Designing Advanced Cryptographic Solutions for Cloud Storage Security Through Dual-Layer Encryption Protocols

Paul Kobina Arhin Jnr[1], James Ben Hayfron-Acquah[2], Frimpong Twum[2] and Alfred Kumah[3]
[1]*Department of Computer Science and I.T, University of Cape Coast, CC-123-1749, Ghana*
[2]*Department of Computer Science, KNUST, AK-385-1973, Ghana*
[3]*Department of IT Infrastructure, University of Health and Allied Sciences, PMB 31, Ghana*
parhin@ucc.edu.gh , jbhayfron-acquah@knust.edu.gh, ftwum.cos@knust.edu.gh , akumah@uhas.edu.gh

Keywords: Cloud Storage, Data Integrity, Hybrid Cryptosystem, ChaCha20-Poly1305, ElGamal Encryption.

Abstract: This paper presents a novel hybrid cryptosystem combining ElGamal and ChaCha20-Poly1305 for securing data in cloud storage environments. Elgamal, an asymmetric encryption algorithm is utilized for secure key exchange to ensure the safe transmission of keys. ChaCha20-Poly1305, a high-performance symmetric encryption algorithm is used for efficient data encryption and authentication, thereby addressing the performance limitations of using traditional asymmetric algorithms alone. Using ChaCha20-Poly1305 for bulk data encryption and transmission significantly improves speed and reduces computational overhead compared to using only asymmetric algorithm. This makes it ideal for modern cloud storage applications. This novel system is made to provide robust defence against common cloud-based attacks such as eavesdropping, replay attacks, brute force and man-in-the-middle attacks. By making use of the strengths of both asymmetric and symmetric encryption algorithms, the system will ensure high security with better performance. The proposed hybrid Algorithm was tested and compared with the traditional Elgamal alone, and it achieved 83% faster encryption, 79.5% faster decryption and 69.5% lower memory usage. This shows its efficiency for secure cloud data storage and transmission.

1 INTRODUCTION

Cloud storage has in our modern world, become a vital key component of information technology infrastructure [1]. Cloud storage provides a large range of computational resources and storage capabilities [2]. This technology has completely transformed data-centric centres, ranging from healthcare and banking to e-commerce and artificial intelligence by offering flexible and scalable services [3][4]. However, the proliferation of cloud-based solutions has also increased concerns about data security and privacy [5]. As companies and individuals continually entrust sensitive information to cloud platform services, there is an urgent need for robust encryption techniques to protect against unauthorized access, data breaches, and other cyber threats [6]. Most classical cloud computing security models use symmetric algorithms for fast data encryption and asymmetric key algorithms for safe key management and secure key exchange [7]. Although most of these traditional models use well known algorithms like AES for bulk data encryption and have withstood many cyber-attacks, they are still not without challenges [8]. These challenges include performance overhead, implementation complexity, and the constantly changing cryptographic environment, which demands continuous scrutiny to maintain confidentiality and integrity in the face of emerging cyber-attacks [9].

ElGamal cryptosystem is a highly secured asymmetric algorithm whose strength lies in the difficulty of the discrete logarithm problem over large finite fields [10]. This makes it very difficult for attackers to solve or breakthrough. Its resilience against known cryptanalytic attacks makes it a reliable choice for key exchange and encryption [11]. This makes the Elgamal cryptosystem secure for encrypting data, exchanging keys, and creating digital signatures [12]. Until today, Elgamal still remains a trusted algorithm in public-key cryptography [13]. Even though Elgamal offers strong security through its complex mathematical foundations, this same complexity can highly affect its performance by making it relatively slow when used alone, particularly in high-volume, cloud-based data [14].

On the other hand, ChaCha20-Poly1305 has emerged as a state-of-the-art Authenticated Encryption with Associated Data (AEAD) algorithm known for its speed, resistance to timing attacks, and robust authentication guarantees [15]. Originally, it was developed as a more efficient alternative to AES in software implementations but has received increasing attention for its consistent performance across a range of platforms and architectures [16].

To address the challenge of the Elgamal cryptosystem performance limitations, this paper proposes a novel hybrid cryptographic scheme that combines the strengths and key exchange capabilities of ElGamal with the efficiency and authenticity of ChaCha20-Poly1305. By creating a hybrid of asymmetric key exchange technique with a high-performance AEAD cipher, the work seeks to introduce an exceptional end-to-end secure system for cloud storage and data transmission that addresses both confidentiality and integrity requirements.

By leveraging ElGamal for secure session key establishment and ChaCha20-Poly1305 for fast bulk encryption and authentication, this hybrid approach ensures strong and robust security for cloud storage and transmission while maintaining high speed. Even if some parts of the cloud infrastructure are compromised, it would be very hard for an attacker to figure out the encryption keys or alter the data without being detected. This makes the system highly secure and efficient for protecting sensitive information stored and transmitted in the cloud.

2 REVIEW OF RELATED LITERATURE

Cloud computing has indeed changed and transformed how data is stored and processed, but it also poses serious security risks. In order to defend and protect cloud storage systems against these risks like data breaches and cyberattacks, various cryptographic algorithms and security mechanisms have been proposed. This literature review examines different cryptographic techniques and security implementations in cloud computing storage.

Peng et al. [17] conducted a thorough review of secure cloud storage based on cryptographic algorithms. The findings from their work proved the importance of encryption algorithms such as AES (Advanced Encryption Standard), RSA (Rivest-Shamir-Adleman), and DES (Data Encryption Standard) in securing cloud storage. According to their research, classical cryptography may result in performance overhead even though it offers robust security.

Salman and Sulaiman [18] also conducted a thorough analysis into multiple cryptographic algorithms, including Blowfish, RC6, and Feistel, and compared their efficiency in cloud environments. The findings from the research showed that, lightweight encryption algorithms such as Blowfish are more suitable for cloud applications due to their lower computational costs.

Ahmad and Garko [19] also conducted a thorough review into hybrid cryptographic algorithms in cloud computing. They analyzed the effectiveness of these algorithms in securing large-scale cloud environments. Their study identified gaps in user authentication and key management in hybrid encryption techniques.

Sharma et al. [20] also conducted an investigation into hybrid cryptographic algorithms in cloud storage, analyzing their encryption and decryption speeds. Their findings also suggested that, even though hybrid cryptographic techniques provide enhanced security, they often introduce additional computational overhead.

A hybrid cryptographic scheme that integrates Elliptic Curve Cryptography (ECC) and Triple Data Encryption Standard (TDES) was proposed by Kaur and Jain [21]. Their experimental results showed that ECC-TDES enhances data security while maintaining accuracy, but it slightly increases encryption time.

Mendez [22] also introduced DNA cryptography as a novel method for enhancing cloud security. The study proposed a Bi-Directional DNA Encryption Algorithm (BDEA), which combines bio-molecular principles to enhance encryption strength. The research discovered that, the novel DNA cryptography offers a promising direction for unbreakable encryption, but practical implementations require further exploration.

Kavya and Acharva [23] conducted a comparative analysis on different homomorphic encryption schemes. They concluded that, while homomorphic encryption is highly secure, it remains computationally expensive.

Madhavi and Sivareddy [24] as well examined using public key encryption algorithms for secure cloud storage. Their work introduced the Dual-Server Public Key Encryption with Keyword Search (DS-PEKS) to mitigate keyword guessing attacks. The findings suggested that DS-PEKS enhances privacy in cloud-based search operations but requires further optimization.

Gadad and Anbusezhiyan [25] also came up with a novel approach for securing cloud storage. This was

done by encoding plaintext into an intermediate compressed format before encryption. Their study proved that, integrating compression with encryption enhances both security and transmission efficiency.

Sharma et al. [26] also conducted a comparative analysis into symmetric encryption algorithms for securing cloud storage. This was done by comparing AES, Blowfish, and DES based on encryption time and throughput. Their findings suggested that optimizing symmetric algorithms can improve both security and transmission speed.

Sasikumar and Nagarajan [27] also conducted a comprehensive meta-analysis on various cryptographic security algorithms for cloud computing. Their study highlighted the integration of machine learning with cryptographic algorithms to enhance security. The research proposed the use of AI-driven anomaly detection to identify security threats in real time.

2.1 Cryptographic Algorithms Used in Leading Cloud Storage Systems

This section discusses some leading cloud storage systems and the cryptographic algorithms they use to enhance security.

Amazon Web Services (AWS) Cloud Storage: This is one of the most commonly used cloud storage system**.** Amazon S3 and AWS Key Management Service (KMS) use Advanced Encryption Standard (AES-256) encryption to secure data at rest. For data in transit, AWS employs Transport Layer Security (TLS) to ensure confidentiality [28].

Google Cloud Storage: Another well-known cloud system is the Google cloud storage. Google Cloud Storage provides multiple encryption options, including AES-256 for server-side encryption and RSA for key management. Google also supports customer-managed encryption keys (CMEK) and customer-supplied encryption keys (CSEK) to enhance data protection [29].

Microsoft Azure Storage: Microsoft Azure Storage using AES-256 encryption for data in storage and uses TLS for data in transit. Azure also implements a hybrid cryptographic approach combining RSA and AES for optimal security [30].

IBM Cloud Object Storage: IBM Cloud also makes use of AES-256 encryption for data in storage and TLS for secure data transfer. It also supports envelope encryption, where an AES data key is encrypted using an RSA key for additional security [31].

Most cloud storage systems use various cryptographic algorithms, with TLS, AES-256 and RSA being widely implemented for strong security. These hybrid encryption approaches are used to enhance data protection, ensuring confidentiality and integrity in cloud environments.

3 METHODOLOGY

At this stage, we implement the hybrid cryptosystem algorithm combining ElGamal and ChaCha20-Poly1305 for cloud storage. we will focus on the four (4) stages of the algorithm, namely key generation, encryption, transmission of data, and decryption. The goal is to combine the asymmetric ElGamal encryption (for securely exchanging a shared key) with the symmetric ChaCha20-Poly1305 encryption (for efficiently encrypting the actual data).

3.1 Key Generation Stage

The first stage in the hybrid cryptosystem involves key generation for both the ElGamal and ChaCha20-Poly1305 algorithms:

1) Elgamal Key Generation (Asymmetric):
   - First, we select a large prime number, p and a generator, g such that g is a primitive root modulo p. These values must be publicly known.
   - Then, Choose a private key x (a random number between 1 and p - 2)
   - Calculate the corresponding public key

     $$y = g^x \bmod p \qquad (1)$$

     Public key: (p, g, y)
     Private key: (x)
2) ChaCha20-Poly1305 Key Generation (Symmetric encryption):
   - First, Choose a random 256-bit key, ksym for ChaCha20
   - Next, Select a random nonce (12-byte) n_{nonce} for the ChaCha20 encryption.
3) The key, ksym and the nonce, nnonce are kept secret between the sender and receiver, but they need to be shared securely.
   Symmetric key: k_{sym}
   Nonce: n_{nonce}

OUTPUT: Public key: **(p, g, y),** Private key: (x **)**
ChaCha20 **:** k_{sym}**,** n_{nonce}

3.2 Encryption Stage

Once the keys are generated, the encryption process follows. Here, the sender will have to securely transmit the symmetric key, k_{sym} to the receiver. This is done by using the ElGamal algorithm, where a random integer, k is chosen by the sender. The sender will then go ahead and compute two values: $C_1 = g^k \bmod p$ and $C_2 = (y^k \cdot m) \bmod p$. where m is the symmetric key k_{sym} , which has been converted to an integer. The result will become an ElGamal ciphertext (C_1, C_2), which contains the encrypted symmetric key.

The symmetric key is then securely transmitted. The actual ***D*** (This refers to the data to be transmitted to the cloud and could be files or information to be uploaded to the cloud storage) is encrypted using the ChaCha20-Poly1305 algorithm. The encryption of the data will then yield two outputs, which are the ciphertext (C_{data}), which is encrypted form of the data as well as the authentication tag (t***ag***), which is also used to verify the integrity and authenticity of the data during decryption. Here, the ChaCha20-Poly1305 algorithm offers both confidentiality and integrity assurance, ensuring that the encrypted data cannot be tampered with during transmission.

ElGamal encrypts k_{sym} to produce

$$C_{elgamal} = (C_1, C_2), \qquad (2)$$

ChaCha20-Poly1305 encrypts D to produce C_{data} , $_{tag}$.

3.3 Transmission of Data

Once the encryption process is completed, the next stage involves the transmission of the encrypted data over a channel which may not be secured to a cloud storage platform. The data transmitted here includes: $C_{elgamal} = (C_1, C_2)$, the ChaCha20-Poly1305 ciphertext C_{data}, and the authentication tag, *tag*. The nonce n_{nonce} which is used for ChaCha20 encryption is also sent to ensure that, the same nonce is used for decryption by the receiver. This process allows for secure storage and retrieval of the data from the cloud, ensuring that both the key and data are encrypted.
($C_{elgamal}$, C_{data}, *tag*, n_{nonce}) is sent over the cloud storage or secure channel.

3.4 Decryption

The final stage in the hybrid algorithm is the decryption of the data by the receiver. Here, in order for the receiver to recover the symmetric key, k_{sym}, the ElGamal ciphertext (C_1, C_2) will have to be decrypted. The receiver will use their private key, x to compute the shared secret $s = C_1^x \bmod p$ and its inverse modulo p, s^{-1}. The symmetric key k_{sym}, is then recovered by computing $m = (C_2 \cdot s^{-1})$, where m is the encrypted symmetric key.
Once the symmetric key k_{sym} is successfully recovered, the receiver can decrypt the actual data which is encrypted with ChaCha20-Poly1305. Using the symmetric key k_{sym} and the nonce n_{nonce} , the receiver then applies the ChaCha20-Poly1305 decryption algorithm to retrieve the original plaintext D. Using the authentication tag, *tag*, if the integrity check passes, the decrypted data is successfully recovered and can now be used by the receiver.

Elgamal Decryption:

$$\text{Compute } s = C_1^x \bmod p \qquad (3)$$

$$\text{Compute } s^{-1} \bmod p$$

$$\text{Recover } k_{sym} = (C_2 \cdot s^{-1}) \bmod p \qquad (4)$$

ChaCha20-Poly1305 Decryption:

$$\text{Decrypt data } D = \text{ChaCha20-Poly1305\_Decrypt}(k_{sym}, n_{nonce}, C_{data, tag})$$

Verification:

Tag from the decrypted ciphertext is computed:

$$tag' = \text{ChaCha20-Poly1305\_GenerateTag}(k_{sym}, n_{nonce}, D) \qquad (5)$$

The recomputed *tag'* is compared with the received authentication *tag:*

```
if (tag' == tag) {
Data integrity verified
}
Else {
Data integrity verification failed
}
```

If the tags match, data integrity is verified. If they don't match, the data is considered compromised.

This novel hybrid cryptosystem combines ElGamal and ChaCha20-Poly1305. It offers a great solution for cloud storage security. For cloud storage services, the hybrid encryption system, it ensures that encryption is done efficiently on the client system before uploading. It will ensure rapid encryption of large volumes of information that will enable quick and secure transmissions even in the midst of systems

where bandwidth and device capabilities are limited. The fast encryption algorithm, ChaCha20-Poly1305 minimizes device resource usage while Elgamal secures the session keys to defend the cloud against any breaches. Security is high compared to traditional RSA or Elgamal. This hybrid will also offer good performance on modern processors even without any hardware accelerators like AES-NI.

4 EXPERIMENTAL RESULTS

The Hybrid cryptosystem which combines Elgamal and ChaCha20-Poly1305 for cloud storage, was implemented on an i5 HP desktop running windows 10 with 3.2GHz of speed and RAM of 8GB. The results are shown in the tables and graphs. The data size used was 5MB. Table 1 shows the Encryption and decryption speed comparison between Elgamal and the hybrid of Elgamal and Elgachat across different key sizes, whiles Table 2 shows the memory usage comparison between Elgamal and Elgachat across different key sizes.

Table 1: Encryption and decryption speed comparison between Elgamal and the hybrid of Elgamal.

Key Size (in Bits)	Cryptographic Algorithms	Encryption Speed (ms)	Decryption Speed (ms)
1024	ElGamal	45.2	85.9
	ElgaChat	14.2	22.2
2048	ElGamal	93.1	132.8
	ElgaChat	17.9	34.4
3072	ElGamal	138.7	193.2
	ElgaChat	22.7	46.6
4096	ElGamal	188.5	261.4
	ElgaChat	34.9	53.6

Figure 1 shows a significant improvement in the proposed hybrid system as compared to Elgamal alone used in encryption. The hybrid system outperforms the Elgamal system in both Encryption and Decryption speeds across all key sizes, having encryption being up to 83.6% faster and with decryption being up to about 79.5% faster.

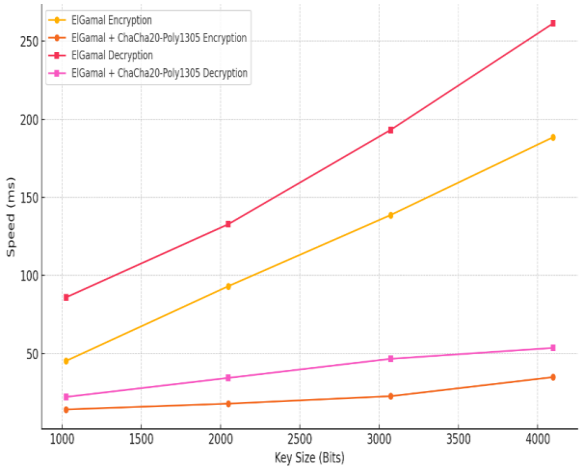

Figure 1: Encryption and decryption speed comparison between Elgamal and the hybrid of Elgamal and ChaCha20-Poly1305

Table 2: Memory Consumption comparison between Elgamal and the hybrid of Elgamal.

Key Size (in Bits)	Cryptographic Algorithms	Encryption (MB)	Decryption (MB)
1024	ElGamal	1.05	1.28
	ElgaChat	0.32	0.63
2048	ElGamal	1.55	1.92
	ElgaChat	0.53	0.95
3072	ElGamal	2.17	2.32
	ElgaChat	1.12	1.35
4096	ElGamal	3.29	4.01
	ElgaChat	1.53	2.01

Comparing the hybrid algorithm with the standalone Elgamal in terms of memory consumption, it is observed that, the hybrid of Elgamal and ChaCha20-Poly1305 uses less memory of up to 69.5% for encryption and about 51% for decryption compared to Elgamal alone. This makes it more memory efficient across all key sizes. This is shown in Figure 2.

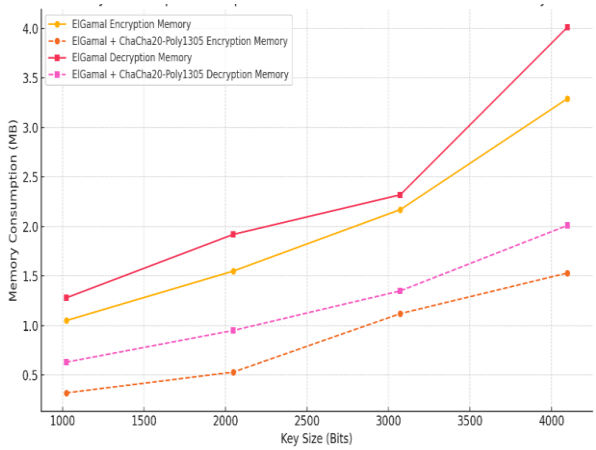

Figure 2: Memory Consumption comparison between Elgamal and the hybrid of Elgamal and ChaCha20-Poly1305.

5 CONCLUSIONS

The Hybrid cryptosystem which combines Elgamal and ChaCha20-Poly1305 for cloud storage, is designed with the aim of providing a highly secure and efficient solution for data encryption, transmission, and decryption. Elgamal is used for secure key exchange and ensures that the symmetric key required for ChaCha20-Poly1305 encryption remains securely transmitted. This addresses the challenge of securely distributing symmetric keys over insecure channels. ChaCha20-Poly1305 is a lightweight and fast encryption algorithm, which provides confidentiality, authenticity, and integrity for the data while maintaining high performance.

Furthermore, ElGamal cryptographic algorithm is relatively slow and inefficient for large datasets and this is where ChaCha20-Poly1305 excels. ChaCha20-Poly1305 is highly efficient and can handle large amounts of data with less computational overhead compared to traditional asymmetric encryption algorithms like RSA or ElGamal. The experiment conducted clearly shows that the hybrid of Elgamal and ChaCha20-Poly1305 outperforms Elgamal algorithm alone in both speed and memory efficiency across all key sizes. This makes it an efficient and scalable choice for cloud storage transactions. ChaCha20-Poly1305 also provides authenticated encryption with additional layer of security against tampering. Even though this hybrid system provides the best of security and efficiency, it may be limited in terms of implementation. Implementing a symmetric and an asymmetric cryptographic algorithm combined may be complex and may be computationally expensive and affect the overall system. The key management complexity increases as the system scales to accommodate more users. Future work may focus on optimizing key exchange process by adopting more efficient key exchange protocols.

Some of the key types of attacks that this system can mitigate include Eavesdropping attack, Replay attacks, side channel attacks, key injection attacks and ciphertext manipulation.

REFERENCES

[1] I. Odun-Ayo, O. O. Ajayi, B. Akanle, and R. Ahuja, “An Overview of Data Storage in Cloud Computing,” 2017 Int. Conf. Next Gener. Comput. Inf. Syst. (ICNGCIS), pp. 29-34, 2017, [Online]. Available: https://doi.org/10.1109/ICNGCIS.2017.9.

[2] F. Safar and R. Al King, “Data Security in Cloud Computing,” Int. J. Wireless Ad Hoc Commun., vol. 7, no. 1, 2023, [Online]. Available: https://doi.org/10.54216/ijwac.070105.

[3] W. E. Kedi, C. Ejimuda, and M. D. Ajegbile, “Cloud computing in healthcare: A comprehensive review of data storage and analysis solutions,” World Journal of Advanced Engineering Technology and Sciences, 2024, [Online]. Available: https://doi.org/10.30574/wjaets.2024.12.2.0291.

[4] R. Sivan and Z. Zukarnain, “Security and Privacy in Cloud-Based E-Health System,” Symmetry, vol. 13, p. 742, 2021, [Online]. Available: https://doi.org/10.3390/sym13050742.

[5] S. Reema, “Cloud Computing as a Solution for Security and Privacy Concerns,” Int. J. Res. Appl. Sci. Eng. Technol., 2023, [Online]. Available: https://doi.org/10.22214/ijraset.2023.49375.

[6] M. Joshi, R. Priya, and M. M. Joshi, “A Review: Analysis of Various Encryption Techniques for Securing Cloud Data,” 2022 4th Int. Conf. Adv. Comput. Commun. Control Netw. (ICAC3N), pp. 1945-1948, 2022, [Online]. Available: https://doi.org/10.1109/ICAC3N56670.2022.10074465.

[7] M. A. Althamir, A. Alabdulhay, and M. M. Yasin, “A Systematic Literature Review on Symmetric and Asymmetric Encryption Comparison Key Size,” 2023 3rd International Conference on Smart Data Intelligence (ICSMDI), pp. 110-117, 2023, [Online]. Available: https://doi.org/10.1109/ICSMDI57622.2023.00027.

[8] J. Khudair, K. A. Ghan, and M. R. B. Baharon, “Comparative Study in Enhancing AES Algorithm: Data Encryption,” Wasit Journal for Pure Sciences, 2023, [Online]. Available: https://doi.org/10.31185/wjps.100.

[9] A. V., A. P. Nirmala, B. K., Aldred Christi, and N. A., “A Review on Cloud Cryptography Techniques to Improve Security in E-health Systems,” 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), pp. 100-104, 2022, [Online]. Available: https://doi.org/10.1109/ICCMC53470.2022.9753999.

[10] A. Emmanuel, O. Aderemi, A. O., and A. O., "A Note on Time and Space Complexity of RSA and ElGamal Cryptographic Algorithms," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 12, no. 7, 2021, [Online]. Available: https://doi.org/10.14569/ijacsa.2021.0120716.

[11] T. Rezk, R. Bhargavan, and M. Blanchet, "Security Analysis of ElGamal Implementations," Inria, 2018.

[12] M. Kumar, R. Sharma, and P. Bhushan, "An Optimal and Efficient Data Security Technique Through Crypto-Stegano for E-Commerce," Computational Intelligence and Neuroscience, vol. 2023, no. 9906588, pp. 1-10, 2023, [Online]. Available: https://doi.org/10.1155/2023/9906588.

[13] H. Guo, X. Li, and J. Zhang, "A Novel Image Encryption Algorithm Based on ElGamal Cryptography and Chaotic System," Physica Scripta, vol. 98, no. 11, pp. 1-12, 2023, [Online]. Available: https://doi.org/10.1088/1402-4896/98/11/115251.

[14] V. O. Waziri, J. Ojeniyi, H. Danladi, A. Isah, A. S. Magaji, and M. Abdullahi, "Network Security in Cloud Computing with Elliptic Curve Cryptography," Network and Communication Technologies, vol. 2, no. 2, pp. 43-58, 2013, [Online]. Available: https://doi.org/10.5539/nct.v2n2p43.

[15] V. Satheesh and D. Shanmugam, "Implementation Vulnerability Analysis: A case study on ChaCha of SPHINCS," 2020 IEEE International Symposium on Smart Electronic Systems (iSES), pp. 97-102, 2020, [Online]. Available: https://doi.org/10.1109/iSES50453.2020.00032.

[16] S. H. Tariq, R. P. Singh, and J. Gill, "Preserving Privacy in Industrial IoT: A Machine Learning Framework Enhanced by Poly1305 Encryption," International Journal for Research in Applied Science and Engineering Technology, 2023, [Online]. Available: https://doi.org/10.22214/ijraset.2023.55767.

[17] Y. Peng, W. Zhao, F. Xie, Z. Dai, Y. Gao, and D. Chen, "Secure cloud storage based on cryptographic techniques," J. China Univ. Posts Telecommun., vol. 19, pp. 182-189, 2012, [Online]. Available: https://doi.org/10.1016/S1005-8885(11)60424-X.

[18] D. Salman and N. Sulaiman, "A Review of Encryption Algorithms for Enhancing Data Security in Cloud Computing," AlKadhim J. Comput. Sci., 2024, [Online]. Available: https://doi.org/10.61710/kjcs.v2i1.68.

[19] S. Ahmad and A. Garko, "Hybrid Cryptography Algorithms in Cloud Computing: A Review," 2019 15th Int. Conf. Electron. Comput. Computat. (ICECCO), pp. 1-6, 2019, [Online]. Available: https://doi.org/10.1109/ICECCO48375.2019.9043254

[20] H. Sharma, R. Kumar, and M. Gupta, "A Review Paper on Hybrid Cryptographic Algorithms in Cloud Network," 2023 2nd Int. Conf. Innov. Technol. (INOCON), pp. 1-5, 2023, [Online]. Available: https://doi.org/10.1109/INOCON57975.2023.10101044.

[21] S. Kaur and L. Jain, "A Hybrid Cryptographic Scheme for Improving Cloud Security Using ECC and TDES Algorithms," Curr. J. Appl. Sci. Technol., vol. 39, no. 4, pp. 73-84, 2020, [Online]. Available: https://doi.org/10.9734/CJAST/2020/V39I4731184.

[22] R. Mendez, "Enhance Data Storage Security DNA Cryptography in Cloud," 2020.

[23] A. Kavya and S. Acharva, "A Comparative Study on Homomorphic Encryption Schemes in Cloud Computing," 2018 3rd IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. (RTEICT), pp. 112-116, 2018, [Online]. Available: https://doi.org/10.1109/RTEICT42901.2018.9012261

[24] M. Madhavi and D. Sivareddy, "A Review of Research on Dual-Server Public-Key Encryption with Keyword Search for Secure Cloud Storage," Int. J. Res., vol. 4, pp. 630-638, 2017.

[25] A. Gadad and D. Anbusezhiyan, "Cloud security: literature survey," Int. J. Electr. Comput. Eng. (IJECE), vol. 13, no. 4, pp. 4734-4742, 2023, [Online]. Available: https://doi.org/10.11591/ijece.v13i4.pp4734-4742.

[26] A. Sharma, R. S. Thakur, and S. Jaloree, "Analyzing the Behavior of Symmetric Algorithms Usage in Securing and Storing Data in Cloud," Springer, pp. 381-389, 2018, [Online]. Available: https://doi.org/10.1007/978-981-10-5523-2_35.

[27] K. Sasikumar and S. Nagarajan, "Comprehensive Review and Analysis of Cryptography Techniques in Cloud Computing," IEEE Access, vol. 12, pp. 52325-52351, 2024, [Online]. Available: https://doi.org/10.1109/ACCESS.2024.3385449.

[28] S. Njuki, J. Zhang, E. C. Too, and R. Richard, "An Evaluation on Securing Cloud Systems based on Cryptographic Key Algorithms," Proceedings of the 2nd International Conference on Algorithms, Computing and Systems, 2018, [Online]. Available: https://doi.org/10.1145/3242840.3242853.

[29] N. L. Kodumru and M. Supriya, "Secure Data Storage in Cloud Using Cryptographic Algorithms," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018, [Online]. Available: https://doi.org/10.1109/ICCUBEA.2018.8697550.

[30] D. S. S. Marar, J. M. Joyson, and L. Ashish, "Secure Cloud Storage Using Different Algorithms in Cryptography," International Journal for Research in Applied Science and Engineering Technology, 2022, [Online]. Available: https://doi.org/10.22214/ijraset.2022.43500.

[31] A. Poduval, A. Doke, H. Nemade, and R. Nikam, "Secure File Storage on Cloud using Hybrid Cryptography," International Journal of Computer Sciences and Engineering, vol. 7, no. 1, 2019, [Online]. Available: https://doi.org/10.26438/ijcse/v7i1.587591.