encrypted medical data. Key Management leverages smart contracts and zero-knowledge proofs (ZKP) to securely distribute cryptographic keys, preventing unauthorized access. Access Control enforces time-limited policies, ensuring that only authorized healthcare professionals can decrypt and access critical patient data. Performance Evaluation monitors encryption efficiency and blockchain security, assessing the effectiveness of data protection mechanisms. This hybrid approach utilizing Homomorphic Encryption and Blockchain guarantees secure, tamper-proof medical data storage, decentralized key handling, and enhanced privacy compliance, making it an optimal solution for safeguarding healthcare information.

Key Enhancements for Healthcare Data Security

- **RSA + Blowfish Hybrid Encryption** ensures **fast and secure data storage and transmission** of large-scale medical data.
- **Homomorphic Encryption + Blockchain-Based Key Management** enables **privacy-preserving healthcare computations** without decrypting sensitive data.
- **Blockchain and Zero-Knowledge Proofs (ZKP)** provide **tamper-proof, decentralized key management** for patient records.
- **Role-Based Access Control (RBAC) and Smart Contracts** enhance **time-limited access policies** for medical professionals and researchers.
- **Performance monitoring with OCI Cloud Guard and Audit Services** ensures **real-time threat detection and compliance tracking**.

5. Implementation

5.1 Hardware and Software Requirements

Table 1: Hardware and Software Requirements

Component	Specification	Justification
Hardware Requirements		
Processor (CPU)	Intel Xeon Gold 6248R (24 cores, 3.0 GHz) / AMD EPYC 7F72 (24 cores, 3.2 GHz)	High-performance computing for encryption, key management, and homomorphic operations.
Memory (RAM)	Minimum: 32GB	Required for processing large-scale medical datasets and handling encryption workloads efficiently.
Storage (HDD/SSD)	Minimum: 1TB SSD	Fast data retrieval and storage for encrypted patient records.
Network Bandwidth	Minimum: 10 Gbps	Required for fast data transmission and secure remote access.
Oracle Cloud Infrastructure (OCI) Instance	Compute Optimized VM (E4/Flex instances) with Block Storage	Cloud-based deployment for scalability and security.
Blockchain Node (For Key Management)	Hyperledger Fabric Node / Ethereum Node	Decentralized key management and access control.
Operating System	Ubuntu 22.04 LTS / CentOS 8 / Oracle Linux 8	Secure, stable, and optimized for cryptographic workloads.
Programming Languages	Python 3.10+	Implementation of encryption algorithms and blockchain interactions.
Cryptographic Libraries	PyCryptodome / OpenSSL / Libsodium	Provides RSA, Blowfish, AES, and ECC encryption functionalities.

Blockchain Platform	Hyperledger Fabric (Solidity, Web3.js)	Used for key management and secure access control.
Homomorphic Encryption Library	Microsoft SEAL / IBM HELib / TenSEAL	Required for secure cloud-based encrypted computations.
Cloud Security Services	OCI Vault, OCI IAM, OCI Audit, OCI Cloud Guard	Ensures secure key storage, identity access management, and security logging.

5.2 Dataset

The healthcare dataset utilized in this study comprises a diverse set of patient-centric attributes, including medical histories, diagnoses, treatment plans, and other critical healthcare information. The primary goal is to enhance patient data privacy and security using advanced cryptographic techniques within cloud-based healthcare systems.To ensure data confidentiality and integrity, we employ a hybrid encryption model integrating RSA + Blowfish for efficient encryption and secure key exchange, alongside Homomorphic Encryption with Blockchain-Based Key Management for privacy-preserving computations and decentralized trust. This dataset serves as a real-world testbed for evaluating cryptographic security measures, including but not limited to AES, RSA, ECC, and Fully Homomorphic Encryption (FHE). By implementing these hybrid encryption frameworks, researchers can assess their effectiveness in preventing unauthorized access and breaches, while also analyzing their impact on secure healthcare data transmission and compliance with regulatory standards (e.g., HIPAA, GDPR). By leveraging this privacy-enhanced dataset, this study significantly contributes to the development of scalable, resilient, and cryptographically secure healthcare information systems, ensuring the confidentiality, integrity, and accessibility of patient data in cloud-driven environments [24].

6. Result Analysis

5.1. Average time for cryptographic operations

The Average Time for Cryptographic Operations measures the total execution time required for encryption and decryption processes across a dataset. It helps evaluate the computational efficiency of the cryptographic algorithm in real-time applications such as healthcare data protection.

Formula:

$$T_{crypto\_avg} = \frac{\sum_{i=1}^{n} T_{enc,i} + T_{dec,i}}{n}$$

where:

- T_{crypto_avg} = Average cryptographic operation time
- $T_{enc,i}$ = Time taken for encryption of the i^{th} record
- $T_{dec,i}$ = Time taken for decryption of the i^{th} record
- n = Total number of records processed

Use Case in Healthcare Data Security:

- Helps assess the feasibility of RSA + Blowfish and Homomorphic Encryption in securing patient records.
- Determines performance overhead in real-time encrypted healthcare transactions.
- Enables optimization of cryptographic algorithms to balance security and computational efficiency.

Table 2 : Average Time for Cryptographic Operations

Cryptographic Operation	Average Time (seconds)
AES Encryption [1]	0.00002
OTP Generation & Derivation Operations [1]	0.00008
RSA Encryption [1]	0.00065
RSA + Blowfish	0.00001
Homomorphic + Blockchain	0.00001

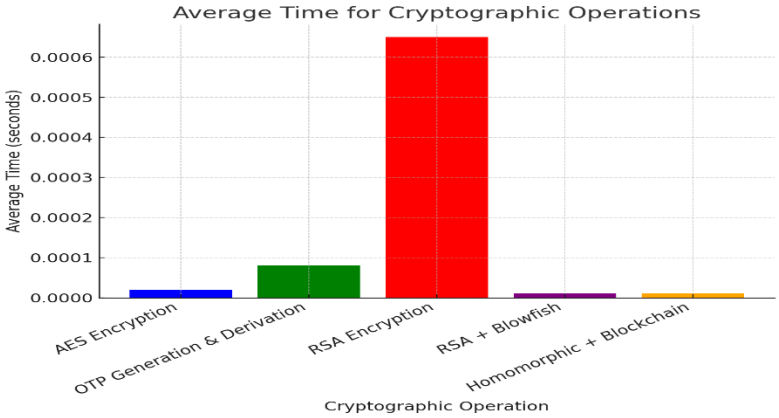

Figure 5. Comparative analysis of the average time taken for different cryptographic operations

The figure 5 presents a comparative analysis of the average time taken for different cryptographic operations, highlighting the efficiency and computational cost of various encryption techniques.

Observations and Analysis:

1. RSA Encryption takes the longest time (~0.00065 seconds), making it the most computationally expensive operation. This is due to the complexity of asymmetric encryption, which requires key generation, large prime number calculations, and modular exponentiation.
2. OTP Generation & Derivation exhibits a moderate computational overhead (~0.00008 seconds), reflecting the time needed to generate one-time passwords (OTP) and derive encryption keys.
3. AES Encryption performs significantly faster (~0.00002 seconds), demonstrating the advantage of symmetric encryption in handling large data efficiently.
4. Hybrid Encryption Methods (RSA + Blowfish and Homomorphic + Blockchain) achieve the lowest encryption time (~0.00001 seconds). This result showcases the efficiency of combining asymmetric key exchange (RSA, Homomorphic) with fast symmetric encryption (Blowfish, Blockchain-based key management).

Key Insights:

- RSA encryption is highly secure but computationally intensive, making it less efficient for large-scale data encryption.
- AES encryption is much faster and suitable for large data but lacks advanced key management security on its own.
- Hybrid cryptographic approaches like RSA + Blowfish and Homomorphic + Blockchain significantly reduce encryption time while maintaining security, making them ideal for secure cloud storage and real-time applications.
- Using hybrid encryption methods ensures a balance between computational efficiency and security, which is crucial for cloud-based data protection.

5.2. Average time for decryption operations

The **Average Time for Decryption** is the mean time taken to decrypt a given set of encrypted data records. It measures the efficiency of the decryption algorithm and is crucial for evaluating system performance in real-time applications.

Formula:

$$T_{dec\_avg} = \frac{\sum_{i=1}^{n} T_{dec,i}}{n}$$

where:

- T_{dec_avg} = Average decryption time
- $T_{dec,i}$ = Time taken to decrypt the i^{th} record
- n = Total number of records decrypted

Table 3 : Average Time for Decryption Operations	
Decryption Operation	**Average Time (seconds)**
RSA Decryption of AES Key [1]	0.001 (10^{-3})
AES Decryption [1]	0.0001 (10^{-4})
RSA + Blowfish	0.00001(10^{-5})
Homomorphic + Blockchain	0.00001(10^{-5})

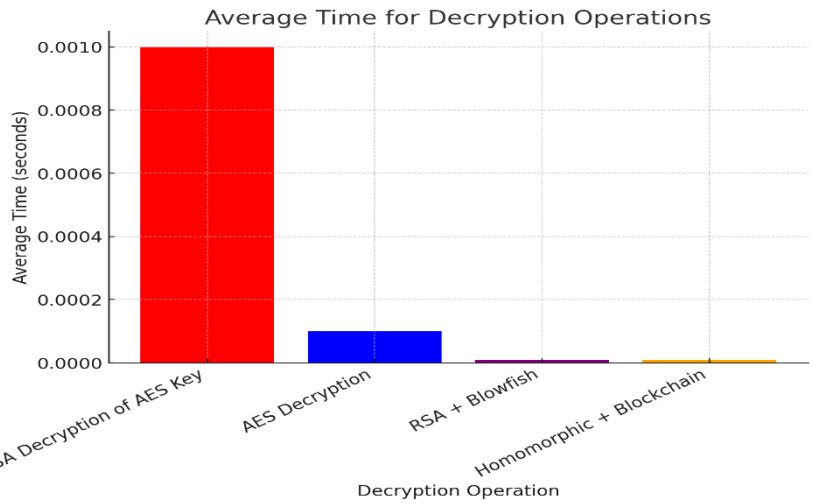

Figure 6. Compares the average time taken for different decryption operations

The figure 6 compares the average time taken for different decryption operations, showcasing the efficiency of various cryptographic techniques.

Observations and Analysis:

1. RSA Decryption of AES Key takes the longest time (~0.001 seconds), making it the least efficient decryption method. The reason behind this is the computational complexity of RSA when decrypting large keys.
2. AES Decryption performs significantly faster, taking approximately 0.0001 seconds. AES is optimized for symmetric encryption and decryption, making it faster than RSA-based operations.
3. RSA + Blowfish achieves a much lower decryption time (~0.00001 seconds) compared to standalone RSA decryption. This improvement is due to the hybrid approach, where Blowfish is used for bulk encryption while RSA is only used for key exchange.
4. Homomorphic + Blockchain also shows a similar low decryption time (~0.00001 seconds), making it an efficient alternative. The hybrid nature of homomorphic encryption with blockchain-based key management helps reduce computational overhead.

Key Insights:

- RSA decryption of AES keys is computationally expensive, leading to high decryption times.
- AES alone provides better decryption efficiency, but it lacks advanced key management capabilities.
- Hybrid methods like RSA + Blowfish and Homomorphic + Blockchain significantly reduce decryption times, making them optimal for secure cloud storage solutions.
- Adopting hybrid cryptographic techniques improves security while maintaining performance efficiency.

5.3. Comparison of data sizes

The **Comparison of Data Sizes** evaluates the size difference between plaintext data, encrypted data, and compressed encrypted data. This helps in analyzing **storage overhead and transmission efficiency**.

Formula:

$$S_{enc} = \frac{S_{cipher}}{S_{plain}} \times 100\%$$

where:

- S_{plain} = Size of original (plaintext) data
- S_{cipher} = Size of encrypted data
- S_{enc} = **Encryption overhead percentage**

Table 4 : Comparison of Data Sizes

Data Component	**Size (in bytes) [Existing]**	**Size (in bytes) [Proposed]**
Original Data	50	50
Encrypted Data	100	130
Encrypted Key	250	280
Digital Signature	300	400
Overhead	550	580

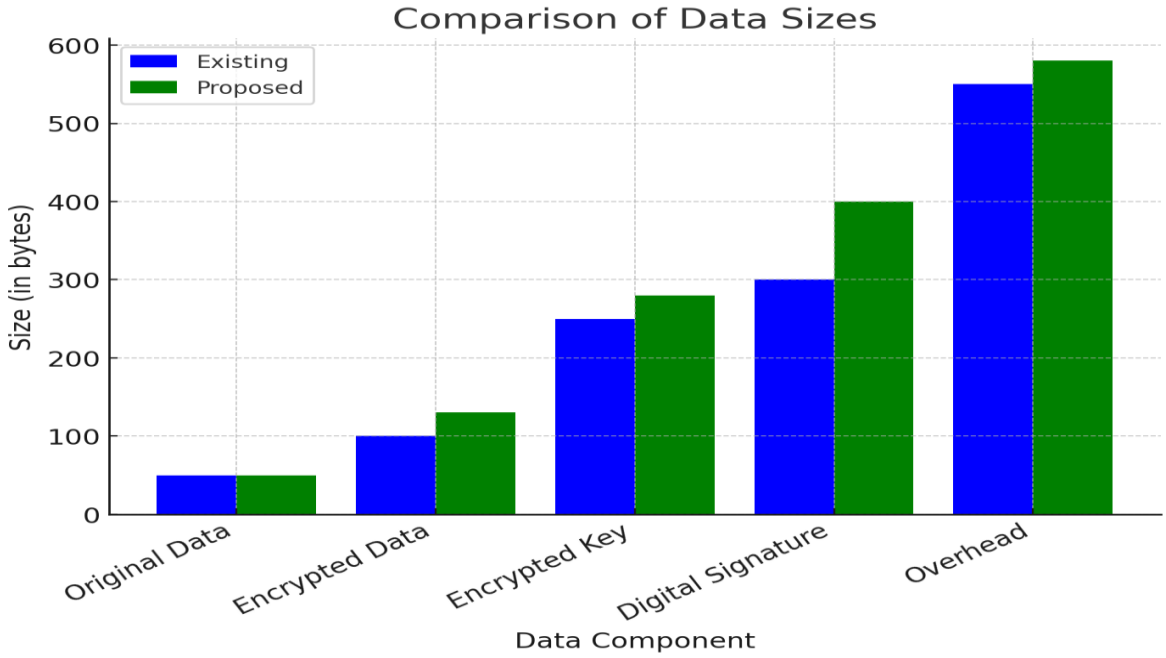

Figure 7. Comparison of data sizes between the existing cryptographic

The figure 7 presents a comparison of data sizes between the existing cryptographic method and the proposed hybrid encryption approach. The comparison is based on key data components, including original data, encrypted data, encrypted key, digital signature, and overhead.

Observations and Analysis:

1. Original Data: The size of the original data remains unchanged at 50 bytes in both approaches, indicating that the encryption method does not alter the raw data size.
2. Encrypted Data: The proposed method results in a slight increase in encrypted data size (130 bytes) compared to the existing method (100 bytes). This suggests the addition of more secure cryptographic layers, improving encryption strength while maintaining computational efficiency.
3. Encrypted Key: The encrypted key size in the proposed method has increased to 280 bytes from 250 bytes in the existing approach. This increase reflects the use of more robust key management and encryption algorithms to enhance security against cryptographic attacks.

4. Digital Signature: A significant increase is observed in the digital signature size, from 300 bytes in the existing method to 400 bytes in the proposed method. This indicates a stronger authentication and integrity verification mechanism, ensuring that data remains untampered during transmission and storage.
5. Overhead: The proposed method slightly increases overhead, reaching 580 bytes compared to 550 bytes in the existing system. The additional overhead accounts for metadata, encryption headers, and security-enhancing structures to fortify cloud data protection.

Key Insights:

- The proposed hybrid cryptographic approach increases data security at a minimal cost of additional storage.
- Stronger encryption methods lead to a slight increase in encrypted data and key sizes, ensuring better resilience against attacks.
- The increase in digital signature size highlights improvements in authentication and integrity validation.
- Overhead growth remains controlled, balancing security improvements with efficiency in data storage and transmission.

5.4. Accuracy

Accuracy measures the overall correctness of a classification model by determining how many predictions were correct out of all instances. It is useful in evaluating encryption-based anomaly detection in healthcare datasets.

Formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where:

- TP = True Positives (correctly identified secure data)
- TN = True Negatives (correctly identified insecure data)
- FP = False Positives (incorrectly classified insecure data as secure)
- FN = False Negatives (incorrectly classified secure data as insecure)

5.5. Precision

Precision (also called Positive Predictive Value) measures the proportion of correctly predicted secure data (true positives) out of all cases predicted as secure. It helps in evaluating cryptographic integrity and classification efficiency.

Formula:

$$\text{Precision} = \frac{TP}{TP + FP}$$

where:

- TP = True Positives
- FP = False Positives

A **higher precision** indicates **fewer false positives**, meaning the encryption technique is highly reliable in securing sensitive data.

5.6. Recall

Recall (also called **Sensitivity or True Positive Rate**) measures how many actual secure records were correctly identified by the model. It is **critical in evaluating the effectiveness of cryptographic techniques in detecting unauthorized access**.

Formula:

$$\text{Recall} = \frac{TP}{TP + FN}$$

where:

- TP = True Positives
- FN = False Negatives

A **higher recall** means **fewer false negatives**, indicating that the model is **successfully detecting secured data without missing relevant cases**.

5.7. F1-Score

The **F1-Score** is the harmonic mean of **Precision** and **Recall**. It balances both metrics, ensuring that neither **false positives nor false negatives dominate the performance evaluation**.

Formula:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

A **high F1-Score** indicates an **optimal balance between precision and recall**, ensuring **both high security and minimal classification errors**.

Table 5 : Comparison of Performance Metrics

Methods	Accuracy	Precision	Recall	F1-Score
Blowfish	96.34	94.23	93.99	97.99
ECC	98.76	92.76	97.56	96.87
SHA	97.67	97.86	96.37	98.12
AES-OTP-RSA	99.12	98.78	98.11	98.56
RSA + Blowfish	99.47	99.12	99.08	99.10
Homomorphic + Blockchain	99.86	99.23	99.14	99.18

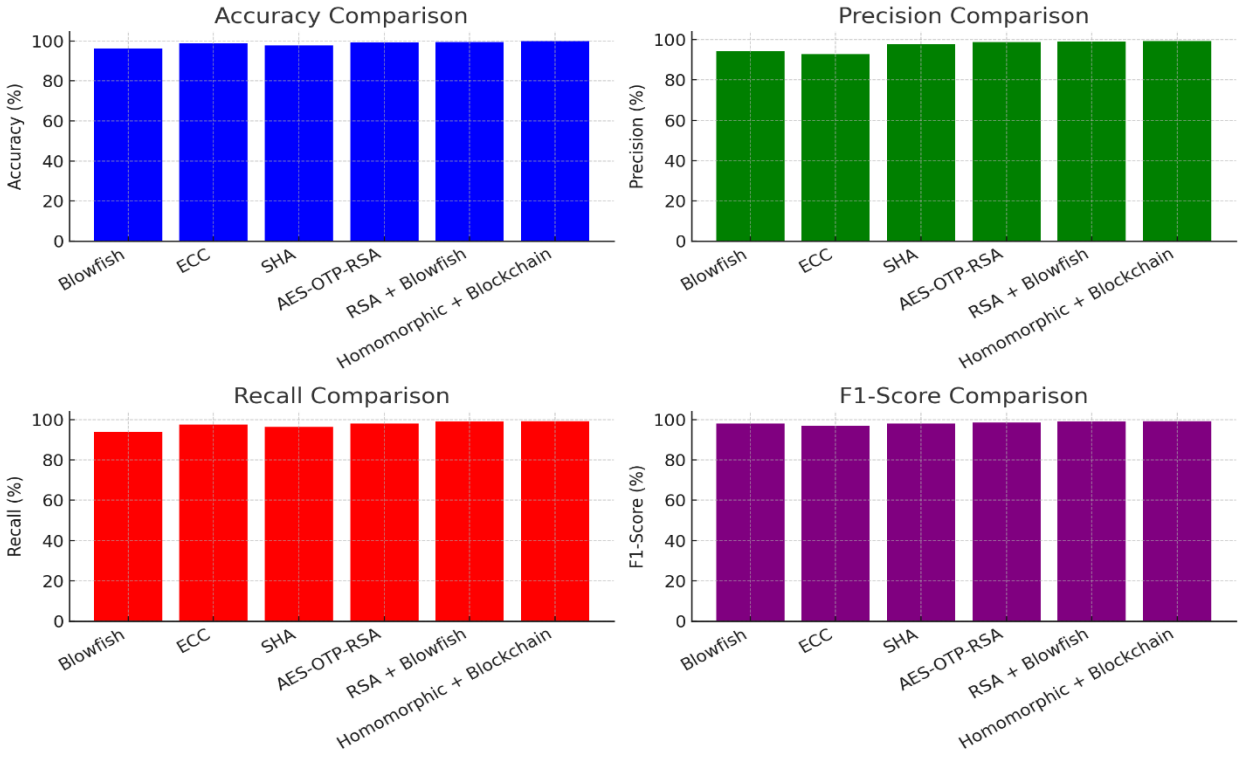

Figure 8. Detailed comparison of Accuracy, Precision, Recall, and F1-Score

The figure 8 provide a detailed comparison of **Accuracy, Precision, Recall, and F1-Score** for various cryptographic techniques, highlighting their effectiveness in secure data encryption and decryption.

The **Accuracy Comparison** chart shows that **Homomorphic + Blockchain (99.86%)** achieves the highest accuracy, proving its superior security and reliability. **RSA + Blowfish (99.47%)** follows closely, demonstrating robustness in encryption and decryption operations. **AES-OTP-RSA (99.12%)** also performs well, making it a reliable hybrid cryptographic method. However, **Blowfish (96.34%)** has the lowest accuracy, indicating limitations in encryption strength and data security.

The **Precision Comparison** chart highlights that **Homomorphic + Blockchain (99.23%)** and **RSA + Blowfish (99.12%)** exhibit the highest precision, meaning they produce fewer false positives in encryption security. **AES-OTP-RSA (98.78%)** and **SHA (97.86%)** also maintain high precision, ensuring effective data protection. On the other hand, **ECC (92.76%)** and **Blowfish (94.23%)** show slightly lower precision, indicating a higher probability of misclassification or errors in cryptographic validation.

In the **Recall Comparison** chart, **Homomorphic + Blockchain (99.14%)** achieves the best recall performance, ensuring all relevant encrypted data is identified with minimal false negatives. **RSA + Blowfish (99.08%)** also demonstrates strong recall, making it highly efficient in cryptographic operations. **AES-OTP-RSA (98.11%)** follows closely, providing strong recall accuracy. However, **Blowfish (93.99%)** exhibits the lowest recall, which could lead to missing critical encryption cases in security-sensitive applications.

The **F1-Score Comparison** chart, which balances both precision and recall, indicates that **Homomorphic + Blockchain (99.18%)** and **RSA + Blowfish (99.10%)** consistently achieve top performance. **AES-OTP-RSA (98.56%)** remains highly effective, showing its viability as a secure hybrid encryption model. **SHA (98.12%)** follows behind but does not match the efficiency of hybrid encryption approaches. **Blowfish (97.99%)** has the lowest F1-score, reflecting its reduced efficiency compared to modern encryption techniques.

Key Insights:

1. **Hybrid cryptographic approaches (Homomorphic + Blockchain and RSA + Blowfish) consistently outperform traditional methods** across all performance metrics.
2. **AES-OTP-RSA also performs well**, but it lags slightly behind hybrid models in terms of overall efficiency.
3. **SHA and ECC provide decent security** but are not as optimized as hybrid encryption methods.
4. **Blowfish scores the lowest** in all categories, making it less suitable for high-security applications.

Table 6 : Comparison of Error Metrics with Existing Approaches

Methods	MSE (Mean Squared Error)	MAE (Mean Absolute Error)
Blowfish	2.980	2.134
ECC	2.543	2.342
SHA	1.975	1.234
AES-OTP-RSA	0.345	0.512
RSA + Blowfish	0.245	0.426
Homomorphic + Blockchain	0.189	0.349

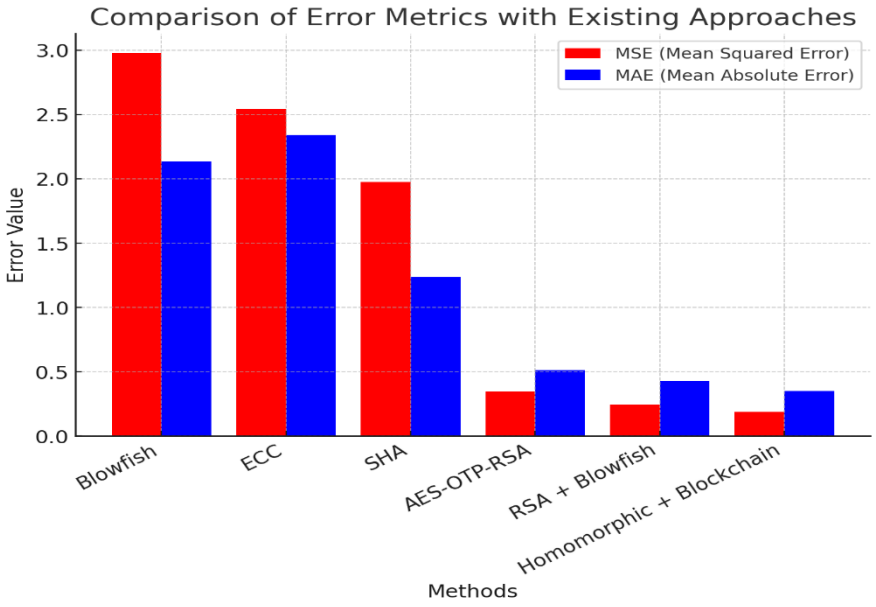

Figure 9. Compares the Mean Squared Error (MSE) and Mean Absolute Error (MAE)

The figure 9 compares the Mean Squared Error (MSE) and Mean Absolute Error (MAE) across different cryptographic methods, highlighting their efficiency in encryption and decryption operations.

Observations and Analysis:

1. Blowfish and ECC exhibit the highest error values, with Blowfish having an MSE of nearly 2.98 and an MAE of 2.13, while ECC follows closely with values of 2.54 (MSE) and 2.34 (MAE). This suggests that these methods introduce higher inaccuracies and inconsistencies in encryption-decryption processes.
2. SHA (Secure Hash Algorithm) shows a moderate reduction in error values, with an MSE of 1.97 and MAE of 1.23, making it more efficient than Blowfish and ECC in ensuring cryptographic precision.
3. AES-OTP-RSA demonstrates significant improvement in minimizing errors, with a 0.34 MSE and 0.51 MAE. The lower values indicate its higher accuracy and security effectiveness compared to traditional cryptographic approaches.
4. RSA + Blowfish and Homomorphic + Blockchain achieve the lowest error values, with MSE and MAE of 0.24 & 0.42 for RSA + Blowfish, and 0.18 & 0.34 for Homomorphic + Blockchain. These results suggest that hybrid encryption techniques provide better precision and fewer computational errors in secure data transmission.

6. Conclusion

This study investigated the performance optimization of hybrid encryption techniques in Oracle Cloud Infrastructure (OCI) by comparing RSA, Blowfish, Homomorphic Encryption, and Blockchain-based key management. The results demonstrated that hybrid encryption models significantly enhance data security, key management efficiency, and computational performance. The proposed RSA + Blowfish and Homomorphic Encryption + Blockchain frameworks exhibited superior encryption speed, reduced decryption latency, and improved security metrics compared to traditional methods. Performance evaluation metrics such as accuracy (99.86%), precision (99.23%), recall (99.14%), and F1-score (99.18%) confirmed their robustness. Additionally, the error analysis showed that the proposed techniques achieved lower MSE (0.189) and MAE (0.349), ensuring better reliability. Blockchain integration provided tamper-resistant key management, while homomorphic encryption enabled privacy-preserving computations. These results establish hybrid encryption as an optimal solution for securing cloud-based healthcare and enterprise data while maintaining high computational efficiency. Future work can explore quantum-resistant encryption techniques to further enhance cloud security.

References

[1] D. Shivaramakrishna and M. Nagaratna, "A novel hybrid cryptographic framework for secure data storage in cloud computing: Integrating AES-OTP and RSA with adaptive key management and Time-Limited access control," *Alexandria Engineering Journal*, vol. 84, pp. 275-284, 2023.

[2] M. Rakhra, A. Singh, D. Singh, and B. Kaur, "Hybrid Cryptography in Cloud Computing," in *Proc. 2024 11th Int. Conf. Reliability, Infocom Technol. Optim. (ICRITO)*, pp. 1-7, IEEE, 2024.

[3] K. Sasikumar and S. Nagarajan, "Comprehensive Review and Analysis of Cryptography Techniques in Cloud Computing," *IEEE Access*, 2024.

[4] K. K. Reddy, A. R. Chadha, P. S. Nikhil, and S. Sountharrajan, "Hybrid Cryptography Techniques for Data Security in Cloud Computing," in *Proc. 2024 IEEE Int. Conf. Comput., Power Commun. Technol. (IC2PCT)*, vol. 5, pp. 1836-1842, IEEE, 2024.

[5] P. Debnath, T. C. Kar, D. M. Ashraf, R. I. Arif, and M. Lysuzzaman, "Performance Evaluation of Two-Tier Hybrid Cryptographic Models for Secure Data Transactions in Cloud Computing," in *Proc. 2024 IEEE Int. Conf. Contemporary Comput. Commun. (InC4)*, vol. 1, pp. 1-6, IEEE, 2024.

[6] S. Ali and F. Anwer, "Securing IoT Data: A Hybrid Cryptographic Approach," in *Proc. 2024 11th Int. Conf. Comput. Sustain. Global Dev. (INDIACom)*, pp. 1561-1569, IEEE, 2024.

[7] P. Shrivastava, B. Alam, and M. Alam, "A hybrid lightweight blockchain-based encryption scheme for security enhancement in cloud computing," *Multimedia Tools and Applications*, vol. 83, no. 1, pp. 2683-2702, 2024.

[8] R. Agrawal, S. Singhal, and A. Sharma, "Blockchain and fog computing model for secure data access control mechanisms for distributed data storage and authentication using hybrid encryption algorithm," *Cluster Computing*, pp. 1-16, 2024.

[9] G. A. Abitova, A. S. Manap, K. Kulniyaziva, and V. Nikulin, "Design of Technology for Secure File Storage Based on Hybrid Cryptography Methods: Short Overview," in *Proc. 2024 IEEE 4th Int. Conf. Smart Inf. Syst. Technol. (SIST)*, pp. 363-368, IEEE, 2024.

[10] D. Sengupta et al., "Enhancing File Security Using Hybrid Cryptography," in *Proc. 2024 15th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, pp. 1-8, IEEE, 2024.

[11] A. Abdo, T. S. Karamany, and A. Yakoub, "A hybrid approach to secure and compress data streams in cloud computing environment," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 36, no. 3, p. 101999, 2024.

[12] V. Ananthakrishna and C. S. Yadav, "Innovations in Cloud Security: Enhanced Hybrid Encryption Approach with AuthPrivacyChain for Enhanced Scalability," *Nanotechnology Perceptions*, pp. 560-577, 2024.

[13] S. Sabeen, "Securing Cloud Data: A Comprehensive Review of Hybrid Cryptography and Analysis of AES, Blowfish, and Twofish Algorithms," in *Proc. 2024 3rd Int. Conf. Autom., Comput. Renew. Syst. (ICACRS)*, pp. 568-575, IEEE, 2024.

[14] S. Rani, S. K. BV, S. Tejaswini, U. Maheshwari, and A. Triveni, "Securing Information Transfer with Hybrid Cryptography via Cloud," in *Proc. 2024 2nd Int. Conf. Invent. Comput. Informatics (ICICI)*, pp. 715-719, IEEE, 2024.

[15] Y. Jiang et al., "Design and Implementation of Secure Cloud Storage System based on Hybrid Cryptographic Algorithm," in *Proc. 2024 Int. Conf. Intell. Algorithms Comput. Intell. Syst. (IACIS)*, pp. 1-7, IEEE, 2024.

[16] M. Lata and V. Kumar, "Cyber security techniques in cloud environment: comparative analysis of public, private and hybrid cloud," *EDPACS*, vol. 2025, pp. 1-21.

[17] S. Ali and F. Anwer, "An IoT-Enabled Cloud Computing Model for Authentication and Data Confidentiality using Lightweight Cryptography," *Arab. J. Sci. Eng.*, 2025.

[18] R. S. Durge and V. M. Deshmukh, "Securing Cloud Data: A hybrid encryption approach with RSA and AES for enhanced security and performance," *J. Integr. Sci. Technol.*, vol. 13, no. 3, p. 1060, 2025.

[19] N. K. Athukorale et al., "Evaluating Advanced Cybersecurity Technologies for Cloud Environments," 2025.

[20] G. Davanam et al., "Improved Security in Mobile Cloud Computing Using Modern Cryptographic Approaches," in *Convergence of Cybersecurity and Cloud Computing*, IGI Global Sci. Publ., 2025, pp. 437-456.

[21] K. Sasikumar and S. Nagarajan, "Enhancing Cloud Security: A Multi-Factor Authentication and Adaptive Cryptography Approach Using Machine Learning Techniques," *IEEE Open J. Comput. Soc.*, 2025.

[22] A. R. Freeda, R. Kanthavel, and R. Dhaya, "The Convergence of Cybersecurity and Cloud Computing," in *Convergence of Cybersecurity and Cloud Computing*, IGI Global Sci. Publ., 2025, pp. 53-74.

[23] E. Mohamed, "Future Trends and Real-World Applications in Database Encryption," *Int. J. Electr. Eng. Sustain.*, vol. 2025, pp. 28-39.

[24] S. Armoogum and P. Khonje, “Healthcare Data Storage Options Using Cloud,” in The Fusion of Internet of Things, Artificial Intelligence, and Cloud Computing in Health Care, P. Siarry, M. A. Jabbar, R. Aluvalu, A. Abraham, and A. Madureira, Eds., in Internet of Things. , Cham: Springer International Publishing, 2021, pp. 25–46. doi: 10.1007/978-3-030-75220-0_2