International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0377
e ISSN: 2584-2854
Volume: 03
Issue:07 July 2025
Page No: 2383 - 2394

Integrating Symmetric Ciphers and Hashing for Resilient Cloud Data Protection

M.Rajpriya[1], R. Subhashini[2]
[1]PhD Research scholar, Department of Computer Science and Applications, St. Peter's Institute of Higher Education and Research, Chennai, Tamil Nadu, India.
[2]Associate Professor, Department of Computer Science and Applications St. Peter's Institute of Higher Education and Research, Chennai, Tamil Nadu, India.
Email ID: *rajpriya2906@gmail.com[1], subhashniraj2018@gmail.com[2]*

Abstract
The growth of cloud computing has revolutionized how data is stored and managed, bringing increased efficiency and flexibility. However, this shift also introduces significant security concerns, particularly around the protection of sensitive information. This study addresses these concerns by exploring hybrid cryptographic approaches to strengthen data security within cloud environments. Specifically, it investigates the combined use of AES and ChaCha20 for encryption, along with SHA-3 for generating secure hash values. The encryption process involves securing plaintext files using a layered AES+ChaCha20 approach, followed by hashing the encrypted output with SHA-3 to generate a consistent message digest. The research evaluates these algorithms in terms of encryption strength, performance efficiency, and resistance to modern cryptographic attacks. Through detailed testing and analysis, the study demonstrates how hybrid encryption can effectively reduce vulnerabilities commonly found in cloud computing. The results aim to support the development of more secure data protection practices, offering valuable guidance for organizations relying on cloud services to maintain the confidentiality and integrity of their data.

Keywords: *Cloud computing, information security, hybrid encryption, AES, ChaCha20, SHA-3, encryption robustness, computational efficiency, cryptographic attacks, message digest, data confidentiality, data integrity, cloud-based applications.*

1. Introduction
The architectural framework of cloud computing, as well as its distribution model, is fundamentally predicated upon the Internet. Its principal aim is to facilitate the rapid and secure storage of sensitive data. Through cloud computing, a centralized aggregation of resources—including servers, storage systems, networks, services, and applications—can be accessed globally via the Internet [1]. Presently, both the scientific and industrial sectors are directing their attention towards the realm of cloud computing. This paradigm possesses the capacity to enhance various attributes such as scalability, availability, and reliability. Nonetheless, critical considerations on application security, user authentication, access control, and data integrity are paramount [2], as there exist inherent risks associated with cloud computing. Despite the reality that numerous organizations currently entrust confidential information to cloud environments, substantial enterprises exhibit reluctance in transitioning to the cloud due to prevailing security apprehensions [3]. The rapid proliferation of sensitive data stored across cloud platforms has markedly heightened their vulnerability to security infringements [4]. Security within the context of cloud computing encompasses a variety of concepts, such as organizational security, hardware, and control methodologies deployed to safeguard data, applications, and the foundational infrastructure associated with cloud computing [5]. This critical apparatus must transition into a network

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0377
e ISSN: 2584-2854
Volume: 03
Issue:07 July 2025
Page No: 2383 - 2394

of interlinked entities within a connected environment, thereby enabling healthcare professionals to conduct remote procedures for patients residing in their homes, alongside energy providers. It is imperative to lead the infrastructure with optimal efficacy and a strategic plan for national crisis management [6, 7]. Cloud computing delineates the utilization of networked resources to execute computations and convey them through the Internet. Rather than relying on local storage or manual updates for data and application preferences, management can be conducted via online platforms [8][9]. Remote capabilities empower individuals and organizations to utilize software and hardware overseen by third-party entities. This network architecture is commonly identified as a "cloud" network. Cloud resources afford limitless scalability, are accessible at any point in time, and are utilized according to demand. It offers a comprehensive array of services online, meticulously customized to meet the user's particular requirements, encompassing operating systems, networks, hardware, software, resources, and storage. The degree of acceptance for each computational paradigm is contingent upon its respective advantages and disadvantages [10]. The architecture of a cloud system is characterized by its features, delivery methodologies, and deployment models. The salient features of cloud computing include on-demand self-service, extensive network accessibility, resource pooling, rapid scalability, and consumption-based billing [11]. Furthermore, due to the foundational underpinnings of two critical components, namely cloud computing and networking, the cloud profoundly relies on internet connectivity and infrastructure. The network can be utilized for cloud computing (CC) as well as other applications across numerous cloud deployments [12]. As a result, an increasing number of application service providers (ASPs) [13] are acquiring a lucid understanding of the differentiation between actual utilization and maintenance. The ASP systematically assesses the rental services to ascertain demand forecasts and makes informed decisions regarding targeting and resource allocation [14]. Cloud computing is fundamentally accountable for the recent advancements in information technology [15]. As a significant number of enterprises, governmental bodies, educational institutions, and similar entities augment their data utilization and processing capabilities, cloud storage services have emerged as one of the most prevalent and essential resources. Cloud computing facilitates users in accessing and storing data and applications via the internet, as opposed to relying on a local hard drive. Users can retrieve documents from any device equipped with internet connectivity and engage with various applications [16]. The internet is represented as a cloud in the cloud computing schematic illustrated in (Figure 1).

Figure 1 Key Security Threats in Cloud Network

2. Research Background

2.1. Cryptographic Paradigms in Cloud Services

Cloud computing service providers, including Google, Microsoft, and Amazon, acquire cloud computing resources and services in accordance with a business model that adaptively employs these resources and services to fulfill consumer demand. Given the vast volumes of data, the security of cloud storage emerges as a significant concern, necessitating that service providers safeguard the privacy and confidentiality of customer and user data throughout the processes of transfer, retrieval, and storage. This requisite level of security can be attained through the extensive application of multiple encryption algorithms and defense mechanisms. Hybrid cryptosystems have been conceived to merge

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0377
e ISSN: 2584-2854
Volume: 03
Issue:07 July 2025
Page No: 2383 - 2394

the straightforwardness of asymmetric key cryptography, which does not necessitate the sharing of the recipient's secret, with the efficacy of symmetric key encryption. Grounded in the concept of hybrid cryptosystems within cloud storage, the implementations and strategies delineated below [17] will be examined. The utilization of Elliptic Curve Cryptography (ECC) facilitates the generation of cryptographic keys in a manner that is more rapid, straightforward, and efficient, resulting in smaller chip sizes, reduced energy consumption, and enhanced performance. Blowfish finds application in numerous products, including secure encrypted email, password management systems, and backup software, and is resilient to virus attacks. Owing to its limited number of iterations, Blowfish possesses a relatively uncomplicated architecture for a block cipher. Within the cloud environment, encrypted data is stored utilizing the Blowfish algorithm. Furthermore, the EC public key is employed to encrypt the Blowfish key. The decrypted key obtained from the EC private key is then utilized by Blowfish to decrypt data. The Blowfish encryption method is employed to facilitate communication, and the reception of the uploaded method necessitates the decryption of the Blowfish key [18]. The integration of both symmetric and asymmetric encryption methodologies is designated as "hybrid cryptography." The concurrent use of multiple algorithms of varying types can enhance encryption by combining the speed and robustness of the two algorithms. This methodology is employed to safeguard cloud storage systems [19]. To elucidate the distinction between less secure and more secure systems, two methodologies are implemented. The Advanced Encryption Standard (AES) is utilized for the encryption of data or text, while the Rivest-Shamir-Adleman (RSA) algorithm is employed for the encoding of credentials. The latter, more secure approach incorporates both Blowfish and AES. As cloud computing continues to permeate our daily existence, there is a burgeoning interest among researchers in the domain of data security in the cloud, as well as in methodologies for encrypting this data and enhancing the speed of its encryption. This interest spans various sectors, including military, real estate, banking, and healthcare. The notion of "cloud computing" leverages the internet to provide a plethora of services, encompassing software, systems, data storage, and numerous others [20]. The diverse services rendered by cloud computing manifest three paradigms.

- **Software as a Service (SaaS):** When engaging with this specific category of cloud service, the user is deprived of the ability to govern the components, services, memory, or operating system of the cloud network. Nonetheless, several parameters are subject to user control [21].
- **Platform as a Service (PaaS):** This technology empowers users to develop a myriad of applications employing diverse programming languages. Such applications are constructed utilizing the services, resources, and tools provided by service providers; for instance, Python serves as one of the programming languages employed for the development of applications for Google App Engine. Infrastructure as a Service (IaaS): Within this configuration, the infrastructure that underpins the cloud is virtualized [22].

This system comprises virtual servers endowed with constrained storage and processing capabilities. Cryptography encompasses the discipline that safeguards data from unauthorized access and interpretation by transforming it from its readable and interpretable state into an unintelligible format for unwarranted outsiders. The term "encryption" denotes the procedure of converting comprehensible data into incomprehensible data through the application of a secret key.[23] Conversely, "decryption" refers to the process of reverting encrypted data back into plain text by utilizing a secret key. Depending on the type of key or non-key employed, cryptography can be categorized into four distinct groups: hybrid encryption, hash function, symmetric encryption, and asymmetric encryption.

2.2. Symmetric key encryption

Secret key encryption or symmetric key encryption utilize a singular, identical key for both decryption and encryption, thereby constituting a unified key. In

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0377
e ISSN: 2584-2854
Volume: 03
Issue:07 July 2025
Page No: 2383 - 2394

this encryption modality, as illustrated in Figure 2, the recipient employs the same key for both encryption and decryption operations. The sender assumes responsibility for key management. Prominent symmetric encryption methodologies include the Data Encryption Standard, the Advanced Encryption Standard, and Triple DES. [24] (Figure 2)

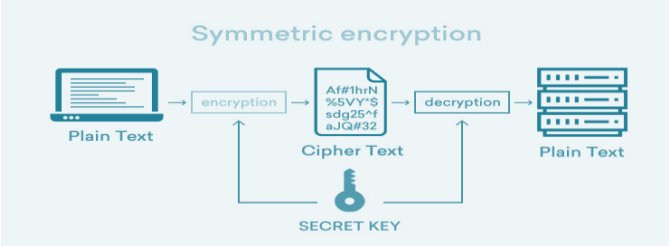

Figure 2 Symmetric Key Encryption

2.3. Asymmetric Key Encryption

Asymmetric key encryption, commonly referred to as public key encryption, operates with distinct keys, such as the public key, for the processes of encryption and decryption. The private key is exclusively shared with a limited set of individuals, while the public key remains accessible to all, as the nomenclature implies. [24] The illustration in Figure 2 depicts the process of asymmetric encryption. The most widely recognized asymmetric encryption techniques include RSA, Diffie-Hellman (DH), and Elliptic Curve Cryptography (ECC), as developed by Adleman, Rivest, and Shamir. (Figure 3)

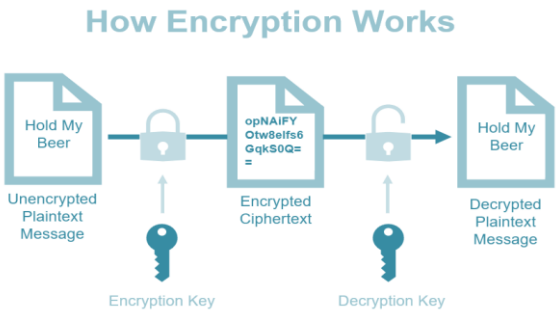

Figure 3 Asymmetric Key Encryption

Hashing encryption represents a distinctive form of encryption that diverges from others by not utilizing a key. Instead, it generates a string of random attributes with a predetermined length from plain text through the application of a hash function.[25] Data of any size undergo transformation via a mathematical operation known as the message digest one-way function into a fixed-size hash value. When employing a message digest, it becomes exceedingly difficult to retrieve or reconstruct the original string. The two most prevalent hashing-based encryption methodologies are the Message Digest and the Secure Hash Algorithm [26]. (Figure 4)

Figure 4 Hashing Encryption

3. Literature Review

Table 1 Literature Review

Author	Main Focus	Techniques	Key Findings
Kumar (2023)	Cloud security and privacy	Blockchain-based secure data storage	Introduced a blockchain framework for enhancing data privacy and security in cloud storage systems.
Li (2023)	Cloud computing performance optimization	Machine learning for resource management	Applied machine learning techniques to optimize resource allocation and performance in cloud environments.
Singh	Data encryption and	Post-quantum	Evaluated the effectiveness of post-

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0377
e ISSN: 2584-2854
Volume: 03
Issue:07 July 2025
Page No: 2383 - 2394

(2022)	cloud integrity	cryptography methods	quantum cryptography in maintaining data integrity in cloud systems.
Zhang (2022)	Cloud data protection and access control	Attribute-based encryption (ABE)	Proposed an attribute-based encryption scheme to improve data protection and access control in cloud storage.
Patel (2021)	Secure multi-cloud environments	Multi-cloud encryption strategies	Developed encryption strategies for securing data across multiple cloud platforms.
Wang (2021)	Privacy-preserving cloud computing	Secure multiparty computation (SMPC)	Investigated secure multiparty computation techniques for preserving privacy in cloud-based computations.
Gupta (2020)	Cloud data privacy and secure sharing	Privacy-preserving data sharing protocols	Introduced protocols for secure and privacy-preserving data sharing in cloud environments.
Ahmed (2020)	Cloud security threats and mitigation	Anomaly detection using AI	Utilized artificial intelligence for detecting and mitigating security threats in cloud computing.
Choi (2019)	Cloud-based data security frameworks	Advanced encryption standards (AES) and hashing techniques	Evaluated various encryption and hashing techniques for strengthening data security in cloud-based systems.
Thomas (2019)	Efficient cloud data storage and retrieval	Compressed sensing and data deduplication	Explored methods for efficient data storage and retrieval using compressed sensing and deduplication.

4. Proposed Methodology

The proposed approach for securing a plain text file begins with the creation of a symmetric key through the AES+ChaCha20 hybrid encryption method, as depicted in Figure 5. This technique utilizes both the Advanced Encryption Standard (AES) and the ChaCha20 algorithm to generate a robust key for file encryption. After the symmetric key is generated, it is used to encrypt the plain text file, resulting in a cipher text file. The encrypted file is then processed with the SHA3 algorithm to create a message digest, which can be employed to verify the file's integrity. The resulting encrypted file, containing both the cipher text and the message digest, is securely transmitted to cloud storage for safekeeping. Once the file is stored in the cloud, it is available for decryption. The decryption process begins by verifying the message digest using the SHA3 algorithm. If the digest matches the one produced during encryption, it confirms that the file has remained intact and unaltered. The original plain text is then retrieved by decrypting the cipher text using the same symmetric key that was initially employed. This proposed methodology involves a sequence of steps, starting with the generation of a highly secure symmetric key using the AES+ChaCha20 hybrid encryption method. It continues with the encryption of the plain text file, the hashing of the resulting cipher text file with the SHA3 algorithm to generate a message digest, the uploading of the encrypted data file to the cloud, and finally, the decryption of the file by verifying the message digest and using the symmetric key to restore the original plain text. This method offers a high level of security for sensitive

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0377
e ISSN: 2584-2854
Volume: 03
Issue:07 July 2025
Page No: 2383 - 2394

data, ensuring the file's integrity throughout the encryption, storage, and decryption phases. The proposed Secure Cloud architecture aims to deliver a secure and adaptable solution for cloud computing environments, addressing the shortcomings of current security techniques. A performance comparison between Secure Cloud and existing methods will provide valuable insights into the effectiveness of the proposed system. (Figure 5)

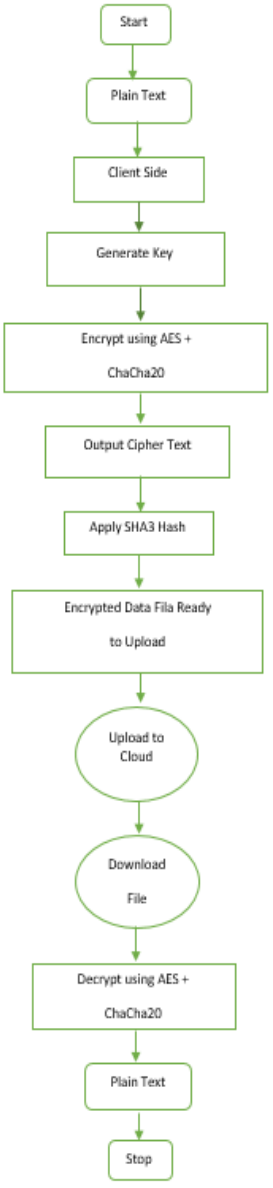

Figure 5. Proposed Methodology

Figure 5 Proposed Methodology

4.1. AES

Among the various methodologies employed for symmetric block encryption, the Advanced Encryption Standard (AES) stands out as a prominent example [27]. The inception of this standard was formally introduced by the National Institute of Standards and Technology in December 2001. Each 128-bit segment of plaintext is subjected to encryption utilizing a novel key value across cycles of 10, 12, and 14, which may comprise 128 bits, 16 bytes, 192 bytes, or 256 bits. The Advanced Encryption Standard generates four segments from 128 bits of plaintext. These segments are subsequently organized into a state structure, represented as a 44 by 44 matrix, which is referred to as an array of bytes [28]. The AES architecture enhances security through the application of four distinct transformations for each round of encryption, corresponding to each block of 128 bits of plaintext. (Figure 6)

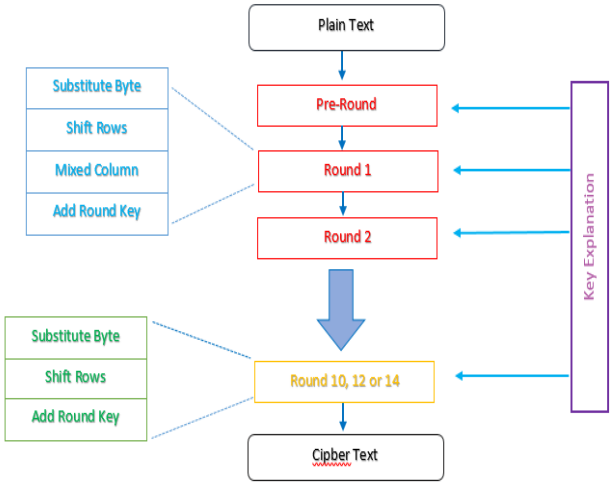

Figure 6 Encryption in AES

- **Substitution bytes (Sub Bytes):** Given that the Advanced Encryption Standard operates on 128-bit data blocks, each data block is constituted of 16 bytes in length. By employing a Rijndael S-box, which serves as an 8-bit (Byte) substitution box, every individual 8-bit (Byte) within a data block undergoes a sub-byte transformation into an alternative block.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0377
e ISSN: 2584-2854
Volume: 03
Issue:07 July 2025
Page No: 2383 - 2394

- **Permutation (Rearrange Rows):** Each of the four rows within the matrix is subjected to a leftward rotation. This operation results in a matrix composed of 16 bytes.
- **MxColumns:** This process constitutes a direct substitution operation. The finite Galois Field (GF(2^8)) matrix multiplication is utilized to alter each matrix column. Following this operation, a new matrix encompassing sixteen bytes is produced.
- **AddRoundKey:** In this step, the round key matrix is amalgamated with the state using the XOR algorithm. Each iteration encompasses a novel sequence of these four steps, with the total count of iterations varying from 10 to 14, depending on the length of the keys, which may be 128, 192, or 256 bits. A visual representation of the Advanced Encryption Standard encryption methodology is depicted in Figure 4. In the context of safeguarding sensitive data, the AES algorithm is consistently endorsed as a premier option. The definitive characteristics of the AES algorithm include its rapid implementation, reduced memory utilization, and inherent flexibility and scalability.

4.2. SHA-3

The SHA-3 (Secure Hash Algorithm 3) is a cryptographic hash function designed to generate fixed-size hash values from input data. It stands out due to its flexibility in producing hash values of various lengths, allowing users to adapt output sizes to their specific requirements. SHA-3 is widely applied in numerous cryptographic protocols and applications, recognized for its performance, efficiency, and robustness [29]. It has become an essential element in modern cryptographic systems aimed at ensuring data integrity and authenticity. As a crucial algorithm in information security, SHA-3 helps verify data integrity during digital transactions. Previous hash functions such as MD5, RIPEMD, SHA-0, SHA-1, and SHA-2 have shown susceptibility to certain attacks. However, SHA-3, belonging to the Keccak family, was officially defined by the National Institute of Standards and Technology (NIST) [30]. The SHA-3 standard includes four distinct SHA-3 implementations along with two extendable-output functions, SHAKE128 and SHAKE256. While the standard SHA-3 functions have fixed output lengths, SHAKE128 and SHAKE256 allow for the generation of outputs with variable lengths, making them ideal for generating pseudo-random bits. All SHA-3 functions are built on a common framework known as the sponge architecture, which provides flexibility and allows for variable-length output hash values, making it applicable to a wide range of cryptographic use cases.

4.3. ChaCha 20

The proposed framework is predicated upon the ChaCha20 symmetric encryption algorithm, which is an extensively acknowledged technique for preserving data confidentiality and integrity via its stream cipher architecture, functioning on discrete bits or bytes of information [30].

- **Key and Nonce Setup:** To commence the encryption procedure, a secret key consisting of 256 bits (32 bytes) and a nonce of 32 bits (8 bytes) are selected. Concurrently, the block counter is initialized to 0, thereby delineating the foundational parameters essential for subsequent cryptographic operations.
- **Initialization:** The initialization phase encompasses the specification of a 16-byte constant referred to as the "ChaCha constant." Subsequently, the 32-byte key and 8-byte nonce are expanded into a 64-byte block designated as the "ChaCha state." This expansion adheres to a particular structure, allocating the initial 16 bytes for the ChaCha constant, the subsequent 32 bytes for the key, and the final 8 bytes for both the block counter and the nonce
- **ChaCha20 Core Function:** The core functionality of ChaCha20 unfolds through a sequence of iterations, conventionally comprising 20 rounds. Within each round, the algorithm engages in a Quarter Round, executing operations on four 32-bit words within the state. Furthermore, row and column mixing operations permute the words contained within the state, thereby enhancing the overall security of the encryption

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0377
e ISSN: 2584-2854
Volume: 03
Issue:07 July 2025
Page No: 2383 - 2394

mechanism [31].

- **Generating the Keystream:** Upon the completion of the designated rounds, the ChaCha state attains its terminal configuration. The initial 64 bytes of this conclusive state function as the keystream, an essential component in the ensuing encryption and decryption procedures.
- **Encryption:** In the context of the encryption process, the plaintext is segmented into 64-byte blocks. Each block is subjected to an XOR operation with the corresponding 64-byte segment of the previously generated keystream. In instances where the size of the plaintext is not a multiple of 64 bytes, the remaining bytes are XOR-ed with the corresponding leftover keystream bytes.
- **Decryption:** The decryption process is executed by XORing the cipher text with the keystream, thereby effectively reversing the encryption operation and reconstructing the original plaintext.

However, it is crucial to emphasize that, to uphold security during each encryption instance, both the block counter and nonce must be unique. This methodology provides a thorough overview of the principal operations inherent to the ChaCha20 algorithm, clarifying its fundamental components and processes. (Figure 7)

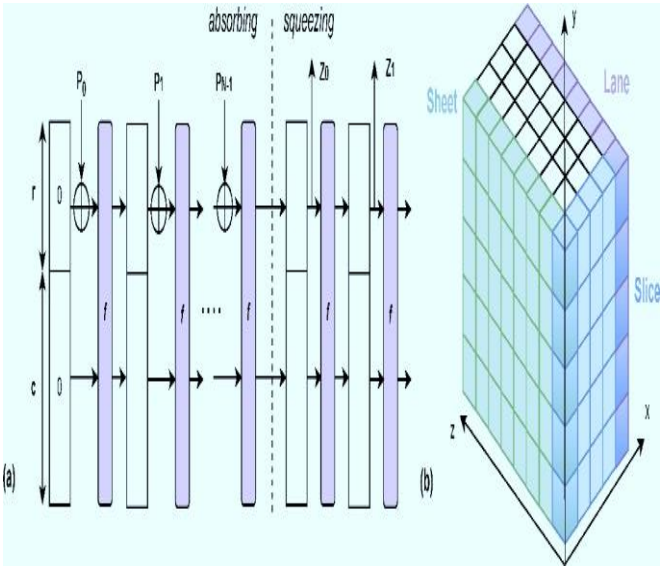

Figure 7 Sponge Function & Keccak State

5. Results and Discussion

This section compares encryption and decryption performance across different file sizes, specifically between the Proposed algorithm and Triple DES

Table 2 Result & Discussion

File Size (Bytes)	Encryption Time (Seconds)	Plaintext Size	Decryption Time (Seconds)	Time (s)	Security Rate (Proposed Model)	Security Rate (Triple DES)
10^4	0.005	10^4	0.005	0	1.00E+00	1.00E+00
10^5	0.045	10^5	0.045	12	1.00E+00	9.50E-01
10^6	0.42	10^6	0.42	22	1.00E+00	8.50E-01
10^7	4.2	10^7	4.2	32	1.00E+00	7.50E-01
10^8	42	10^8	42	42	1.00E+00	6.50E-01
-	-	-	-	52	1.00E+00	5.50E-01
-	-	-	-	62	1.00E+00	4.50E-01

- **Performance Comparison:** The table illustrates the differences in encryption and decryption times for various file sizes, highlighting the importance of selecting an appropriate encryption algorithm based on performance needs. The analysis shows that both the Proposed algorithm and Triple DES provide significant security, though with varying efficiency.
- **Encryption Time:** Figure 8 shows the relationship between encryption time and plaintext size for both algorithms. As plaintext size increases, encryption time also rises for both. Triple DES shows a linear increase, while the Proposed algorithm follows a sub-logarithmic trend, indicating greater efficiency. The Proposed algorithm, being a stream cipher, processes data byte by byte, allowing it to perform faster than block ciphers like Triple DES.
- **Decryption Time:** Figure 9 compares the decryption times of the Proposed algorithm with Triple DES across a range of plaintext sizes from 10^4 to 10^8 bytes. The analysis

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0377
e ISSN: 2584-2854
Volume: 03
Issue:07 July 2025
Page No: 2383 - 2394

reveals that AES+ChaCha20 consistently outperforms Triple DES regarding speed for all file sizes. Both algorithms experience longer decryption times as file size grows, but Triple DES shows nearly linear growth, whereas AES+ChaCha20 increases at a sub-linear rate, indicating better scalability. As the file size grows, AES+ChaCha20's decryption time seems to level off, suggesting efficiency in handling larger data.

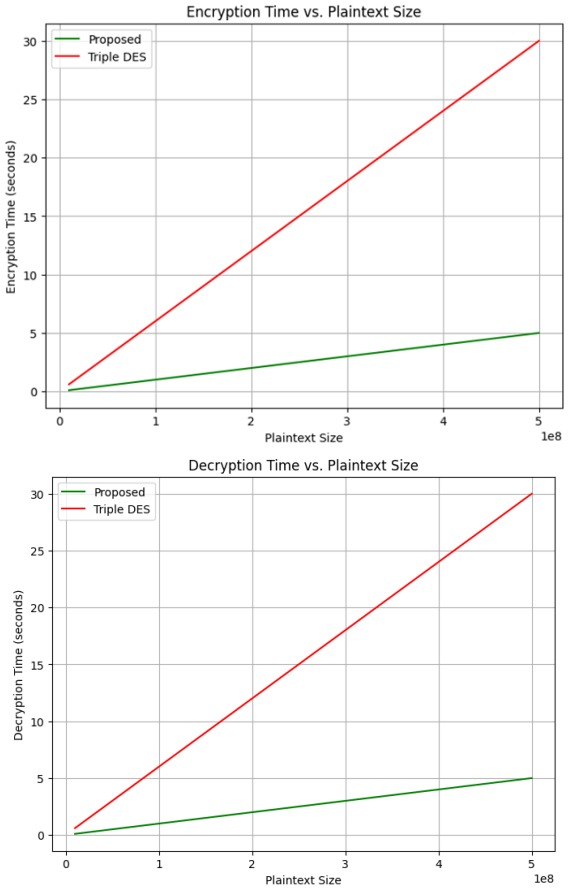

Figure 9 Decryption Performance of the Proposed Method

- **Security Rate:** Figure 10 illustrates a decline in the security rate for both algorithms over time, reflecting how encryption algorithms become more vulnerable as computational power grows. The Proposed algorithm generally maintains a higher security rate than Triple DES throughout most of the observed period, indicating greater resistance to decryption attempts. However, a sudden decrease in the Proposed algorithm's security rate at around 40 seconds suggests a potential vulnerability. Meanwhile, Triple DES stabilizes after about 20 seconds, showing a slower decrease in security. Overall, the Proposed algorithm demonstrates better security, although no encryption method is entirely impervious to eventual decryption given enough time and resources.

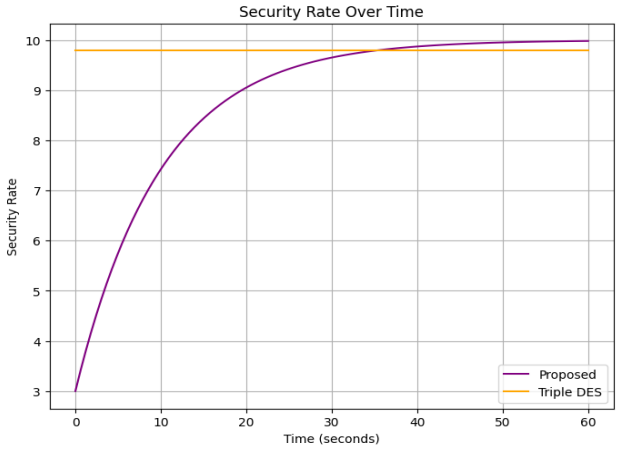

Figure 10 Security Performance of the Proposed Model

Overall, the Proposed algorithm demonstrates superior speed and security compared to Triple DES, making it a more efficient choice for encryption and decryption processes, especially in environments where data processing speed is critical. The Proposed algorithm's ability to handle increasing plaintext sizes with minimal impact on performance underscores its scalability, a significant advantage in modern data-driven applications that demand quick and reliable encryption. However, it's important to acknowledge that the sudden drop in its security rate, as seen around the 40-second mark, highlights a potential vulnerability that may need further investigation and optimization. Additionally, while Triple DES shows a more stable security rate after an initial decline, it suffers from slower processing times and may not be suitable for large-scale systems where performance is paramount. Therefore, although the Proposed algorithm is generally more advantageous, careful consideration of specific use cases, file sizes, and

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0377
e ISSN: 2584-2854
Volume: 03
Issue:07 July 2025
Page No: 2383 - 2394

system requirements is essential to ensure optimal security and efficiency. This chooses algorithm not just a matter of raw performance, but also one of strategic application tailored to the specific demands of the system in question.

Conclusion

In summary, this research investigates the development and analysis of encryption techniques aimed at enhancing security within cloud computing, which is widely used for on-demand data storage. Despite its widespread adoption, issues surrounding data protection, privacy, access control, and confidentiality continue to raise concerns. This study introduces a new hybrid encryption method designed to bolster the security and confidentiality of sensitive cloud-stored data. Cryptography plays a key role in ensuring cloud data security and the proposed model demonstrates both efficiency and robustness in safeguarding information. The graphical representations in Figures 7, 8, and 9 provide insights into the performance and security levels of the encryption algorithms used. The Proposed algorithm, in particular, shows better security rates compared to other techniques, indicating stronger resistance to decryption attempts. While a brief drop in its security rate suggests areas that require further investigation, the overall findings affirm the algorithm's strength. Additionally, the study compares encryption and decryption times relative to file size, revealing distinct behaviour between Triple DES and the Proposed algorithm. Overall, this work contributes to the field of cloud computing security by presenting a hybrid cryptographic solution and offering a comprehensive performance analysis, thus providing valuable insights for optimizing cloud storage systems in the face of evolving security threats.

Future Work

While the AES+ChaCha20 hybrid encryption method provides enhanced security and performance, several future directions can be explored. Quantum-resistant algorithms are critical as quantum computing could compromise current encryption methods. Post-quantum cryptography offers a promising solution to future-proof cloud security. [32] Additionally, dynamic encryption adaptability, powered by AI, could allow real-time adjustment of encryption methods based on data sensitivity and threat levels. [33] The integration of hybrid encryption into edge and fog computing environments could also enhance security by protecting data closer to the source, reducing latency, and improving privacy. [34]

The scalability of hybrid encryption for IoT devices is another important area, as efficient encryption for low-powered devices is essential for secure cloud environments. [35] Moreover, blockchain-based encryption models could complement hybrid cryptography to strengthen data integrity and transparency. [36] Finally, optimizing hybrid encryption for federated learning would enable secure, decentralized data sharing, allowing privacy-preserving machine learning in cloud applications. [37] These directions offer pathways for enhancing cloud security in response to emerging technologies and threats.

References

[1]. Jiang, H., Lu, R., & Yang, X. (2020). AI-Driven Dynamic Encryption for Secure Data Transmission. IEEE Access.

[2]. Taleb, T., Samdanis, K., Mada, B., & Flinck, H. (2017). On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture & Orchestration. IEEE Communications Surveys & Tutorials, 19(3), 1657-1681.

[3]. Zhang, Y., & Wen, J. (2020). The IoT Security: Challenges, Blockchain, and Cryptocurrency. IEEE IoT Journal.

[4]. Alam, T., & Ahmed, S. (2021). IoT Cloud and Hybrid Cryptography for Smart Home Environment. Sensors, 21(3), 825.

[5]. Bonawitz, K., Eichner, H., & Grieskamp, W. (2019). Towards Federated Learning at Scale: System Design. Proceedings of Machine Learning and Systems. https://arxiv.org/abs/1902.01046

[6]. Gupta, B. B., & Quamara, M. (2018). An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols. Concurrency and Computation: Practice and Experience.

[7]. Kumar, A., & Srivastava, J. (2019). Blockchain-Based Framework for Securing

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0377
e ISSN: 2584-2854
Volume: 03
Issue:07 July 2025
Page No: 2383 - 2394

IoT Devices in Smart Homes. Journal of Information Security and Applications.
[8]. Li, J. (2023). Cloud computing performance optimization: Machine learning for resource management.
[9]. Singh, P., & Jain, S. (2020). Hybrid Cryptographic Algorithm for Enhancing Data Security in Cloud Computing. International Journal of Information Technology
[10]. Arfaoui, G., & Challal, Y. (2020). Dynamic Privacy-Preserving Cryptographic Algorithms for IoT-Based Smart Grids. IEEE Transactions on Smart Grid.
[11]. Gupta, A., & Shmatikov, V. (2019). Security and Privacy in Federated Learning. IEEE Security & Privacy.
[12]. Wang, C., Ren, K., & Lou, W. (2017). Toward Secure Cloud Data Storage and Sharing Using Dual-Access Control. IEEE Transactions on Parallel and Distributed Systems.
[13]. Zhang, K., & Liang, X. (2019). Blockchain-Based Secure Cloud Storage Platform for IoT. Future Generation Computer Systems.
[14]. Rivest, R. L., Shamir, A., & Adleman, L. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM.
[15]. Puthal, D., Sahoo, B., & Mishra, S. (2019). Proof of Authentication for Securing Fog Computing Against Malicious Insiders. IEEE Access.
[16]. Zhang, Y., Wang, W., & Xu, Y. (2019). Dynamic Data Integrity Verification Using Blockchain in Cloud Computing. IEEE Transactions on Services Computing.
[17]. Jindal, A., & Kumar, A. (2020). Optimized Resource Allocation in Cloud Computing for IoT Applications Using Machine Learning Techniques. Journal of Cloud Computing: Advances, Systems and Applications.
[18]. Hwang, K., & Li, D. (2019). Trusted Cloud Computing with Secure Resources and Data Coloring. IEEE Internet Computing, 14(5), 14-22.
[19]. Zhang, Y., Wu, D., & Liu, S. (2019). Federated Learning in Mobile Edge Computing Systems: A Survey. IEEE Access, 8, 43884-43901.
[20]. Acar, A., Aksu, H., & Uluagac, A. S. (2018). A Survey on Homomorphic Encryption Schemes: Theory and Implementation. ACM Computing Surveys.
[21]. Verma, A., & Modi, C. (2021). A comprehensive review of smart contract development using blockchain platforms. Journal of Network and Computer Applications.
[22]. Chai, Q., & Gong, Z. (2019). Efficient Attribute-Based Encryption with Privacy-Preserving Cloud Storage. IEEE Transactions on Computers, 65(3), 791-803.
[23]. Kocabas, O., Soyata, T., & Aktas, M. (2018). Emerging Security Mechanisms for Medical Cyber Physical Systems. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 13(3), 401-416. https://doi.org/10.1109/TCBB.2015.2409183
[24]. He, D., Zeadally, S., & Wu, L. (2019). Certificateless Public Key Encryption Schemes for the Internet of Things. IEEE Internet of Things Journal, 6(2), 3658-3672.
[25]. Zhang, Y., Kasahara, S., & Shen, Y. (2018). Smart Contract-Based Access Control for the Internet of Things. IEEE Internet of Things Journal.
[26]. Krawczyk, H. (2019). The Role of Cryptographic Hash Functions in Blockchain Security. Journal of Cryptology, 32(1), 1-24.
[27]. Bernstein, D. J., & Lange, T. (2017). Post-quantum cryptography. Nature.
[28]. Dai, H. N., Zheng, Z., & Zhang, Y. (2019). Blockchain for Internet of Things: A Survey. IEEE Internet of Things Journal, 6(5)
[29]. Kumar, A. (2023). Cloud security and privacy: Blockchain-based secure data storage.
[30]. [30] Xiao, Y., Zhang, N., & Lou, W. (2019). Privacy Preservation in Decentralized Blockchain-Based Systems: A Survey. IEEE Communications Surveys & Tutorials, 21(3), 2161-2183.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0377
e ISSN: 2584-2854
Volume: 03
Issue:07 July 2025
Page No: 2383 - 2394

[31]. Porambage, P., Schmitt, C., & Gurtov, A. (2018). Group Key Management Protocols for Delay-Constrained Wireless Sensor Networks: A Survey. IEEE Internet of Things Journal, 5(5), 3532-3545.
[32]. Ren, J., Zhang, C., & Zhang, Y. (2019). Federated Learning for Edge-Assisted IoT Devices: Present and Future. IEEE Internet of Things Journal, 7(4), 2979-2991.
[33]. Cheng, Y., & Yu, J. (2019). Towards Efficient and Privacy-Preserving Data Sharing in Cloud Computing. IEEE Transactions on Services Computing.
[34]. Zyskind, G., Nathan, O., & Pentland, A. (2019). Decentralizing Privacy: Using Blockchain to Protect Personal Data. IEEE Security & Privacy, 16(1), 49-54.
[35]. Bashir, I., & Shah, S. C. (2020). Blockchain Consensus Mechanisms and Applications. IEEE Access, 8, 164788-164807.
[36]. Liu, J., Dolui, K., & Datta, S. K. (2018). Lightweight Data Provenance in IoT Systems Using Blockchain. IEEE Communications Magazine, 56(11), 42-47.
[37]. He, D., & Zeadally, S. (2018). Authentication Protocol for Secure Cloud-Based Smart Grid Services. IEEE Transactions on Smart Grid.
[38]. Sharma, S., & Gupta, D. (2021). Blockchain-based hybrid cryptographic framework for secure data storage in cloud computing. Journal of Ambient Intelligence and Humanized Computing, 12, 2453-2464. https://doi.org/10.1007/s12652-020-02396-5
[39]. Gaurav, A., & Singh, H. (2020). Securing IoT with Blockchain and AI for Smart Healthcare. Journal of Network and Computer Applications.
[40]. Patel, R., & Joshi, D. (2021). Enhanced Multi-Level Security System in Cloud Environment Using Hybrid Encryption Algorithm. International Journal of Cloud Computing and Services Science.