Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 2 (2025)

Secure and Efficient Data Hiding in the Cloud with Quantum-Resistant Encryption

Mrs. Salma Banu[1], Dr. Vijayalaxmi Biradar[2], Arshad Ahmad Khan Mohammad[3],

[1]Research Scholar, Kaling University, salmabanuphdece@gmail.com

[2]Associate Professor, Kalinga University

[3]Assistant Professor, GITAM Deemed to be University

Article History:

Received: *18-07-2024*

Revised: *09-09-2024*

Accepted: *29-09-2024*

Abstract:

Traditional steganographic methods in multi-cloud environments face significant challenges in cloud security, including complex key management, high computational overhead, and vulnerabilities to various attacks regaring secure data concealment. This paper presents a novel steganographic mechanism tailored for single cloud environments that address these issues. The proposed mechanism leverages a combination of homomorphic encryption, elliptic curve cryptography (ECC), and lattice-based cryptographic techniques to ensure robust security against classical and quantum attacks. Confining operations to a single cloud simplifies credential and key management, reducing operational complexity and enhancing system efficiency. The proposed approach mitigates the risks associated with multi-cloud environments and offers strong resistance to brute-force, differential, and steganalysis attacks. The implementation and validation of the mechanism demonstrate its efficacy in maintaining data integrity, confidentiality, and availability while minimizing computational resources. The results indicate that this approach significantly improves the security and performance of steganographic operations in cloud storage environments, setting a new standard for secure data concealment in cloud computing.

Keywords: component, Cloud, Quantum-Resistant Encryption

1. Introduction

The rise of cloud computing has changed how data is stored and managed, making it easier to reach, scale, and be flexible than ever before [1]. But the move to cloud-based systems has also brought about big security problems, especially when it comes to keeping private data safe from attacks and access by people who shouldn't have it. Traditional steganographic methods, which are often used in multi-cloud environments, have tried to solve these problems by hiding private information in material that looks harmless and is shared across multiple cloud platforms [4]. These methods add an extra layer of security, but they also make things more difficult by making it hard to handle keys, using a lot of extra computing, and leaving you open to complex attacks.

In theory, multi-cloud strategies are better at spreading risk, but because they need to handle multiple sets of credentials and encryption keys, they are often less efficient and pose more security risks [2,3]. Dependence on different cloud service providers can also cause security policies to be inconsistent, which makes setting up a unified security strategy even harder. To keep private information safe in the cloud, we need steganography methods that are easier to use, safer, and more effective as cloud computing increases [6,7].

In this study, we proposed a new way to hide information that works only in a single cloud, which would solve some of the problems with current multi-cloud methods. Advanced cryptography methods, such as homomorphic encryption, elliptic curve cryptography (ECC), and lattice-based cryptography [8], are used in the proposed method to make sure strong security against both classical

and quantum threats. The proposed system is very resistant to attacks like brute-force because it uses these cutting-edge cryptographic methods.

The proposed mechanism also makes it easier to embed and retrieve data by cutting down on the need for complicated key management and the amount of computing power needed to hide data securely. This is done by limiting operations to a single cloud environment, making it easier to handle credentials, and making the system run more efficiently. Implementing and testing this method shows that it works to protect data's security, privacy, and availability, setting a new standard for safe data hiding in cloud computing.

In the following sections, this paper will explore the existing challenges in traditional steganographic methods, outline the design and implementation of the proposed mechanism, and present a comprehensive analysis of its performance and security features. The results indicate that this novel approach mitigates the risks associated with multi-cloud environments and significantly improves the efficiency and security of steganographic operations in cloud storage.

2. Existing Work

In cloud security, steganography has been extensively researched as a method for embedding sensitive information within cloud storage systems. Traditional approaches often utilize multi-cloud environments, where data fragments are distributed across various cloud providers to mitigate the risk of data exposure [2,6,7]. These methods integrate sophisticated cryptographic techniques to secure the embedded data, ensuring it remains invisible to unauthorized entities.

A. Essential Techniques in Existing Work:

1. Multi-Cloud Data Dispersion: Sensitive data is partitioned and distributed across multiple cloud platforms. This dispersion minimizes the likelihood of total data compromise but introduces complexities in data management and retrieval.

2. Asymmetric and Symmetric Encryption: Techniques such as RSA (Rivest-Shamir-Adleman), AES-256 (Advanced Encryption Standard), and Elliptic Curve Cryptography (ECC) are employed to encrypt data before embedding it within cloud storage. RSA, used for key exchange, relies on the mathematical difficulty of factoring large integers, while AES, a block cipher, is utilized for its efficiency in symmetric encryption.

Table 1: Comparison Of Encryption Techniques In Existing Multi-Cloud Systems

Technique	Advantages	Disadvantages
RSA	Strong security, well-established	Computationally expensive, large keys
AES-256	Efficient, strong, symmetric encryption	Requires secure key management
Elliptic Curve Cryptography (ECC)	Smaller keys, strong security	More complex implementation
Multi-Cloud Dispersion	Reduces risk of total data exposure	Complex management and retrieval

3. Steganographic Embedding: Steganographic techniques ensure that data embedding alters the carrier's statistical properties as little as possible, making detection via steganalysis infeasible. Techniques like Least Significant Bit (LSB) embedding are commonly used.

4. Complex Key Management Systems (KMS): Multi-cloud systems often require intricate KMS to manage encryption keys, with mechanisms for secure key storage, retrieval, and rotation across disparate cloud environments.

B. Problems in Existing Work

Despite the sophistication of existing approaches, several critical challenges persist:

1. **High Complexity in Credential and Key Management**: Managing multiple sets of credentials and keys across various cloud providers introduces a high potential for security lapses and operational inefficiencies.

2. **Large and Complex Stego-Key Management**: Data fragmentation across multiple platforms necessitates managing extensive stego-keys, increasing the risk of key compromise and operational overhead.

3. **Significant Computational Overhead**: The distribution, encryption, and secure data storage across multiple platforms require substantial computational resources, leading to inefficiencies.

4. **Detection Vulnerabilities**: While designed to resist steganalysis, the complexity of multi-cloud systems may inadvertently introduce detectable patterns, compromising security.

5. **Interoperability Issues**: Dependence on multiple cloud providers can lead to inconsistent security policies and potential vulnerabilities due to varying levels of provider security.

C. Aim

This research proposes a highly secure and computationally efficient steganographic method within a single cloud environment [5]. The proposed method seeks to eliminate the complexities and inefficiencies inherent in multi-cloud systems while providing robust security guarantees for the embedded data.

D. Objectives

1. **Simplification of Operations**: Streamline the data embedding and retrieval process by confining it to a single cloud environment, thereby reducing operational complexity and enhancing performance.

2. **Enhanced Security Framework**: Implement advanced cryptographic techniques that provide robust security against cryptographic attacks, including quantum-resistant algorithms.

3. **Efficiency and Performance**: Optimize the method to minimize computational overhead and maximize efficiency, even for large-scale data embedding operations.

4. **Advanced Key Management**: Develop a secure, scalable key management system that supports quantum-resistant key generation, distribution, and rotation within a single cloud environment.

E. Problem Definition

Traditional steganographic methods that rely on multi-cloud environments introduce significant complexity and inefficiency. The substantial drawbacks are the requirement to manage multiple credentials, handle large and complex stego-keys, and navigate inconsistent security policies across different cloud providers. Therefore, a simplified yet secure steganographic method operating within a single cloud environment is essential to ensure data integrity, confidentiality, and robust security against unauthorized access.

3. Proposed Mechanism

We propose a novel cryptographic steganography mechanism that operates entirely within a single cloud environment to address these challenges. This mechanism leverages advanced mathematical constructs, including elliptic curve cryptography (ECC), quantum-resistant lattice-based cryptography, homomorphic encryption, modular arithmetic, non-linear cryptographic permutations, and substitutions, as well as hash-based file mapping, to embed sensitive information into cloud storage files securely. The proposed solution aims to simplify operations, reduce computational overhead, and provide high security against classical and quantum attacks.

A. Considerations, Assumptions, Pre-Sharing, Agreements

Considerations

1. **Data Size and Computational Complexity**: The method assumes that the cloud environment can efficiently handle the data size and computational requirements. The system must be scalable and capable of embedding and retrieving large volumes of data without significant performance degradation.

2. **Cloud Infrastructure Support**: The cloud environment must support complex file operations, including elliptic curve cryptographic operations, homomorphic encryption, modular arithmetic, file creation, deletion, and fine-grained access control.

3. **Adherence to Advanced Security Policies**: The cloud environment must adhere to stringent security policies, including quantum-resistant encryption standards, elliptic curve key management, and comprehensive audit trails to track all data operations.

Assumptions

1. **Trust in the Cloud Service Provider (CSP)**: The mechanism assumes that the CSP is entirely trustworthy, with infrastructure secure against insider threats, quantum attacks, and external breaches.

2. **No Data Loss or Corruption**: It is assumed that the data stored in the cloud is secure against loss or corruption, ensuring that embedded data remains intact until retrieval.

3. **Integrity of File Storage**: The method assumes that files in the cloud storage retain their integrity, meaning they are not tampered with or altered by unauthorized entities.

Pre-Sharing and Agreements

1. **Pre-Sharing of Cryptographic Keys**: Cryptographic keys for permutation, substitution, and hashing must be securely pre-shared among authorized users using post-quantum key exchange protocols like NewHope or a quantum-resistant Key Management Service (KMS).

2. **User Agreements and Compliance**: All data embedding and retrieval users must adhere to strict protocols regarding key usage, access controls, and data handling procedures. Agreements should include terms for regular quantum-resistant audits and compliance with advanced security policies.

B. Proposed Mechanism Explanation

The proposed mechanism consists of several advanced stages, each incorporating sophisticated cryptographic operations to ensure the security and integrity of the embedded data.

Secret Encoding Using Modular Arithmetic and Homomorphic Encryption

The secret message $'s'$ is encoded into a base $'B$, representation using modular arithmetic. The encoded message is then homomorphically encrypted to allow computations on the ciphertext without decrypting it, preserving data privacy:

$$S_B = \sum_{i=0}^{n-1} S_i . B^i \, mod P$$

$$\mathrm{E}(S_B) = \mathrm{E}(\sum_{i=0}^{n-1} S_i . B^i \, mod P)$$

Where: S_i represents the individual elements of the secret message. B^i is the chosen base for determining the encoding complexity. P is a large prime number that ensures the non-cyclic properties

of the group and provides security against algebraic attacks. $\mathrm{E}(S_B)$ represents the homomorphically encrypted encoded message, enabling secure computation on the ciphertext.

Elliptic Curve Cryptographic Permutation

The homomorphically encrypted message undergoes a permutation defined by a secure elliptic curve-based permutation key Kp. The permutation function $'\pi'$ is derived from a pseudo-random permutation function (PRP) implemented over an elliptic curve:

$$\pi(\mathrm{E}(S_B)) = \mathrm{E}(S_{\pi(0)}, S_{\pi(1)} \dots\dots\dots\dots S_{\pi(n-1)})$$

Where: $'\pi'$ represents the permutation sequence that reorders the elements of S_B based on the elliptic curve's properties, enhancing non-linearity and complexity and ensuring resistance to quantum and classical cryptanalysis.

Substitution Using Lattice-Based Cryptographic S-box

The permuted, encrypted data is further transformed using a cryptographic substitution box (S-box) S based on lattice-based cryptography, which provides security against quantum attacks. This non-linear transformation increases entropy and resistance to lattice reduction attacks:

$$S(\pi(E(S_B))) = \{S(S_{\pi(0)}), \mathrm{S}(S_{\pi(1)}) \dots\dots\dots\dots \mathrm{S}(S_{\pi(n-1)})\}$$

Where $'S'$ is a highly non-linear function derived from lattice-based cryptographic constructs, ensuring that the relationship between input and output is complex and resistant to both classical and quantum cryptanalysis

Hash-Based File Mapping with Post-Quantum Hash Functions

The transformed data is then embedded into the cloud storage by mapping each substituted element to a specific file in the cloud using a secure post-quantum hash function H:

$$F_{\tilde{S}}[\mathrm{i}] = \mathrm{H}(\mathrm{S}(\pi(\mathrm{E}(S_B)[\mathrm{i}]))\mathrm{mod\ N})$$

Where: $F_{\tilde{S}}[\mathrm{i}] =$ is the file in the stego-folder corresponding to the i-th element of the transformed sequence. $'H'$ is a post-quantum hash function such as SHAKE256 (a variant of SHA-3), ensuring that the mapping is uniform, collision-resistant, and effectively obfuscates the location of the embedded data.

Quantum-Resistant Key Management and Distribution

Cryptographic keys are generated using quantum-resistant algorithms and managed by a quantum-resistant Key Management System (KMS). The KMS handles key storage, secure distribution, and regular rotation:

1. **Elliptic Curve Permutation Key Kp**: Generated using quantum-resistant cryptographic algorithms like Curve25519, securely stored in the KMS, and distributed using quantum-resistant key exchange protocols.

2. **Lattice-Based Substitution S-box S**: Generated from lattice-based cryptographic constructs and securely managed within the KMS.

3. **Post-Quantum Hash Function H**: Selected to ensure security against quantum attacks, with SHAKE256 providing the necessary cryptographic strength.

Keys are rotated periodically to prevent key exhaustion and mitigate the risk of quantum-based attacks.

Table 2: Key Management System (KMS) Overview

Key Type	Generation Method	Distribution Protocol	Rotation Frequency
Elliptic Curve Permutation Key	Quantum-resistant Curve25519	Quantum-resistant key exchange	Every 30 days
Lattice-Based Substitution S-box	Lattice-based cryptographic constructs	Secure KMS	Every 60 days
Post-Quantum Hash Function Key	SHAKE256	Secure KMS	As required

4. Proposed Mechanism Algorithm

A. EMBEDDING ALGORITHM

Input: Secret message s, Prime number P, Base B Cloud storage credentials C Elliptic curve permutation key Kp, Lattice-based substitution S-box S, Post-quantum hash function H, Number of files N in cloud folders

Output: Stego-folder $F_{\tilde{S}}$ containing the embedded secret

Steps:

1. **Modular Encoding with Homomorphic Encryption:** Encode and encrypt the secret message:

$$S_B = \sum_{i=0}^{n-1} S_i . B^i \, mod P$$

$$\mathrm{E}(S_B) = \mathrm{E}(\sum_{i=0}^{n-1} S_i . B^i \, mod P)$$

2. **Elliptic Curve Permutation:** Permute the encrypted sequence:

$$\pi(\mathrm{E}(S_B)) = \mathrm{E}(S_{\pi(0)}, S_{\pi(1)} \dots \dots \dots \dots . S_{\pi(n-1)})$$

3. **Lattice-Based Substitution:** Apply the S-box substitution:

$$S(\pi(E(S_B))) = \{S(S_{\pi(0)}), \mathrm{S}(S_{\pi(1)}) \dots \dots \dots \dots . \mathrm{S}(S_{\pi(n-1)})\}$$

4. **Post-Quantum Hash-Based Mapping to Cloud Files:** Map the substituted data to specific files in the cloud:

$$F_{\tilde{S}}[\mathrm{i}] = \mathrm{H}(\mathrm{S}(\pi(\mathrm{E}(S_B)[\mathrm{i}]))\mathrm{modN})$$

5. **Store in Cloud:** Store the mapped files in the stego-folde $F_{\tilde{S}}$ in the cloud.

B. EXTRACTION ALGORITHM

Input: Stego-folder $F_{\tilde{S}}$, Prime number P, Base , Cloud storage credentials C, Elliptic curve permutation key Kp, Lattice-based substitution S-box S, Post-quantum hash function H

Output: Retrieved secret message s

Steps:

1. **Retrieve Stego-Folder from Cloud:** Access and retrieve the files from $F_{\tilde{S}}$
2. **Inverse Hash Mapping:** Reconstruct the substituted sequence:

$$S(\pi^{-1}(F_{\tilde{S}}[\mathrm{i}])) = H^{-1}(F_{\tilde{S}}[\mathrm{i}])$$

3. **Inverse Substitution:** Apply the inverse S-box:

$$\pi^{-1}(\mathrm{S}^{-1}(\mathrm{F}S_{\tilde{S}}[\mathrm{i}])) = s_B[\mathrm{i}]$$

4. **Inverse Permutation:** Reverse the permutation:

$$s_B = \pi^{-1}(\mathrm{S}^{-1}(\mathrm{F}S_{\tilde{S}}))$$

5. **Homomorphic Decoding:** Decrypt and decode the modular base sequence to retrieve the original secret:

$$s = \sum_{i=0}^{n-1} s_B[\mathrm{i}].B^{-i}\, mod P^{-1}$$

Table 3: Summary of Embedding and Extraction Processes

Process Step	**Operation**	**Output**
Encoding & Encryption	Modular Arithmetic + Homomorphic Encryption	Encrypted encoded message
Permutation	Elliptic Curve PRP	Permuted encrypted sequence
Substitution	Lattice-Based S-box	Substituted encrypted sequence
Hash Mapping	Post-Quantum Hash Function	Mapped files in cloud storage
Decoding & Decryption (Reverse)	Inverse operations of above	Retrieved secret message

5. Performance Analysis

We implemented the system in a controlled environment to validate the proposed mechanism using various data types, including text, images, and binary files. The validation process focused on testing security against quantum and classical attacks, efficiency, and robustness.

Validation Metrics

- **Data Integrity**: Ensured that the retrieved data perfectly matched the original data, with no losses or corruption.[9]
- **Security Testing**: Conducted penetration testing, including quantum-based attacks, to evaluate the robustness of the mechanism.
- **Performance Metrics**: Measured the time complexity and resource utilization during the embedding and retrieval processes in a quantum-safe environment.

Performance Analysis Environment

The experiment conducted within a **single cloud storage environment** to evaluate the performance of proposed steganographic schemes

Performance Metrics

- **Execution Time**: The time taken to complete the embedding and retrieval processes was measured to evaluate the efficiency of the system.
- **Resource Utilization**: Monitored CPU, memory usage, and network bandwidth during the process to assess the impact on system performance.
- **Scalability**: Tested the system's ability to handle increasing data sizes and complexity, ensuring it can scale effectively.

Table 4: Performance Metrics

Metric	**Measurement**	**Result**
Execution Time	Time to embed/retrieve data	Optimal for large datasets
Resource Utilization	CPU, memory, and bandwidth usage	Efficient usage
Scalability	Handling increasing data size/complexity	Scaled effectively

Security Features

1. **Brute-Force Resistance**: Using large prime numbers PPP, complex permutation keys Kp, and non-linear S-boxes S ensures that brute-force attacks are computationally infeasible. Given a key space of size 2^n, where n is the key length in bits, a brute-force attack requires an average 2^{n-1}operations, making it infeasible even with quantum computing power.

2. **Resistance to Differential Cryptanalysis**: The non-linear transformations applied during the permutation and substitution stages significantly increase the entropy of the data, making it resistant to differential cryptanalysis. The probability PD of a differential cryptanalysis attack succeeding can be minimized using non-linear operations where:

$$P_D = \prod_{i=1}^{r} max\Delta_i$$

where $\Delta_i$represents the differential probabilities at each round r of the encryption process. The use of non-linear S-boxes reduces $max\Delta_i$, making P_D close to zero.

3. **Steganalysis Resistance**: The hash-based file mapping ensures that the statistical distribution of the embedded data closely resembles that of normal data, making it indistinguishable from steganalysis tools. Let H(x) be the post-quantum hash function (e.g., SHAKE256), where the probability of collision P_C is defined by:

$$P_C = \frac{1}{2^{2m}}$$

Where m is the output length in bits of the hash function, this low collision probability ensures that the mapping is unique and resistant to attacks.

Attack Prevention Mechanisms

- **Cryptographic Hashing**

The system uses a post-quantum hash function H to ensure that data mapping within the cloud storage is secure and collision-resistant. Given an input x and a secure hash function H, the output is:

$$H(x) = y$$

The hash function is resistant to pre-image attacks (finding x given y), second pre-image attacks (finding another input x′ such that $H(x') = H(x)$, and collision attacks (finding two different inputs

x_1 and x_2 such that H(x_1)=H(x_2). The collision resistance is significant in steganography, as it ensures that each piece of data maps uniquely to a file in the cloud, preventing unauthorized access or manipulation.

- **Quantum-Resistant Key Management**

The quantum-resistant key management system ensures that cryptographic keys are securely stored, distributed, and rotated. The key exchange is based on quantum-resistant algorithms like NewHope or lattice-based cryptography, ensuring that even quantum computers cannot break the key exchange process. The security of key management is derived from the difficulty of solving lattice-based problems, such as the Learning With Errors (LWE) problem, where the goal is to find a vector s given a set of equations:

$$A \cdot s + e = b$$

Here, A is a known matrix, b is a known vector, e is a small error vector, and s is the secret vector to be found. The hardness of LWE is based on the fact that finding s is computationally infeasible, even with quantum algorithms. The rotation of key K ensures that even if an adversary were to obtain a key, it would only be valid for a short period, limiting the impact of any potential breach:

$$K_{i+1} = \text{PRNG}(K_i)$$

where PRNG is a pseudo-random number generator ensuring that each new key is unpredictable and uncorrelated with previous keys.

- **Multi-Layered Encryption**

The system employs multi-layered encryption, combining homomorphic encryption, elliptic curve cryptography (ECC), and lattice-based cryptography. The system's security relies on the difficulty of solving problems in these domains, such as the Elliptic Curve Discrete Logarithm Problem (ECDLP) and lattice-based problems like the Shortest Vector Problem (SVP).

- **Homomorphic Encryption**: Enables computations on encrypted data without decryption. Given ciphertexts $c_1 = E(m1)$ and $c\_2 = E(m2)$, the encryption supports operations such that:

$$E(m1) \cdot E(m2) = E(m1 + m2)$$

- **Elliptic Curve Cryptography (ECC)**: Based on the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP), where given a point P on an elliptic curve and a scalar k, it is computationally infeasible to find k given P and $Q = kP$. The key size in ECC is significantly smaller compared to RSA for the same level of security:

- **Lattice-Based Cryptography**: Based on complex problems like the Shortest Vector Problem (SVP), where finding the shortest non-zero vector in a lattice Λ is computationally infeasible. The security of lattice-based cryptography is believed to be strong even against quantum computers.

$$SVP\colon\ Find\ v \in \Lambda\ such\ that\ \|\ v\ \|\ is\ minimal$$

The combined use of these cryptographic techniques ensures that even if one layer were to be compromised, the other layers would still provide robust security.

6. Results Discussion

Key Outcomes

1. **Efficiency**: The proposed mechanism significantly reduced computational overhead compared to traditional multi-cloud steganography methods. Embedding and retrieval processes were performed within optimal time frames, even for large datasets.

2. **Security**: The system demonstrated strong resistance to all forms of cryptanalysis and steganalysis tested during validation, with no successful breaches recorded.

3. **Simplicity**: By confining operations to a single cloud environment, the mechanism simplified credential management, key handling, and overall operational complexity.

Table 5: Results Summary

Metric	Measurement	Outcome
Security	Resistance to quantum and classical attacks	Strong resistance, no breaches
Efficiency	Execution time and resource usage	Optimal for large datasets
Scalability	Handling increasing data size/complexity	Effective, further optimization is possible

7. Conclusion

This research successfully developed and validated an advanced cryptographic steganography mechanism within a single cloud environment. The proposed method provides a secure, efficient, and easy-to-implement solution for embedding sensitive data within cloud storage by eliminating the complexities associated with multi-cloud systems and focusing on robust cryptographic techniques. The system strongly resisted attacks, ensuring the data's confidentiality, integrity, and availability.

Reference

[1] Abdul, Arif Mohammad, M. Balraju, and Sudarson Jena. "Trusted System In Cloud Environment." International Journal of Engineering Research & Technology (2013): 865-869.

[2] Hashmi, Syed Shakeel, et al. "Enhancing Data Security in Multi-Cloud Environments: A Product Cipher-Based Distributed Steganography Approach." International Journal of Safety & Security Engineering 14.1 (2024).

[3] Arif, Mohammad Abdul, et al. "Brute Force Attack on Distributed data Hiding in the Multi-Cloud Storage Environment More Diminutive than the Exponential Computations." Ingenierie des Systemes d'Information 27.6 (2022): 915.

[4] Leonel, Moyou Metcheka, and Ndoundam René. "Distributed data hiding in multi-cloud storage environment." Journal of Cloud Computing 9.1 (2020).

[5] Mossebo Tcheunteu, Stéphane Willy, Leonel Moyou Metcheka, and René Ndoundam. "Distributed data hiding in a single cloud storage environment." Journal of Cloud Computing 10.1 (2021): 43.

[6] Banu, Mrs Salma, Vijayalaxmi Biradar, and Arshad Ahmad Khan Mohammad. "Secure And Undetectable Multi-Cloud Steganography: Leveraging Non-Alteration Techniques For Covert Communication." Educational Administration: Theory and Practice 30.6 (2024): 1789-1798.

[7] Banu, Mrs Salma, Vijayalaxmi Biradar, And Arshad Ahmad Khan Mohammad. Enhanced Data Security In Multi-Cloud Environments: An Advanced Product Cipher-Based Distributed Steganography Approach. (2024). Cahiers Magellanes-Ns, 6(2), 835-840.

[8] Morkel, T., and J. H. P. Eloff. "Encryption techniques: a timeline approach." *Information and Computer Security Architecture (ICSA) Research Group, University of Pretoria* 2 (2004).

[9] Dr. Mohammed Abdul Bari,Arul Raj Natraj Rajgopal, Dr.P. Swetha ," *Analysing AWSDevOps CI/CD Serverless Pipeline Lambda Function's Throughput in Relation to Other Solution*", International Journal of Intelligent Systems and Applications in Engineering , JISAE, , , 12(4s), 519–526, Nov 2023, ISSN:2147-6799