GMR
The Original

A CLOUD-BASED DATA PRIVACY FRAMEWORK USING HYBRID ENCRYPTION ALGORITHMS FOR SECURE DATA TRANSMISSION

Nagaraju Pamarthi[1], Nagamalleswara Rao N[2]

[1]Research Scholar, Acharya Nagarjuna University College of Engineering and Technology, Guntur, AP, India*
[2] Professor & HoD, Department of Computer Science and Engineering, RVR & JC College of Engineering, Guntur, AP, India

ABSTRACT
The increasing popularity of cloud computing is creating an increased demand for reliable, flexible, and energy efficient encryption techniques. In this paper we present a new hybrid encryption framework that uses AES to provide fast symmetric block ciphers, RSA to allow for secure asymmetric public-key cryptography, and the bitwise XOR operator to add additional key-space through key masking and/or complexity to previously encrypted data. We propose a three layered approach using obfuscated pre-encrypted data, a standard AES encryption method and then applying a bitwise XOR mask over the AES encryption key; followed by a secure public-key exchange based upon RSA or MECC. Finally, an optional layer of obfuscation can be applied in order to protect against the most common attack vectors currently being used. Our simulations were run using varying input sizes (from 2KB to 12KB) which showed our method produced the fastest encrypt and decrypt times (2.09s and 2.45s respectively), the largest throughput values (.0196 for encrypt and .0167 for decrypt) and minimal power usage of 5.56. Furthermore, compared to other hybrid algorithms (AES+RSA, AES+DH, 3DES+RSA, etc.), we also found our method provided the highest level of security at 96. Also, when comparing our work with similar studies (OB-MECC at 94, ECC at 81, and Base64 at 72), it was determined that our method exceeded them in terms of security. These results confirm the effectiveness of the proposed hybrid model in providing strong cryptographic security while optimizing performance and energy efficiency for cloud-based storage and transmission systems.

KEYWORDS: Cloud computing, Hybrid encryption, data privacy, key management

1. INTRODUCTION

The exponential increase in data creation has compelled enterprises to consistently enhance their data storage infrastructure. Businesses of today have adopted cloud technology due to the types and volume of information they process, as well as the number of users of the business. Many organizations have moved their applications to the cloud primarily to resolve the issues they face surrounding security [1-4], information, and business, as well as other business-related risks to include physical risk, operational risk and many other aspects of an organization's operations. The advent of cloud computing has provided an economical and time saving option and thus enterprises can increase their storage capacity by outsourcing data to cloud environments [5]. Cloud services are already used by about 94 percent of organizations and more than 100 zetta bytes (1 trillion gigabytes) of data will be stored in the cloud by 2025. This emphasizes the increased reliance of businesses on cloud-based storage. Searchable encryption refers to cryptographic technique that tends to improve storage and retrieval of encrypted data [6-11]. The technology has a significant potential to businesses through safe and quick scanning and updating of encrypted data. Despite the convenience of cloud storage, there are serious security issues, particularly data privacy [12]. By using the third-party cloud service providers, the risk involved is that the provider will access, manipulate, or monitor stored data in addition to intercepting communication between the user and the server. In order to control these risks, a number of companies prefer encrypting their information before sending them to the cloud. However, despite the fact that encryption ensures the security of data, it inhibits effective data retrieval. The retrieval may reveal sensitive information to the server, thus undermining the data privacy in the process of communication.

Cloud data storage, while its many benefits, presents significant security risks that need thorough analysis to provide a solid solution and mitigate the issues associated with local data storage [13]. Proof of retrievability (POR) refers to the assurance that the integrity of stored data is maintained and remains unaltered by external resources. These verification systems prevent alterations to cloud storage archives by routinely assessing the data for potential modifications, hence preserving integrity. The rapid production of data complicates the ability of small companies to replace or grow their infrastructure when substantial data storage is necessary.

Cloud computing is regarded as the only remedy for the escalating storage expenses of IT organizations. The expenses associated with data storage devices are exorbitant, and with vast amounts of data created every minute, it becomes financially burdensome for enterprises to often replace their technology [14]. Exclusively outsourcing the data to the cloud addresses the issue. Storing data in the cloud entails transferring the user's information to distantly situated data

centers, so relinquishing control over their own data. This engenders several complex security and privacy concerns that need resolution. Meeting the privacy and security requirements for data sharing in cloud architecture would significantly encourage a substantial number of users to use this technology, given the benefits outlined below.

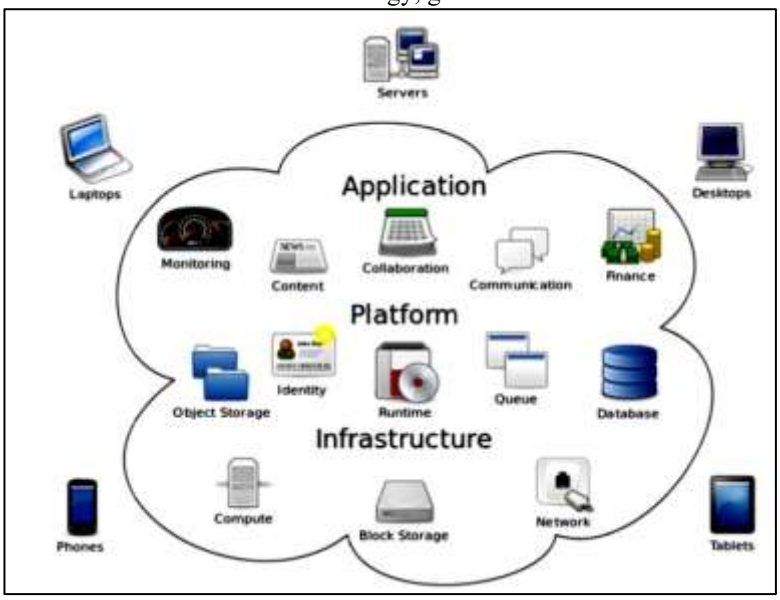


Figure 1. Cloud computing logical diagram

1.1 Issues and security in cloud

Gartner identified seven prevalent security concerns that customers should address with providers prior to selecting a cloud computing solution. The first aspect is privileged user access, pertaining to individuals authorized to keep and access client data. Clients should inquire with Cloud Service Providers (CSPs) on their recruitment policies and the individuals accountable for these processes. The second aspect is regulatory compliance. Regulatory compliance pertains to periodic external audits and security certifications [15]. The third aspect is data location. In several nations, limitations are imposed on the international transmission of data [16]. Moreover, virtualization technologies obstruct the determination of data location, necessitating that CSPs adhere to local privacy regulations to protect the data. The fourth aspect is data segregation. This refers to the segmentation or classification of data, enabling cloud customers to access certain information without impacting that of others. Cloud Service Providers should engage security personnel to execute this task, and data segregation should be conducted by an external security agency. The fifth key factor to consider is data restoration. Data restoration is important so that you can restore your information if there is a disaster or some other unexpected situation. The CSP has to ensure that it has reliable storage and that all the data stored can be completely recovered. The sixth element is investigative support. Investigative support implies that the CSP will have mechanisms in place to track down illegal activity in the cloud. The final element is sustainable over time. One of the great things about Cloud Service Providers (CSPs) is they may allow access to your data for an extended period of time or on demand at any point in time, including after your business ceases operation. Figure 2 below illustrates the security challenges in cloud computing.

Figure 2. Security issues in Cloud computing [17]

This study develops a new hybrid encryption method that combines AES, RSA and XOR operations to improve security, confidentiality and performance of data in cloud computing applications. The multi-layered approach utilizes the powerful processing speed of AES, the safe key exchange feature of RSA, and the complexity of XOR-based key obfuscation to create a strong encryption system. Our numerous trials and performance evaluation show that the proposed technique substantially cuts down both encryption and decryption time, enhances throughput, and performs better than traditional hybrid encryption algorithms in terms of computational efficiency. Also, the framework shows a high level of resistance to usual cryptanalysis methods like brute-force and man-in-the-middle making it a scalable and viable option for secure data storage and communication in cloud-based systems.

2. REVIEW OF LITERATURE

Siyal et al., (2025) [18] introduced Adaptive Attribute-Based Honey Encryption, also known as AABHE, is a complex method that enhances data security by combining honey encryption with Ciphertext-Policy Attribute-Based Encryption, also known as CP-ABE. The AABHE ensures that sensitive information is not accessible to people who are not authorized to access it. Compared to other encryption methods, such as CP-ABE, KB-ABE, and AES+ABE, the proposed method for protecting large files in HDFS achieves a security robustness of 98% and an encryption efficiency of 95% by surpassing the performance of these other methods. Through the alleviation of the Hadoop security vulnerabilities, it is through the mitigation that AABHE has enhanced its defenses against data breaches and increased the reliability of Hadoop as a solution to processing and storage of large volumes of data.

Ali et al., (2025) [19] presented an innovative secure framework that incorporates biometric-based multi-factor authentication for users, a lightweight key exchange method for IoT devices facilitating mutual authentication with a cloud server, and a hybrid encryption strategy employing elliptic curve cryptography (ECC) alongside a genetic algorithm. This study provide with authenticating the user through the use of biometrics, username and password and authentication of the IoT device. The encrypted form of sensitive sensor data is then sent to the cloud server. The authentication scheme employed in the integrity check to provide integrity is the SHA 512 algorithm employed in the framework. ECC is used to obtain a dynamically changing secret key in order to make the system even more secure adding to the cryptographic complexity and resistance to attacks. The suggested framework has a computational cost of 4 h + Et + Dt that is reduced compared to the previous methods. The efficiency of the proposed solution is evaluated by comparing it to the state-of-the-art algorithms, such as DES, AES, 3DES, RSA and Blowfish.

Gaitond et al., (2025) [20] introduced an enhanced Blockchain-Integrated Optimized Cryptographic Framework (BIOCF) that focuses on data privacy via Paillier Homomorphic Encryption (PHE) and incorporates a hybrid optimization model—Greylag Goose Optimization (GGO) and Crayfish Optimization (CO)—for the dynamic generation and management of cryptographic keys. Validation of the generated model is accomplished by the use of established cryptographic methods through experimentation. An incredibly quick encryption time of around 1.23 seconds was achieved by the recommended model in BIOCF, while a decryption time of less than 1.6 seconds was achieved for data amounts of up to 10 MB by the model. It takes 0.684 seconds to generate the key, and there is an additional 2.5 seconds of overhead cost. 48.5% of the available resources are used throughout the key generation process. At 250 TPS, it is possible to achieve a throughput of up to 140.23 kb/s at its highest extent. The MAPE for encryption time was 4.85%, with an MAE of 0.12, an RMSE of 0.16, and an R2 score of 0.842. The model also attained an RMSE of 0.16.

Gupta et al., (2025) [21] This paper reviews studies related to inter-cloud service providers' impact on privacy, focusing on how multi-cloud environments provide added security and accountability in reducing risks. AES (Advanced Encryption Standard) is a type of symmetric key encryption method used to encrypt/decrypt data within cloud storage, therefore providing additional security/confidentiality for users. By using the BFTM(Byzantine Fault Tolerance Model), researchers could verify that user-inputted data was accurate. The proposed AAES(Augmented Advanced Encryption Standard) solution also provides extensive protection and security of data while being retrieved and stored. The effectiveness of the AAES was tested in a simulated environment, using real-world data/losses. A feasibility study was also conducted to evaluate/compare AAES vs. other methods currently in use. The results of this study demonstrate that the AAES method is superior to other methods for addressing the problems associated with data security in the cloud.

Pandey et al., (2025) [22] presented a robust encryption algorithm for pictures, including three phases: confusion-diffusion, encryption, and optimization. Lorenz chaotic map is employed in the confusion phase to add to the level of unpredictability that already exists in the visual data. The unique hybrid chaotic map, termed the Logistic-Piecewise Linear Chaotic Map (LPWLCM), significantly improves the diffusion process. Encryption is executed using Elliptic Curve Cryptography (ECC), which provides a high degree of security with minimal key sizes. This would guarantee that the encrypted picture remains indecipherable while in its encrypted state. The optimization stage improves the cipher picture through the assistance of a Genetic Algorithm, the aim of which is to achieve the highest attainable level of strength in terms of the quality of cryptography and performance. The extensive experiments have revealed a major security and computing performance benefits of this proposed approach over previous ones. Therefore the method proposed can be a very effective measure for secure image transmission across a wide range of applications. Also the experimental data supports the claim and thereby confirms its suitability for secured digital communication systems in the real world.

Shahid et al., (2025) [23] suggested an innovative method to amalgamate blockchain technology with a Chaotic Tent map encryption strategy to address these challenges. A Blockchain-powered Chaotic Tent Map Encryption Scheme (BCTMES) has therefore been developed due to this necessity to protect image transactions. The picture is encrypted initially with Chaotic Tent map encryption technique and the image is translated into a cipher format that is resistant to various wide range of attacks. Once the encrypted photo is uploaded to the cloud, the document with the signature is uploaded to the blockchain, which provides the safety of its storage. Another safeguarding of manipulation is provided by the fact that the authenticity and integrity of the picture is proved when it is retrieved with the assistance of the consensus process of block chain. BCTMES was demonstrated to drastically increase such crucial security measures as entropy, correlation coefficient, key sensitivity, PSNR, UACI, and NPCR in accordance with a comprehensive research work related to their performance. The substantial key size enhances the system's resilience while also offering robust protection against brute-force assaults.

Singamaneni et al., (2024) [24] examined that the Key Policy Attribute-based Encryption (KPABE) standard emerges as the best suitable security framework for public key encryption. KP-ABE has been widely used to implement access control systems. Nonetheless, due to its high operational costs, it is challenging to use this encryption standard on networks with constrained resources, such as IoT and IIoT. Fortunately, the cloud must evolve into a foundational framework to support IoT functionalities; it is intriguing to use cloud resources for significant IoT-related operations. They envisioned an efficient Quantum KP-ABE architecture for cloud-centric IoT applications; thus, the suggested method focused on using the computational storage and power resources of cloud-based servers and dependable associate hosts to execute intensive operations. An assessment of the recital aims to demonstrate the effectiveness of the suggested framework. The experimental evaluation demonstrated that the computation time of the model's encryption procedure was 48% more efficient than current methodologies. Evaluations of memory occupancy effectiveness indicated that our method used 22% less cloud storage than established standards, even with 1 GB of user data. Comparing different encrypted standards, the new model achieved 18% less time for key generation compared to the traditional hash-based standards, showing that it can be used for fast cryptographic operations.

Alemami et al., (2023) [25] examined that cloud computing has proliferated across many enterprises owing to its benefits, including cost efficiency, resource aggregation, extensive network accessibility, and simplified management. It improves the efficacy of physical resources via the optimization of collective use. This study compared different encryption algorithms including: AES, DES, RSA, and IDEA in terms of security, data encipherment memory and encipherment time to establish the best encryption algorithm to implement to secure cloud data against cyber threats. The findings proved that RSA and IDEA offer a low level of security compared to that of AES, Blowfish and DES. The AES technique effectively encrypts substantial data amounts, demanding little encipherment time and surpassing other algorithms in speed, while the Blowfish approach requires limited memory use.

Priyadarshini et al., (2022) [26] presented a Cross-Breed Blowfish and MD5 (CBM) method to enhance the security of health data in the CPS cloud. The design suggested incorporates a wireless sensor network, wherein the data collected by the sensor network is sent via the network's nodes. The Fuzzied Effective Trust-Based Routing Protocol (FET-RP) is used to identify the most optimum route for data transmission. The BAO technique is used to determine the best route. The route-finding method transmits information between two locations. To validate the superior efficacy of the proposed technique, they compare its performance metrics with established methods, including RSA, Twofish, ICC, and FHEA. The performance metrics for Cross Breed Blowfish, MD5, and FET-RP reveal an encryption speed of 60 ms, a decryption speed of 55 ms, a latency of 60 s, a throughput of 97 mbps, a security level of 98, and an execution time of 57 ms, demonstrating an improvement of 10-15 times over conventional approaches. The proposed encryption demonstrates a significantly elevated security level, and our concept serves as a viable option in actual applications.

Garg et al., (2022) [27] examined a novel data security algorithm DHSE is suggested based on the hybrid encryption scheme of Advanced AES, IBE and ABE. The suggested solution categorizes the data into three levels, namely low sensitivity, moderate sensitivity, and high sensitivity. The information is shared through the formation of named-data packets (NDPs) through the labeling of names. Round number can be chosen based on the size of key; 128 bit keys need at least 10 rounds in DHSE. The encryption times for AES, IBE, and ABE are 3.25 ms, 2.18 ms, and 2.39 ms, respectively. The execution time of the DHSE encryption algorithm averages 2.07 ms, which is much lower than that of other competing methods. The average decryption times for AES, IBE, and ABE are 1.77 ms, 1.09 ms, and 1.20 ms, respectively, whereas the average decryption time for the DHSE technique is much lower at 1.07 ms. The result shows that the scheme is reasonable and could guarantee the information secrecy with minimal encryption and decryption time.

Thillaiarasu et al., (2021) [28] discussed the activation of several cloud frameworks, along with their security, confidentiality, and potential risks. The multi-cloud architecture was developed to provide dual encryption and decryption for users via a two-stage process. Cloud service customers should benefit from the prominent characteristics of the cloud using feature-based encryption standards. The primary emphasis on cloud faults, which therefore results in data loss from the cloud, is considered while implementing security measures for servers. The suggested method achieves a decrease of 0.3% and 5.62% in system time for directory creation with 10,000 and 20,000 files, respectively, in comparison to the traditional B-Tree method. The suggested encryption approach offers dual security and secrecy of data to end-users via feature-based encryption.

He et al., (2020) [29] examined as technology advances and customer requirements in information technology diversify, the prominence of cloud computing is becoming more evident. The establishment of computer infrastructure inside any business requires substantial financial investment, time, and human resources, which may sometimes exceed the firm's operating capabilities. Consequently, corporations often exploit such technologies with the aim of advancing their interests. Cloud computing and cloud services are fully open, decentralized, and, as such, represent an appealing target to attackers. This study discussed key concepts of cloud computing and its application in the context of the importance of issues of security. The suggested solution also improves security of the cloud computing platform by using data mining and decision tree algorithm. The low computational expense and the absence of dependence on the quantity of clients makes the suggested approach practical.

Nagaraju et al., (2020)[30] examined one of the emerging technologies that is in development is cloud computing (CC) receives increased attention both in academia and industries. It offers diverse such advantages as sharing computing resources, flexibility of services, cost minimization, etc. The Cloud Services Provider (CSP) shall be responsible of the data that are delivered to the cloud. Possibility of viewing the stored information and the utilization. The primary hurdle in the use of cloud is the sensitive raw data by stranger's services. Therefore, the main area of concern is Data Security (DS) and privacy is a hindrance during the implementation of the CC. There are myriads of methods in existence guaranteeing the confidentiality of the data, but do not fully protect the data. In order to beat these shortcomings, the Obfuscation is introduced in this study and Modified Elliptical Curve Cryptography (MECC) algorithm on the basis of obfuscation defending against malicious attacks as of data, which is called OB-MECC. According to the paper's shallower analysis, the processing algorithms described above were employed (i.e., SC, position updating, CC, binary conversion of data to 8-bit binary, decimal and using 2D complex forms in each and their ASCII representations, etc.) to generate the OB for the data. Then, the created OB from the previous step was encrypted using the MECC algorithm and stored as encrypted data on the cloud. The retrieved did not exist until the encrypted data was decrypted using the reverse of the encryption technique along with the reverse of the OB creation, result in the original unobfuscated data (e.g., content). All findings show that when comparing to the various existing approaches for protecting and ensuring maximum confidentiality and safety of OB data in the manner proposed using either OB-MECC. The analysis of the literature on encryption methods has revealed that current encryption methods either attempt to separately improve security or decrease computational complexity or both. However, most of these current methods will experience longer execution times, greater power consumption and will not offer much resistance to highly advanced forms of cryptography. Moreover, existing methods do not offer enough balance between security, computational efficiency and energy consumption as a unified set of criteria. Based on the information that has been gathered above, a need for an appropriate hybrid encrypting framework that gives more effective security with shorter execution times and lower power consumption is therefore warranted for use in cloud-based software applications.

3. Problem statement

As cloud computing continues to grow rapidly, a tremendous amount of sensitive information is being created, saved, and shared over the Internet through various types of distributed cloud infrastructures. The benefits of cloud platforms include scalability, flexibility, and accessibility; however, they also present major security risks such as unauthorized access to your information, loss or corruption of this data, and a variety of attacks (malicious or otherwise). Using traditional symmetric (e.g., AES) or asymmetric encryption techniques (e.g., RSA) alone in today's ever-changing environment presents many challenges. While symmetric encryption algorithms (e.g., AES) have fast encryption speeds they require secure key sharing; while asymmetric encryption algorithms (e.g., RSA) provide secure key exchange, however, they require a much higher level of processing power than symmetric algorithms. There are several existing hybrid encryption solutions available that partially address these issues but still do not provide an appropriate balance between security strength and performance/efficiency/scalability. Therefore, a robust, lightweight, and efficient encryption framework is needed that combines multiple cryptographic techniques with effective key management, low latency, and increased security against modern attack methods (specifically for data stored on and transmitted over cloud computing platforms).

4. METHODOLOGY

A previous work [30] has multi-step obfuscation with MECC, leading to strong reverse resistance and slightly better speed than ECC but still reaching a bottleneck due to preprocessing overhead, since MECC was slower. This problem urges the invention of new hybrid schemes gadgets, and the discussed approach improves security, efficiency, and attack resistance by encompassing the merits of AES, RSA, and XOR. Here a hybrid encryption framework is put forward to safeguard database transfers by conjunction of AES (symmetric), RSA (asymmetric), and XOR multi-layered masking pipeline. Initially data categorization is done; then pre-encryption obfuscation is applied to eliminate the data patterns. The data undergoes AES encryption using a separate key (K_AES) and IV, and in the meantime, XOR masking is used to hide the key. The masked key is then secured by means of RSA/MECC encryption, after which it is obfuscated to make it resistant against being found out. The ciphertext can likewise be obfuscated. All elements (ciphertext, masked key, IV, and metadata) are finally put together and sent. At the receiver's end, by means of key de-obfuscation, RSA decryption, and XOR unmasking, the AES key is recovered; similarly, ciphertext de-obfuscation and AES decryption give the pre-obfuscated data, which is, lastly, de-obfuscated to get the plaintext. Figure 3 depict the flowchart of proposed hybrid algorithm as shown below.

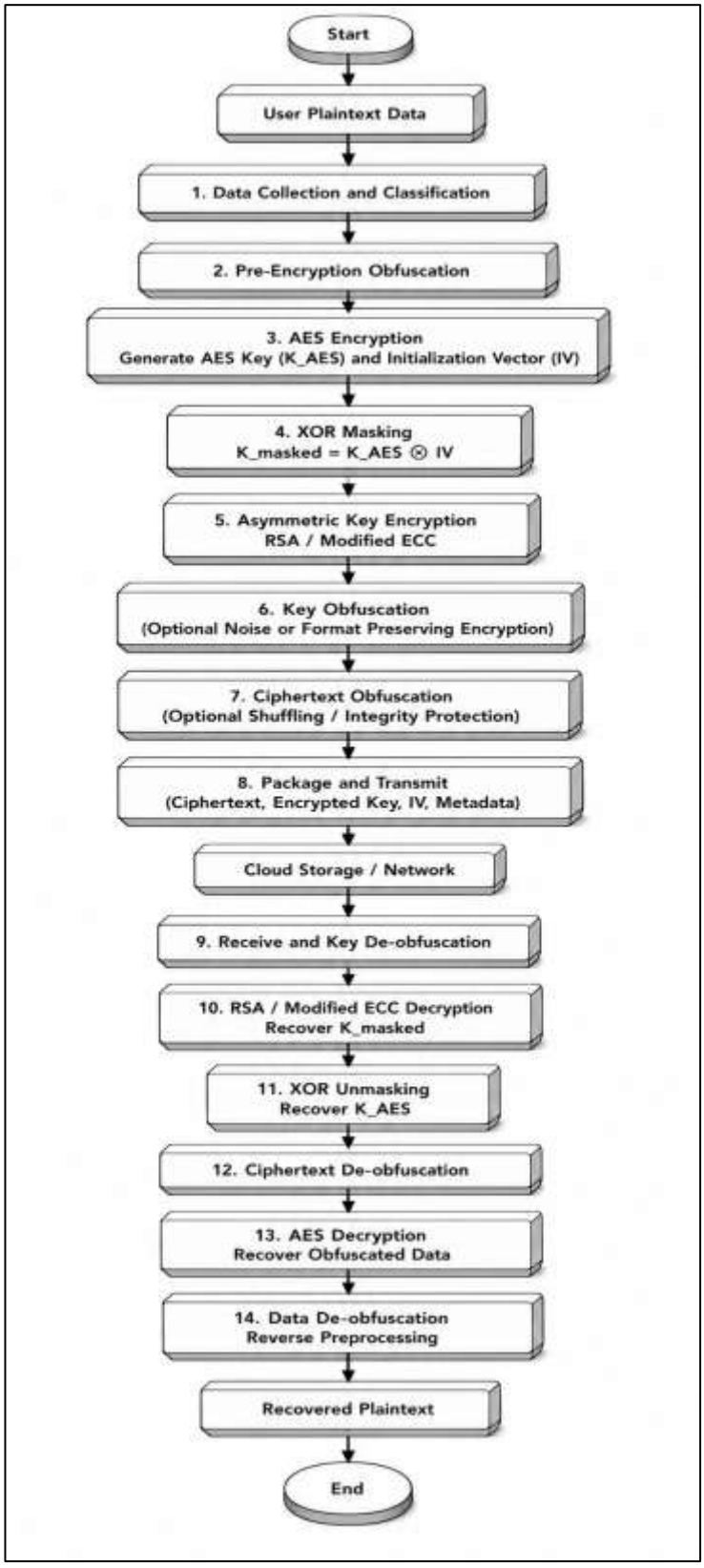


Figure 3. Flowchart of Proposed Hybrid algorithm

Step 1: Data Collection & Classification

During encryption, the first step is to gather plain text data from different sources such as databases, sensors, or user inputs. After that, this data is segmented according to its level of sensitivity, which will then be used to select the suitable encryption measures and mask methods. For example, very confidential data like health records or financial dealings might need the strongest cryptographic safeguards whereas less sensitive items like logs or metadata might be given a lower level of protection. This step is important to make sure that security measures do not overload system performance unnecessarily while at the same time optimizing resource usage.

Step 2: Pre-Encryption Obfuscation
Before any encryption is applied, the plaintext data undergoes pre-encryption obfuscation. This step modifies the data in such a way that it loses any discernible patterns or structural clues that could help attackers during cryptanalysis. Techniques may include character scrambling, byte-level shuffling, or padding with random data. The primary aim is to make the original data resistant to frequency analysis and pattern recognition, even before encryption is applied.
Step 3: AES Encryption (Generate K_AES, IV)
In this phase, the obfuscated data is encrypted using the AES, a widely adopted symmetric encryption algorithm known for its speed and security. A unique secret key (K_AES) and an Initialization Vector (IV) are generated to ensure that each encryption session is distinct, even if the same data is encrypted multiple times. The AES encryption transforms the data into ciphertext (C_data) that is unreadable without the proper decryption key and IV.
Step 4: XOR Masking
To protect the AES key (K_AES) from direct exposure, it is XOR-masked using the initialization vector (IV) as follows:
K_masked = K_AES $\oplus$ IV.
This step adds a layer of security by randomizing the AES key further, ensuring that even if one component (like IV or K_AES) is known, the actual key cannot be easily reconstructed. XOR operations are lightweight yet powerful in obscuring data at the bit level, which increases the overall robustness of the system.
Step 5: Asymmetric Key Encryption (RSA / MECC)
An XOR-mask will be used to create the AES encrypt entry (K_masked), which will then be encrypted with a public key via an asymmetric key method (RSA or MECC). This is especially vital to provide secure exchange of keys in unpredictable environments, such as cloud types. The use of asymmetric encryption to create a public/private (or two-key) pair will mean that only the desired end user will have the corresponding private key, allowing them to decrypt the transmitted AES key and thus give both confidentiality and secure transmission of the key through the use of an asymmetric encryption method.
Step 6: Key Obfuscation (Format-Preserving / Noise)
To further complicate the encrypted key and prevent pattern recognition, key obfuscation techniques are applied. These may include format-preserving encryption, which maintains the structure but not the content of the key, or the addition of synthetic noise to distort any predictable byte sequences. This ensures that the encrypted key cannot be easily identified or targeted for attack, further strengthening the encryption pipeline.
Step 7: Ciphertext Obfuscation (Optional Stego / Shuffle)
Optionally, the ciphertext resulting from AES encryption (C_data) is passed through steganographic or shuffling operations. For example, it may be embedded into seemingly benign data files like images or shuffled with other encrypted blocks. This disguises the ciphertext, reducing the chance of it being flagged or detected by malicious agents scanning for encrypted content in transit or in storage.
Step 8: Package & Transmit
All essential elements—including the obfuscated ciphertext, the encrypted and masked AES key, the initialization vector, and any relevant metadata—are packaged together and transmitted via a cloud network or stored in cloud infrastructure. This packaging step ensures that the recipient has all the necessary components to successfully decrypt and reconstruct the original data while maintaining its security throughout transmission.
Step 9: Receive & Key De-Obfuscation
On the recipient's end, the package is received and processed. The first task is to de-obfuscate the encrypted AES key, reversing any format-preserving encryption or noise masking applied earlier. This step restores the encrypted AES key to a usable form in preparation for RSA or MECC decryption.
Step 10: Asymmetric Decryption + XOR Unmask
Using their private RSA or MECC key, the recipient decrypts the obfuscated AES key to retrieve K_masked. Then, by performing the inverse XOR operation with the IV:
K_AES = K_masked $\oplus$ IV
the original AES key used in the encryption stage is recovered securely. This dual-layer security ensures that even if either the key or the IV was compromised, the full AES key remains protected.
Step 11: Receive & Ciphertext De-Obfuscation
At this point, the encrypted data (C_data_obf) also undergoes de-obfuscation to reverse any shuffling, padding, or steganographic embedding performed earlier. The objective is to restore the AES-encrypted ciphertext to its original, decryptable format (C_data).
Step 12: AES Decryption (Recover δD)
Using the retrieved AES key (K_AES) and the original IV, the AES decryption process is initiated. This transforms the encrypted data back into its pre-obfuscated form, yielding partially readable data (δD), which still contains artificial modifications applied during Step 2.
Step 13: Data De-Obfuscation (Reverse pipeline)
The final step is to reverse the initial obfuscation techniques that were applied before encryption. By removing any added noise, rearranged bits, or padding, the original plaintext structure is reconstructed accurately. This ensures that the output perfectly matches the input data prior to encryption.
Step 14: End (Recovered Plaintext)
At the end of the process, the system successfully delivers the fully decrypted and de-obfuscated plaintext data, maintaining its integrity, confidentiality, and security throughout the entire transmission and storage cycle.
The encryption and decryption process of proposed algorithm is depicted below.

Algorithm 1: Pseudocode
Encryption phase
Input: Plaintext Data (D) Output: Encrypted Package {C_data_obf, C_key, IV, meta} 1. D_classified = classify(D) 2. D_obf = pre_obfuscate(D_classified) 3. K_AES, IV = generate_AES_key_IV() 4. C_data = AES_encrypt(D_obf, K_AES, IV) 5. K_masked = XOR(K_AES, IV) 6. C_key = RSA_encrypt(K_masked, PublicKey) 7. C_key_obf = key_obfuscate(C_key) 8. C_data_obf = optional_ciphertext_obfuscate(C_data) 9. Package = {C_data_obf, C_key_obf, IV, meta} 10. Transmit (Package)
Decryption Phase
Input: Received Package {C_data_obf, C_key_obf, IV, meta} Output: Recovered Plaintext D 11. C_key = deobfuscate_key(C_key_obf) 12. K_masked = RSA_decrypt(C_key, PrivateKey) 13. K_AES = XOR(K_masked, IV) 14. C_data = deobfuscate_ciphertext(C_data_obf) 15. D_obf = AES_decrypt(C_data, K_AES, IV) 16. D = deobfuscate(D_obf) 17. Return D

5. RESULT AND DISCUSSION

Table 1 presents a comparative analysis of encryption times (in milliseconds) for different hybrid encryption algorithms across varying input sizes ranging from 2 KB to 12 KB. Among the evaluated methods—AES + RSA, AES + Diffie-Hellman, 3DES + RSA, TwoFish + RSA, and a Proposed Hybrid—the proposed hybrid consistently demonstrates the lowest execution time across all input sizes, indicating superior performance and efficiency. As input size increases, execution time rises for all algorithms; however, the proposed method scales more effectively, maintaining relatively lower processing times. This efficiency is further reflected in the total average execution time, where the proposed hybrid records 2.09 seconds, outperforming others such as AES + Diffie-Hellman (3.08s) and TwoFish + RSA (3.66s), establishing it as the most computationally efficient solution in this evaluation.

Table 1. Hybrid encryption algorithm execution time

Input Size (KB)	AES + RSA (ET MS)	AES + Diffie Hellman (ET MS)	3DES + RSA (ET MS)	TwoFish + RSA (ET MS)	Proposed Hybrid (ET MS)
2	34.91	68.79	54.85	74.56	20.9
4	64.86	140.06	102.46	187.5	38.14
6	173.63	269.31	221.47	356.86	108.34
8	391.28	432.34	411.81	523.3	278.45
10	702.46	825.46	763.96	915.31	612.2
12	1137.35	1347.05	1235.2	1602.1	1035.63
Total A/Time (Second)	2.5	3.08	2.79	3.66	2.09

The results of comparative methods on two different encryption times can be found in Table 1 and Figure 4. In terms of execution time, the proposed hybrid encryption algorithm outperforms all other methods by having the lowest average encryption time of approximately 2.09 seconds. The decreased execution time results from fewer resources being consumed by the AES method, which is utilized for symmetric encryption, while the RSA method is only being used for secure key exchange, thus resulting in less processing overhead. XOR masking, which provides very little processing overhead while enhancing key security, is also used. As input size increases from 2 KB to 12 KB; the execution time for all methods will increase, however, results show that the proposed method has better scalability than all other methods and has lower latency. The average encryption time results of 2.09 seconds support that the proposed method is better suited for cloud computing applications that require both fast and secure processing of data in real-time.

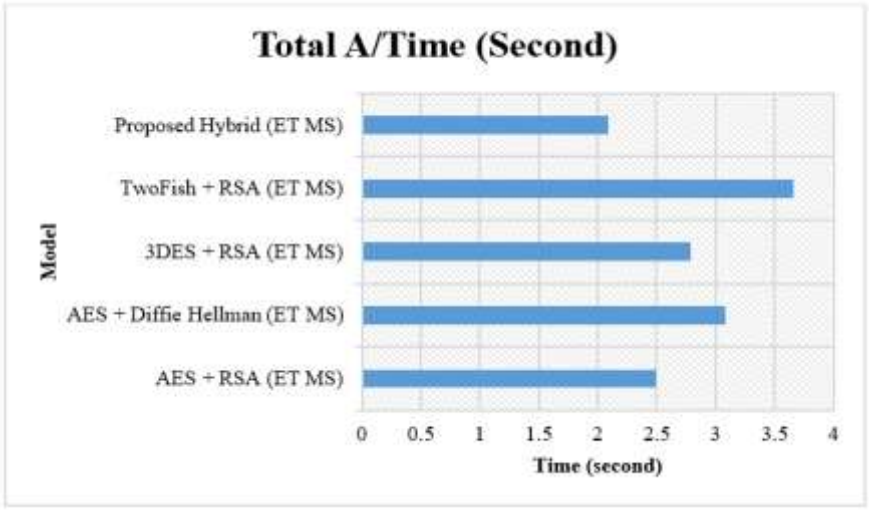


Figure 4. Comparative analysis of execution time

Table 2 provides a comparative assessment of decryption execution times (measured in milliseconds) for different hybrid encryption algorithms across varying input sizes, ranging from 2 KB to 12 KB. Performance wise, the algorithms that were compared include AES + RSA, AES + Diffie-Hellman, 3DES + RSA, TwoFish + RSA, and the Proposed Hybrid model. From the results, it is evident that the Proposed Hybrid algorithm is the fastest decryption time performer not only for smaller data but also as the data size increases. In fact, at 12 KB, the decryption time of the Proposed Hybrid algorithm is 1146.8 ms, which is much lower than TwoFish + RSA (1826.31 ms) as well as AES + Diffie-Hellman (1481.26 ms). On average, the total decryption time of Proposed Hybrid is 2.45 seconds, which clearly demonstrates its superior efficiency and scalability in decryption when compared with the older hybrid methods.

Table 2. Decryption execution time

Input size (KB)	AES+RSA	AES + Diffie Hellman	3DES + RSA	Two Fish + RSA	Proposed Hybrid
2	36.89	82.15	72.43	90.67	31.76
4	95.7	179.36	150.13	236.19	56.13
6	255.65	310.23	299.16	419.26	130.46
8	486.12	526.79	510.26	681.82	378.12
10	798.5	912.37	862.44	1104.11	711.5
12	1385.42	1481.26	1451.72	1826.31	1146.8
Total A/Time (Second)	3.05	3.49	3.35	4.36	2.45

Figure 5 visualizes the data presented in Table 2, highlighting the total average decryption time of each hybrid encryption scheme. The bar graph reveals that The Proposed Hybrid model is the one that has the lowest decryption time. The order of models with increasing decryption time is: First is the Proposed Hybrid model, Least time Then AES + RSA, Next 3DES + RSA, Followed by AES + Diffie-Hellman, and finally two fish + RSA, which has the highest decryption latency. This is a more comprehensive and clearer way of presenting the same information, which reflects the computational efficiency and performance superiority of the New Hybrid algorithm. This is the perfect solution for cloud-based systems where fast and secure data retrieval is a critical factor.

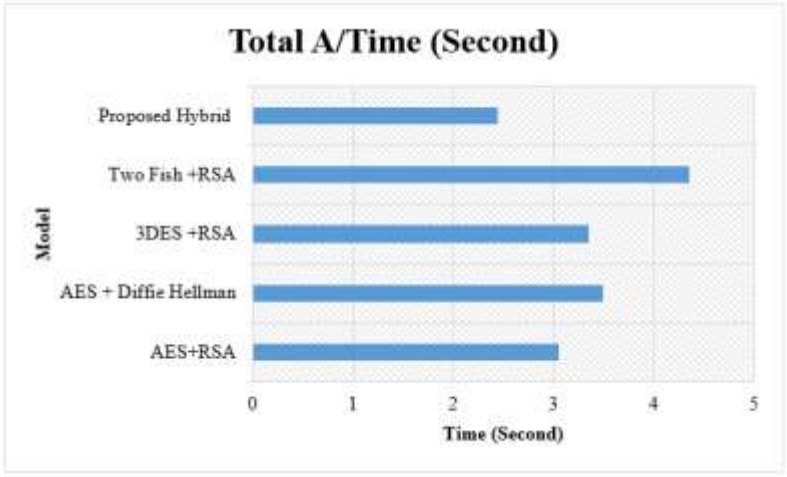


Figure 5. Comparative analysis of Decryption execution time

As shown in Table 2 and Figure 5, the proposed hybrid algorithm has the smallest decryption time of all approaches compared. The reason that this model has reduced decryption latency is due to the combined efficiencies of using AES for symmetric decryption and using RSA solely for secure key recovery, thus minimizing the amount of computation required. In addition, the process of unmasking the data via XOR uses simple bitwise operations, which provides virtually no extra processing expense. All methods showed an increase in decryption time as the input data volume increased from 2 to 12 KB. However, the proposed model provided greater scalability with lower latency than all of the other models, as evidenced by the average decryption time of 2.45 seconds, which indicates that this model is very effective in meeting the needs of cloud environments for rapid and secure access to data.

Table 3 presents the throughput values for both encryption and decryption processes of the evaluated algorithms. Throughput here refers to the amount of data processed per unit of time, measured in terms of encryption and decryption efficiency. The Proposed Hybrid algorithm achieves the highest throughput with 0.0196 for encryption and 0.0167 for decryption, indicating faster processing speed compared to other techniques like AES + Diffie-Hellman (0.0133/0.0117) and TwoFish + RSA (0.0112/0.0094). These figures demonstrate that the Proposed Hybrid not only delivers strong encryption but also ensures rapid and optimized processing, making it effective for high-performance and scalable cloud environments.

Table 3. Throughput value of Hybrid algorithm for Encryption and Decryption

Type	AES+RSA	AES + Diffie Hellman	3DES+RSA	TwoFish + RSA	Proposed Hybrid
Encryption	0.0164	0.0133	0.0147	0.0112	0.0196
Decryption	0.0135	0.0117	0.0123	0.0094	0.0167

Figure 6 graphically illustrates the encryption and decryption throughput of each hybrid algorithm. The Proposed Hybrid outperforms all other encryption models in both metrics, with visibly taller bars indicating its superior processing capacity. This performance is critical for applications that require real-time or near real-time secure communication, such as dynamic cloud databases, where both speed and security are of paramount importance.

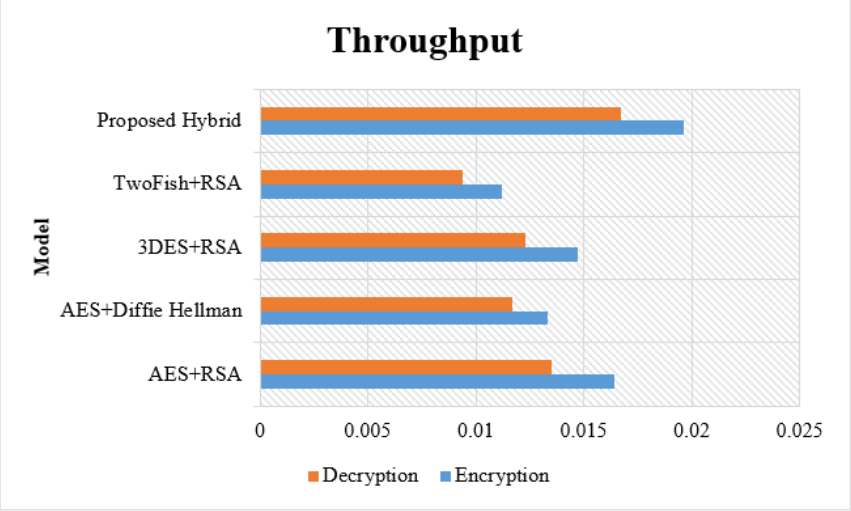


Figure 6. Throughput of Hybrid algorithm

The hybrid encryption framework has the highest throughput with respect to encryption and decryption operations as seen in Table 3 and Figure 6 due to the following; Both AES's computational efficiencies and XOR masking's lightweight characteristics contribute to much higher throughput values than AES+RSA, AES+Diffie-Hellman, 3DES+RSA and TwoFish+RSA. In the case of the hybrid encryption algorithm, the throughput value for encryption is 0.0196, and the throughput value for decryption is 0.0167. The performance of the encryption and decryption operations are both suitable for use in large-scale and/or real-time cloud storage applications that require high performance and strong security features because the proposed hybrid encryption framework was able to process larger amounts of data within shorter time periods than previous methods.

The comparative analysis in Table 4 demonstrates that the proposed hybrid algorithm achieves the lowest power loss (5.56) among all considered methods. While 3DES+RSA and AROA-BMCG [30] also provide relatively low losses (5.63 and 5.97 respectively), algorithms like AES+Diffie Hellman and AES+RSA incur significantly higher power costs (11.93 and 9.34). This highlights the superiority of the proposed hybrid approach in terms of energy efficiency, making it more practical for real-world deployment in power-constrained environments.

Table 4. Power loss of hybrid algorithm

Metrics	AES+RSA	AROA-BMCG [31]	AES + Diffie Hellman	3DES+RSA	TwoFish + RSA	Proposed Hybrid
Power loss	9.34	5.97	11.935.63	5.63	8.71	5.56

As shown in Figure 7, the proposed hybrid algorithm achieves the minimum power loss (5.56), outperforming conventional hybrid schemes. While 3DES+RSA (5.63) and AROA-BMCG (5.97) [31] show competitive results, other methods such as AES+RSA (9.34), TwoFish+RSA (8.71), and AES+Diffie Hellman (11.93) exhibit significantly higher power losses. This indicates that the proposed hybrid approach is more suitable for energy-constrained environments.

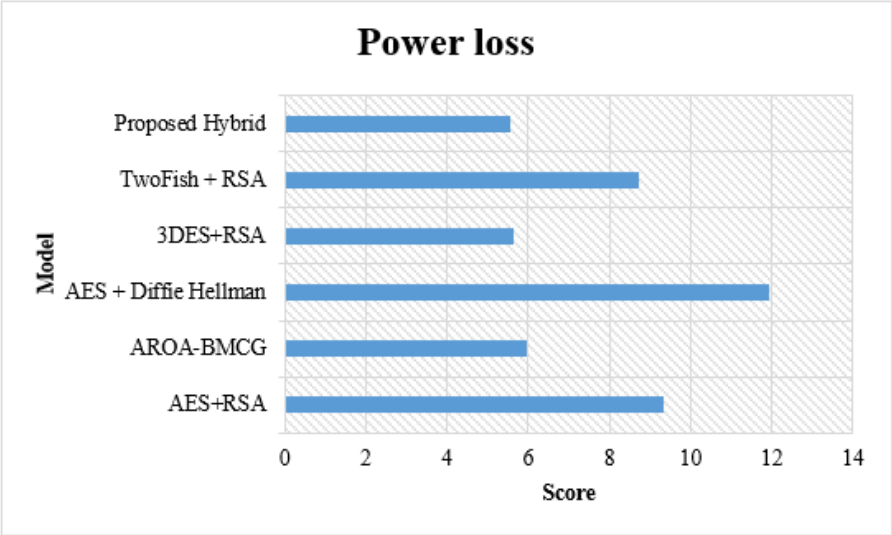


Figure 7. Power loss of hybrid algorithm

The proposed hybrid encryption method exhibits the least amount of power loss of any of the methods considered, as shown in Table 4 and Figure 8. The main reason for the improved energy efficiency of the proposed hybrid encryption algorithm is due to the low computational requirements of both AES and XOR operations (compared to traditional hybrid techniques) resulting in lower amounts of processing energy being used, due in large part to only using RSA for key exchange instead of using it to encrypt the entire data. This clearly helps to significantly reduce the overall processing burden of the proposed framework, which recorded a total power loss of 5.56, significantly lower than that of AES+RSA (9.34), AES+Diffie-Hellman (11.93), and TwoFish+RSA (8.71). Therefore, the reduction in power consumption in the proposed model is an indicator that the model is very well-suited for energy-constrained environments, such as those found in IoT devices, mobile cloud systems, and resource-constrained applications, where it is essential to reduce energy consumption in order to increase system efficiency and the lifetime of the systems.

The security performance of the newly proposed hybrid encryption method is assessed against established and well-known techniques. In addition to the numerical comparisons shown in Table 5, Figure 8 displays a graphical representation of these same comparisons. The new hybrid system achieves a superior security benchmark of 96 as compared to existing encryption methods of OB-MECC [30], ECC, and Base64. This data effectively demonstrates that the combination of AES, RSA, XOR Masking, and Obfuscation within the framework of the new hybrid approach provides the ability to resist attacks on cryptographic systems for a much longer duration than would be exhibited by traditional and single-solutions systems.

Table 5. Security level of proposed model with existing technique

S. No	Techniques	Security Level
1	Proposed algorithm	96
2	OB-MECC [30]	94
3	ECC	81
4	Base 64	72

Figure 8 reinforces this comparison through a bar chart, where the tallest bar represents the proposed algorithm's superior security performance, making it evident at a glance that it offers the most robust protection among the evaluated methods.

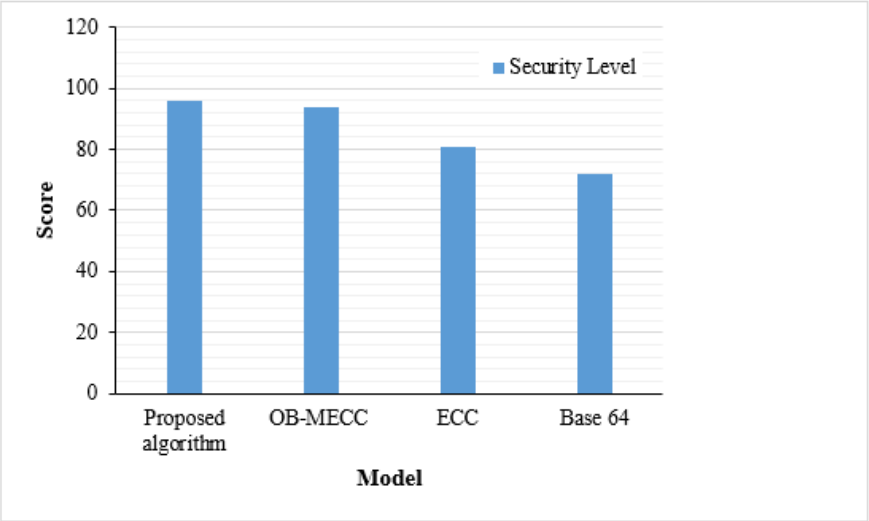
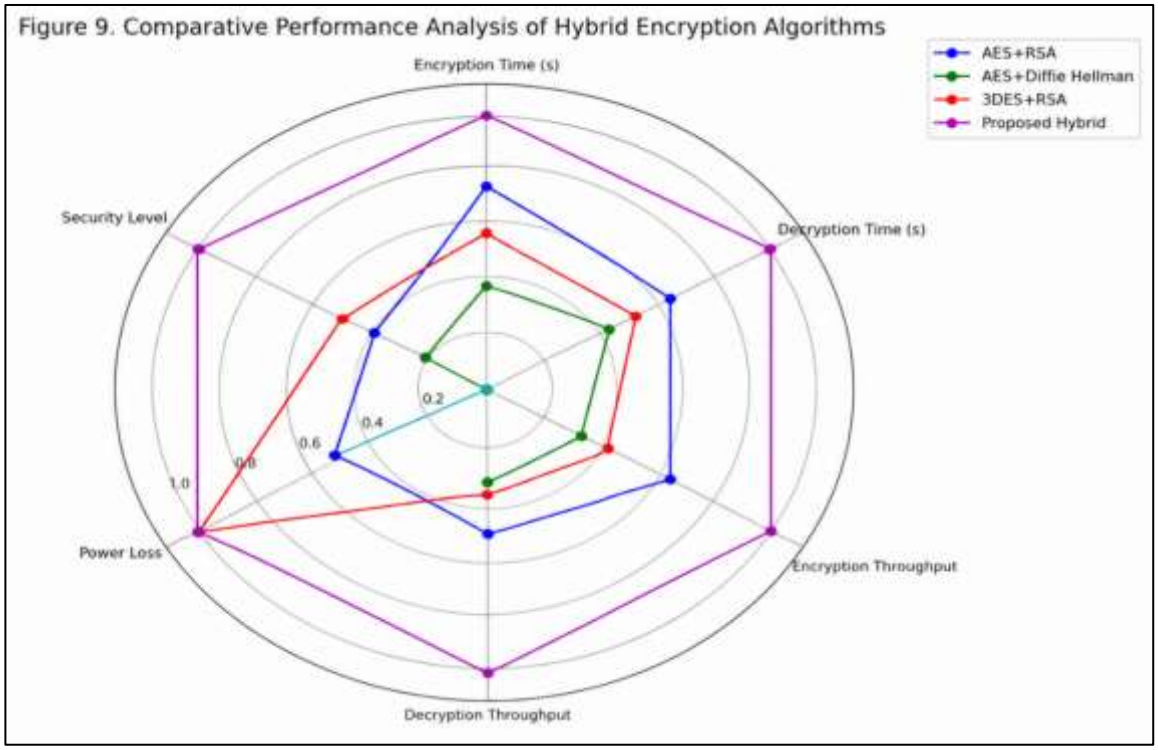


Figure 8. Security level of proposed algorithm

As shown in Table 5 and in Figure 8, hybrid encryption is the strongest security mechanism of all of the techniques that we compared. It achieves enhanced security by using multiple layers of protection including AES symmetric encryption, RSA secure key exchange, XOR masking, and other obfuscation methods. Unlike traditional methods that rely on a single cryptographic method, our hybrid method enhances the resistance to brute force, key disclosure, and man in the middle attacks by taking advantage of both the strong and weak aspects of symmetric and asymmetric encryption. The total security level (96) of our proposed hybrid method is higher than those of OB-MECC (94), ECC (81), or Base64 (72) demonstrating that our hybrid approach will provide better confidentiality and robustness when used to store and transmit data in the cloud. Therefore, our hybrid approach provides a good balance between security strength, computational efficiency, and energy use and makes it suitable for use in today's cloud computing environment.



A complete assessment of the proposed hybrid encryption algorithms is shown in figure 9 with respect to their execution times, throughput levels, power usage, and security levels. The newly developed hybrid framework shows an all-around superior level of performance because it offers lower times for both encryption and decryption, greater throughput, minimal lost power, and better overall security. Unlike traditional approaches that focus on either securing data or improving computational efficiency, our proposed framework provides a balanced compromise between performance and resiliency. Therefore, results indicate that our framework is an appropriate solution for use in cloud computing with the needs to provide secure, energy efficient, and fast transmission of data.

6. CONCLUSION AND FUTURE WORK

This research proposed a unique Encryption scheme that uses the AES/RSA/XOR masking for the design of a hybrid encryption framework to increase both the security and efficiency of data stored in the cloud. The results obtained from experiments with the proposed framework showed the fastest time to Encrypt data was 2.09 sec., Decrypt data was 2.45 sec., and Thruput achieved values were 0.0196, 0.0167, respectively, representing the lowest time for encryption, decryption, and highest thruput values compared to the next best competitor (AES + RSA, AES + Diffie-Hellman, TwoFish + RSA, OB-MECC) in all areas of performance. The results of the framework demonstrated that it will have a minimum power loss of 5.56 and the maximum level of security of 96. Additionally, the superior performance of the proposed framework is due to effective use of AES for Encrypting schema for high speed, using RSA for securely exchanging keys, and utilizing XOR as an additional key protection layer with a minimal processing burden. As a result of these efficiencies the proposed framework is ideally suited to provide secure storage of data in the cloud, as well as support IoT and any energy-sourcing applications that require high levels of security with efficient utilization of resources. Future research will focus on using the proposed hybrid encryption framework across real-time environments of the cloud that will include assessment/growth based on larger data sets, and utilizing different types of network conditions. Future enhancements may also support the integration of block chain technology/distributed trust management, the use of lightweight cryptographic technology for IoT devices and machine learning based intrusion detection techniques to provide adaptive/intelligent security against the increasing threat of Cyber Crimes.

REFERENCES

[1].Pamarthi, Nagaraju, and N. Nagamalleswara Rao. "Exponential Ant-Lion Rider Optimization for Privacy Preservation in Cloud Computing." In Web Intelligence, vol. 19, no. 4, pp. 275-293. Sage UK: London, England: SAGE Publications, 2021.

[2].Rong, Chunming, Son T. Nguyen, and Martin Gilje Jaatun. "Beyond lightning: A survey on security challenges in cloud computing." Computers & Electrical Engineering 39, no. 1 (2013): 47-54.

[3].Rao, R. Velumadhava, and K. Selvamani. "Data security challenges and its solutions in cloud computing." Procedia Computer Science 48 (2015): 204-209.

[4].Chen, Deyan, and Hong Zhao. "Data security and privacy protection issues in cloud computing." In 2012 international conference on computer science and electronics engineering, vol. 1, pp. 647-651. IEEE, 2012.

[5].Mogarala, Aruna Guruvaya, and K. G. Mohan. "Security and privacy designs based data encryption in cloud storage and challenges: A review." In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-7. IEEE, 2018.

[6].Kashyap, Gautam Siddharth, Prerna Kamani, Malvika Kanojia, Samar Wazir, Karan Malik, Vinay Kumar Sehgal, and Rajkumar Dhakar. "Revolutionizing agricultureA comprehensive review of artificial intelligence techniques in farming." Research Advances in Intelligent Computing (2024): 82-112.

[7].Wazir, Samar, Gautam Siddharth Kashyap, and Parag Saxena. "Mlops: A review." arXiv preprint arXiv:2308.10908 (2023).

[8].HaBib, Honey, Gautam Siddharth Kashyap, Nazia Tabassum, and Nafis Tabrez. "Stock price prediction using artificial intelligence based on LSTM–deep learning model." In Artificial Intelligence & Blockchain in Cyber Physical Systems, pp. 93-99. CRC Press, 2023.

[9].Kashyap, Gautam Siddharth, Karan Malik, Samar Wazir, and Rijwan Khan. "Using machine learning to quantify the multimedia risk due to fuzzing." Multimedia Tools and Applications 81, no. 25 (2022): 36685-36698.

[10]. Kashyap, Gautam Siddharth, Jatin Sohlot, Ayesha Siddiqui, Ramsha Siddiqui, Karan Malik, Samar Wazir, and Alexander EI Brownlee. "Detection of a facemask in realtime using deep learning methods: Prevention of COVID-19." In Research Advances in Network Technologies, pp. 224-240. CRC Press, 2024.

[11]. Alharbi, Fares, and Gautam Siddharth Kashyap. "Empowering Network Security through Advanced Analysis of Malware Samples: Leveraging System Metrics and Network Log Data for Informed Decision-Making." International Journal of Networked and Distributed Computing 12, no. 2 (2024): 250-264.

[12]. Tahir, Muhammad, Muhammad Sardaraz, Zahid Mehmood, and Shakoor Muhammad. "CryptoGA: a cryptosystem based on genetic algorithm for cloud data security." Cluster Computing 24, no. 2 (2021): 739-752.

[13]. Sun, Pan Jun. "Privacy protection and data security in cloud computing: a survey, challenges, and solutions." Ieee Access 7 (2019): 147420-147452.

[14]. Bahrami, Mehdi, and Mukesh Singhal. "CloudPDB: A light-weight data privacy schema for cloud-based databases." In 2016 International Conference on Computing, Networking and Communications (ICNC), pp. 1-5. IEEE, 2016.

[15]. Ko, Ryan KL, Peter Jagadpramana, Miranda Mowbray, Siani Pearson, Markus Kirchberg, Qianhui Liang, and Bu Sung Lee. "TrustCloud: A framework for accountability and trust in cloud computing." In 2011 IEEE World Congress on Services, pp. 584-588. IEEE, 2011.

[16]. Teneyuca, David. "Internet cloud security: The illusion of inclusion." Information Security Technical Report 16, no. 3-4 (2011): 102-107.

[17]. Shakir, Mohanaad, Maytham Hammood, and Ahmed Kh Muttar. "Literature review of security issues in saas for public cloud computing: a meta-analysis." International Journal of Engineering & Technology 7, no. 3 (2018): 1161-1171.

[18]. Siyal, Reshma, Muhammad Asim, Long Jun, Mohammed Elaffendi, Sundas Iftikhar, Rana Alnashwan, and Samia Allaoua Chelloug. "Adaptive Attribute-Based Honey Encryption: A Novel Solution for Cloud Data Security." Computers, Materials & Continua 82, no. 2 (2025).

[19]. Ali, Salman, and Faisal Anwer. "An IoT-Enabled Cloud Computing Model for Authentication and Data Confidentiality using Lightweight Cryptography." Arabian Journal for Science and Engineering (2025): 1-23.
[20]. Gaitond, Rekha, Gangadhar S. Biradar, and Sujata Terdal. "Blockchain-integrated optimized cryptographic framework for securing cloud data." Knowledge-Based Systems 324 (2025): 113830.
[21]. Gupta, Megha, Sakshi Sobti, Kshipra Jain, N. Gobi, Nitish Vashisht, and Zahid Ahmed. "Design of cloud-based security model with multiple CSPs." International Journal of System Assurance Engineering and Management (2025): 1-12.
[22]. Pandey, Kartikey, and Deepmala Sharma. "Novel image encryption algorithm utilizing hybrid chaotic maps and Elliptic Curve Cryptography with genetic algorithm." Journal of Information Security and Applications 89 (2025): 103995.
[23]. Shahid, Usman, Shamsa Kanwal, Mahwish Bano, Saba Inam, Manal Elzain Mohamed Abdalla, and Zaffar Ahmed Shaikh. "Blockchain driven medical image encryption employing chaotic tent map in cloud computing." Scientific Reports 15, no. 1 (2025): 6236.
[24]. Singamaneni, Kranthi Kumar, Anil Kumar Budati, and Thulasi Bikku. "An efficient Q-KPABE framework to enhance cloud-based IoT security and privacy." Wireless Personal Communications (2024): 1-29.
[25]. Alemami, Yahia, Ali M. Al-Ghonmein, Khaldun G. Al-Moghrabi, and Mohamad Afendee Mohamed. "Cloud data security and various cryptographic algorithms." International Journal of Electrical and Computer Engineering 13, no. 2 (2023): 1867.
[26]. Priyadarshini, R., Abdul Quadir Md, N. Rajendran, V. Neelanarayanan, and H. Sabireen. "An enhanced encryption-based security framework in the CPS Cloud." Journal of Cloud Computing 11, no. 1 (2022): 64.
[27]. Garg, Deepak, Shalli Rani, Norbert Herencsar, Sahil Verma, Marcin Wozniak, and Muhammad Fazal Ijaz. "Hybrid technique for cyber-physical security in cloud-based smart industries." Sensors 22, no. 12 (2022): 4630.
[28]. Thillaiarasu, N., S. Chenthur Pandian, and Naveenbalaji Gowthaman. "Novel heuristic scheme to enforce safety and confidentiality using feature-based encryption in multi-cloud environment (MCE)." In Information and Knowledge in Internet of Things, pp. 441-456. Cham: Springer International Publishing, 2021.
[29]. He, Qian, and Hong He. "A novel method to enhance sustainable systems security in cloud computing based on the combination of encryption and data mining." Sustainability 13, no. 1 (2020): 101.
[30]. NagaRaju, Pamarthi, and N. Nagamalleswara Rao. "OB-MECC: An Efficient Confidentiality and Security Enhancement for Cloud Storage System." Journal of Cyber Security and Mobility (2020): 577-600
[31]. Paramarthi, Nagaraju, Nagaraju Pamarthi, and Nagamalleswara Rao. "A Data Obfuscation Method Using Ant-Lion-Rider Optimization for Privacy Preservation in the Cloud." International Journal of Distributed Systems and Technologies (IJDST) 13, no. 5 (2022): 1-21.