Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

Enhancing Data Security in WLAN Networks Using a Hybrid Cryptographic System

Bushra Jaber M.Jawad[1]*

[1] University of Kerbala , Karbala, Iraq
*Corresponding author E-mail: bushra.j@uokerbala.edu.iq

https://doi.org/10.46649/fjiece.v4.2.16a.26.9.2025

Abstract. *Wireless Local Area Networks (WLANs) are one of the fastest growing technologies today. In recent years, WLANs have become one of the most adopted technologies. Nevertheless, wireless communications presents a huge security risk. As WLAN is susceptible to many attacks, encryption methods are the building block of WLAN security. However, it can be difficult to develop a practical system that provides high security but also has decent performance. The research proposes a new solution for a dual hybrid encryption method namely Advanced Encryption Standard with Elliptic Curve Cryptography (AES–ECC) and Advanced Encryption Standard with Rivest–Shamir–Adleman (AES–RSA). The paper compares two hybrid approaches to develop a unified framework. So, the key novelty is to combine the two approaches under the same framework. In the method proposed, symmetric encryption (AES) secures the sent data while asymmetric encryption (ECC or RSA) encrypts and securely sent an important secret key. The main contribution of this paper is to offers equal characterization of AES–ECC and the AES–RSA hybrids in WLAN environments as well as to provide an unbiased comparison of their results and security, resources usage and processing time. Experimental testing shows how effective and practical this framework is for high-level security and performance of WLAN digital communications.*

Keywords: *WLAN, Symmetric encryption, Asymmetric encryption, Hybrid cryptography, AES algorithm, REA algorithm, ECC algorithm, Throughput.*

1. INTRODUCTION

A Wireless Local Area Network or WLAN refers to a radio-based technology employed to expand wired LANs to ensure mobility and convenient connection. An access point (AP) allows modems or PCs with wireless adapters to join the network, just like Ethernet for wired networks. WLAN functions based on IEEE 802.11 standards and is commonly referred to as Wi-Fi. WLAN is better in terms of performance compared to other wireless technologies and also has many advantages such as reduced cost, quicker installation, and potentially extensive coverage for offices, homes, and campuses [1], [2], [3]. WLAN structure under IEEE specifications comprises more than one station (device), Base Service Sets (BSS), Extended Service Sets (ESS), and Distribution Services (DS) linking various ESSs. WLANs facilitate two straightforward modes of linking: infrastructure mode, where the devices are linked to the network using a central AP such as a router to link to the Internet; and ad hoc mode, in which devices are linked peer-to-peer with no base station. Such features render WLAN an economical, ubiquitous, and dynamic solution to the communications of the day [4]. Data can be transmitted wirelessly on shared channels through WLAN technology, and therefore vulnerable to all types of security attacks like eavesdropping and unauthorized usage. Therefore, it is essential to know the operation of symmetric and asymmetric encryption methods for WLAN security.Symmetric and asymmetric methods have been widely categorized based on various cryptographic algorithms. Symmetric approachs normally support bulk

encryption of data in an efficient manner. An example of a symmetric scheme is the advanced encryption standard (AES). They differ from asymmetric schemes in that they have shorter key sizes. For example, elliptic-curve (ECC) offers secure key exchange and digital signatures [5], [6], [7], [8], [9]. To improve symmetric encryption, various operational sequence methods or block cipher modes exist. These include electronic block cipher mode (ECB), cipher block chaining (CBC), plaintext cipher-block chaining, ciphertext feedback (CFB), output feedback (OFB), and counter mode (CTR). These methods differ in the levels of security and performance characteristics that is offers [10], [11], [12]. CBC is a method of encryption that adds randomness to the process using an initialization vector. This random element makes it hard to predict how notable encryption will perform. Additionally, it helps increase CBC's resistance to common types of attacks. However, one limitation of CBC is that it cannot encrypt in parallel. In combination, these algorithms and modes of operation constitute the practice of ensuring data protection, i.e. the control of confidentiality, integrity and trust in the system itself [13], [14].

AES, RSA and ECC denote the underlying types of an encryption algorithm where AES and RSA, together with ECC, provide overlapping and complementary security services. AES is a powerful symmetric algorithm that is fast and secure for the encryption of bulk data. It encrypts data in blocks of 128 bits. The key size can be 128, 192, or 256 bits and the algorithm will perform 10, 12, or 14 rounds respectively for each of the key size [15]. Specifically in speed, AES excels in performance. It is well-designed for simple and speedy encryption [16], while still demonstrating appeal for most applications. The speed of AES real-time use can also be substantially increased through parallel processing while maintaining significant encryption security [17], [18], [19], [20]. In contrast, RSA works on an asymmetric public-private key pair [21], [22]. RSA is a significant security protocol to provide secure communications and shared digital signature features, and allows for security when both parties use RSA encryption, by using the difficulty to factor large integers to provide this security guarantee. RSA is long-running and should generally be thought of as slow relative to symmetric encryption, like AES, particularly in key generation and encryption/decryption processes [23], [24]. ECC also uses asymmetric principles but relies on the mathematical properties of elliptic curves to generate smaller, highly efficient keys. ECC achieves equivalent security to much larger RSA keys, making it particularly suitable for mobile devices and resource-constrained environments. Its efficiency and versatility allow ECC to be combined with other cryptographic protocols, providing a flexible solution for modern secure communications [25], [26], [27]. To conclude, AES should continue to be considered as the questionable leading standard of the multiple forms of data encryption out there due to being able to balance both strong security with efficient speed against many forms of attacks, but continued research on methods of improving AES and data encryption in general is greatly warranted in the face of other processes and technology developing and improving. Asymmetric encryption is essential for secure communications but faces challenges such as high computational demands for key generation and encryption, complex key management, and potential vulnerabilities to attacks like side-channel or chosen-ciphertext attacks. These factors can impact performance and security if not properly managed [28], [29],[30].To overcome some of these challenges, this research proposes a dual framework based on two hybrid encryption schemes: AES-RSA and AES-ECC, to achieve a balance between efficiency and security.

The purpose of this work was to explore mechanisms to improve data protection while still achieving high performance in WLAN environments for sensitive applications and systems.

The structure of the research is organized in this way: Section two engages related work, specifically hybrid systems from earlier work; Section three offers the proposed framework, along with component descriptions, along with the hybrid mechanism for AES-RSA and AES-ECC; Section four explains implementations for the promoted framework, with instances of the promotion and some of the metrics considered.

Section five presents the discussion of results through performance and with multiple graphs for reporting on necessary comparisons, and section six concludes with 'Conclusion and Future Work' to summarize major accomplishments and considerations for the future.

2. RELATED WORK

2.1. Hybrid AES-RSA

The hybrid AES-RSA method put forward includes AES, which is a symmetric block cipher that encrypts the data with a 128-bit key, and RSA, which encrypts the AES key using a 2048-bit public key, which makes it more secure than traditional RSA only encryption [31]. The hybrid AES-RSA method will be applied in cloud computing and data security and will utilize the RSA and AES algorithms. The hybrid method will encrypt each byte and secure it using both algorithms. All of this will result in the hybrid ciphertext being secure, meaning that it can be decrypted to recover the original data [32]. Also in [33], combines the strengths of both AES and RSA to enhance file encryption efficiency and security. This approach addresses the limitations of using each algorithm separately, providing improved performance and robust data protection. The hybrid AES-RSA strategy leverages the security of the RSA algorithm with the rapid speed of AES algorithm to speed up the efficiency of data encryption or decryption. While in [34], method incorporated is presented, which more security and efficiency in the routing and also address key management problems RSA and AES based hybrid encryption technique for enhancing data security in cloud computing.

2.2 Hybrid AES-ECC

An enhanced AES-ECC model with key dependent dynamic S-Box for the security of mobile applications using cloud computing[35]. Hybrid AES-ECC model for the security of data over cloud storage. This paper proposes scheme for sharing data while maintaining data security and integrity over the cloud. [36]. Web applications want low processing overhead so that computation happens at the fastest speeds possible. In our existing hybrid model the AES algorithm is substituted with Blowfish security algorithm. The ECC algorithm could provide almost identical security as RSA with a much larger modulus and a similarly sized key. The MD5 algorithm is employed to verify data originality and a modified Kerberos protocol is invoked for client authentication [37]. Increasing security while reducing latency and resource consumption in FPGA implementations, the authors proposed hybrid AES-ECC cryptosystem combines AES, which offers high speed encryption of plaintext and ECC, which offers secure key distribution and decryption in [38]. The hybrid AES-ECC cryptosystem uses a co-design approach and rely on AES-ECC optimizations. We basically combine the matrix multiplication of AES MixColumns with the S-box for very fast software implementation (on NIOS-II processor). The authors also proposed an optimized ECC hardware architecture based on López-Dahab scalar multiplication (on Cyclone IV.E.). In [39], a manner of cloud data exchange is suggested. The expected improvement of the method has been concentrated on the security of the cloud computing platform. A hybrid Modified Elliptic Curve Cryptography (MECC) method was tested against the Advanced Encryption Standard (AES) method, in conjunction, will maximize the security of a cloud data exchange [40].

Although hybrid AES–RSA and AES–ECC schemes have been explored in earlier research, the novelty of our approach lies in the adaptive dual-hybrid model that enables choosing either AES–RSA or AES–ECC based on the computational environment and device performance. This flexibility enhances the system’s practicality and efficiency across various network conditions and resource-limited devices, which has not been covered in previous hybrid encryption studies frameworks.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

3. METHODOLOGY

Due to the threats facing WLAN networks and the nature of the environment, which is vulnerable to many attacks, especially with regard to weak encryption technologies that affect the security of sensitive data transmission. In this research, DUAL-hybrid cryptosystem framework incorporates the Advanced Encryption Standard (AES) algorithm, along with Elliptic Curve Cryptography (ECC) and Rivest-Shamir-Adleman (RSA), to take the advantages of both symmetric and asymmetric encryption while mitigating their disadvantages. AES can encrypt large data files efficiently, whereas RSA and ECC can potentially resolve difficulties with key distribution by using a public-private key structure. The proposed DUAL- hybrid framework including two processes, in the first process: The plaintext is being split into blocks and encrypted with AES algorithm while the symmetric key protected with ECC. A successful attacker is challenged with decryption of ciphertext without the key because the ECC protects the symmetric key. The second process: uses an RSA algorithm instead of ECC which has fast encryption rates as well as powerful resistance to several attacks.

The two processes, being implement in our proposed framework, represent the goal of being efficient and the other with owing to security. The DUAL-hybrid cryptosystem framework improves evidence of processing performance as well as evidencing the integrity of the document.

Elliptic Curve Cryptography (ECC) uses key generation methods that are important for secure communication. Key generation usually involves creating a public-private key pair using elliptic curves over finite fields. A random integer is selected as a private key, and the private key is multiplied against a point on the curve to get the public key. This method ensures not only computationally efficient, but it uses sufficient security.

With WLAN network vulnerabilities and environment exposure, particularly the use of weak encryption technologies that undermine the protection of sensitive data from data-in-transit attacks, this research presents the DUAL-hybrid cryptosystem model that integrates the Advanced Encryption Standard (AES), Elliptic Curve Cryptography (ECC), and Rivest Shamir Adleman (RSA) encryption. The model exploits the capabilities of symmetric and asymmetric encryption and makes up for their limitations. AES works effectively to safeguard huge data files during the encryption process, and RSA and ECC play vital roles when it comes to key distribution since public/private key infrastructures can adequately cater to an application's needs. There are two discretely deployed processes in the framework. In the first, plaintext is segmented into blocks and encrypted with AES and the symmetric key safeguarded using ECC. An attacker attempting the decryption process will be confronted with the limitation of decrypting the ciphertext without the key, since ECC is securing the symmetric key. In the second process, RSA is used in place of ECC as a hybrid encryption method due to the speed of encryption and strong potential resistance applied to plaintext during decryption of various types of attacks the ciphertext potentially would be exposed to during that time. The framework employs these two processes to secure the data and improve safety. The DUAL-hybrid cryptosystem enhances processing efficiencies and ensures that the information is secure when it is transmitted. The procedure for Elliptic Curve Cryptography (ECC) generally consists of generating the key pairs that are the basis for assuring security user to user. In generality, this dynamic is composed of generating a public and private key pair using elliptic curves over finite fields. The private key is then randomly chosen as an integer value and, the private key is multiplied by a point on the elliptic curve, creating the public key. All in all, it is a high computationally efficient, and while optimizing security.

The important part is that many cryptographic protocols allow batch processing for key generation and can be performed in coordination with other protocols or within the same environment. All of these

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

examples demonstrate the ability of ECC to be responsive to modern computer processing speed and capabilities, while preserving the best elements of security we find in ECC today.

AES key encryption using ECC or RSA algorithms to take advantage of methods for enhanced security and performance. The procedure starts with generating a secure, random 128-bit secret key for the AES algorithm. Secure random key generation is inherent to the safety of plain text data because the key must be exchanged securely to prevent any eavesdropping. ECC protects the AES key during transmission by using a public key encryption scheme. When the sender transmits the AES secret key to the recipient, they encrypt it with the recipient's public ECC key, forming a ciphertext encrypted with ECC that only the recipient can decrypt using their private key. This is less cumbersome to assign keys because interception or theft of only the AES ciphertext does not allow the bad actor to use that key without either the private ECC key or the private RSA secret key.

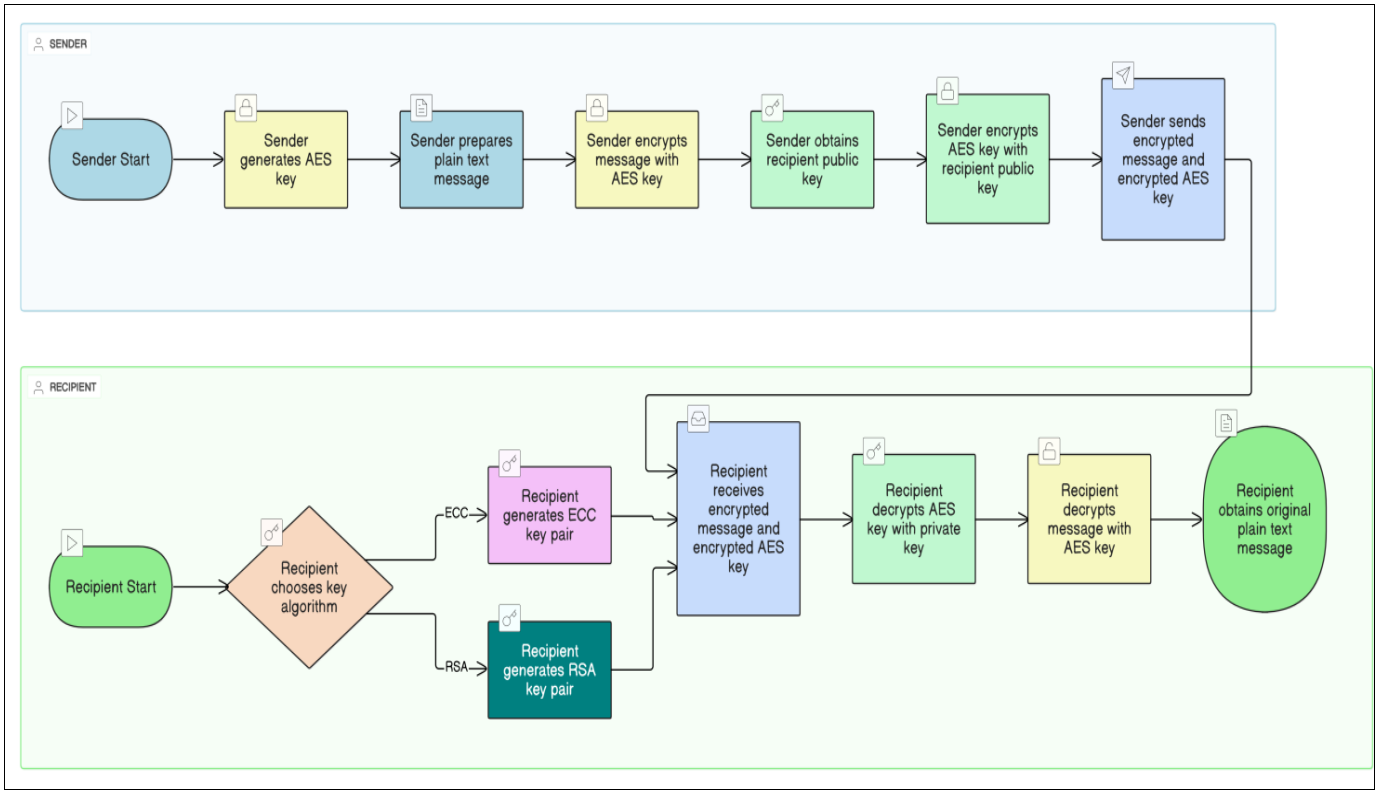

Fig. 1.: Flow Diagram of the Proposed Dual-Hybrid Cryptography Framework

In this method, both parties, the sender and the recipient, will create their own separate ECC and RSA key pairs first, and then the sender will create the AES secret key. After the AES secret key is created, the sender will encrypt the AES secret key using the recipient's public ECC and RSA key along with any encrypted communications. The recipient's public key will authenticate the communication by being in the encrypted data. The recipient can decrypt using their private ECC and RSA key to obtain the original AES secret key, which can now be used to decrypt any data encrypted with that generated key. The encryption and decryption pseudocodes of the proposed work are explained in Figures 2 and 3.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

```
Algorithm Hybrid_Encrypt(Plaintext, SymmetricKey, PublicKey)
Input: Plaintext message, SymmetricKey (AES key), PublicKey (RSA or ECC)
Output: Ciphertext, EncryptedKey

1. // Step 1: Encrypt the data with AES
2. Ciphertext ← AES_Encrypt(Plaintext, SymmetricKey)

3. // Step 2: Encrypt the AES key using Asymmetric encryption
4. EncryptedKey ← Asymmetric_Encrypt(SymmetricKey, PublicKey)

5. // Step 3: Combine the results for transmission
6. Output ← (Ciphertext, EncryptedKey)

7. Return Output
```

Fig. 2. Pseudocode of Encryption Process of Hybrid System

AES_Encrypt and AES_Decrypt: use AES in a chosen mode (e.g., CBC, CFB, CTR). In our experimental setup used CBC mode. Whereas Asymmetric_Encrypt and Asymmetric_Decrypt can be RSA (encrypt/decrypt with public/private key) or ECC. The dual framework means you run the above process twice: once with AES–RSA or once with AES–ECC. Then compare performance and security results.

```
Algorithm Hybrid_Decrypt(Ciphertext, EncryptedKey, PrivateKey)
Input: Ciphertext, EncryptedKey, PrivateKey (RSA or ECC)
Output: Plaintext

1. // Step 1: Use the private key to recover the AES key
2. SymmetricKey ← Asymmetric_Decrypt(EncryptedKey, PrivateKey)

3. // Step 2: Decrypt the data using the recovered AES key
4. Plaintext ← AES_Decrypt(Ciphertext, SymmetricKey)

5. Return Plaintext
```

Fig. 3. Pseudocode of Decryption Process of Hybrid System

Lastly, here is a brief description of the proposed system:

- Allows a dynamic adaptability feature that gives the system the option of choosing between AES-RSA and AES-ECC given the performance of the device and the network state.
- Improved performance and flexibility particularly in heterogeneous environments (e.g., IoT, WLAN, cloud, etc.)
- Provides a complete evaluating of the two encryption/decryption times, memory usage, throughput, and latency for both dual models.
- Comprises energy-awareness and scalability to aid in enabling our hybrid framework to be more practical.
- Our proposed framework has a clear contribution in which integrate AES with RSA or ECC; and manage trade-offs between security strength and performance based on system restrictions.

4. EXPERIMENTAL SETUP

This experimentation focuses on examining AES key generation enhancement, including key hardware and software specifications and metrics for evaluation of improved performance.

4.1. EXPERIMENTAL ENVIRONMENT

In the experimental environment working towards AES key generation performance improvement, hardware and software will be a consideration for appropriate performance evaluation. An experimental environment can be heavily reliant on processing power and a variety of resources from random-access memory (RAM) to utilize implemented modified AES algorithms to improve key generation.

The choice of software tools greatly affects the evaluation process as well. The main an Integrated Development Environment (IDE) specifically designed for Python programming is Pycharm, which allows efficient synthesis, implementation and simulation of the developed algorithms. The testing framework includes both proprietary solutions and open source libraries to measure performance effectively. Hardware selection also significantly affects outcomes; The proposed method is implemented on a laptop with an Intel i7 core processor and 16GB RAM, operating system is Windows 11Pro.

4.2. EVALUATION OF PERFORMANCE MEASUREMENTS

When evaluating the efficiency of proposed framework of hybrid cryptographic systems. The performance metrics are: the encryption and decryption speed, the time required to convert plain text to cipher text, memory usage represents the amount of RAM is currently using. It's a crucial metric for assessing performance, and CPU usage refers to the amount of processing power a computer's CPU is consumption at any given time. The following steps is to calculate the selected metrics.

4.2.1 CALCULATION STEPS OF ENCRYPTION/DECRYPTION TIME

This measures the time taken to perform encryption or decryption on a given dataset.

1. Start a timer right before starting the encryption or decryption operation.
2. Execute the encryption/decryption algorithm on the file or data block.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

3. Stop the timer immediately after the operation completes.
4. Compute the elapsed time:

$$\text{Time (seconds)} = \text{End time} - \text{Start time} \quad (1)$$

4.2.2 CALCULATION STEPS OF ENCRYPTION/DECRYPTION MEMORY USAGE

This metric measures how much memory (RAM) is consumed during encryption or decryption.

1. Record the memory usage before starting the encryption/decryption process.
2. Execute the encryption/decryption algorithm.
3. Record the memory usage after completion.
4. Compute the difference:

$$\text{Memory Usage (MB)} = \text{Memory after}_{\text{Enc/Dec}} - \text{Memory before}_{\text{Enc/Dec}} \quad (2)$$

4.2.3 CALCULATION STEPS OF ENCRYPTION/DECRYPTION THROUGHPUT

Throughput measures the amount of data processed per unit of time, usually in bytes per second (B/s) or megabytes per second (MB/s).

$$\text{Throughput} = \frac{\text{Total Encrypted Plaintext}}{\text{Encryption Time}} \quad (3)$$

5. RESULTS AND DISCUSSION

When evaluating the efficient of proposed framework of hybrid cryptographic systems. The performance metrics are: the encryption and decryption speed, the time required to convert plain text to cipher text, memory usage represents the amount of RAM is currently using. It's a crucial metric for assessing performance, and CPU usage refers to the amount of processing power a computer's CPU is consumption at any given time.

5.1 EXPERIMENTAL RESULTS OF AES AND HYBRID AES-RSA

Figure 4 shows the encryption time, memory usage, and throughput (y-axis) plotted against different file sizes (x-axis: 50 KB, 100 KB, 500 KB, 1000 KB, 5000 KB). The plot compares the pure AES algorithm with the hybrid AES-256-RSA approach.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

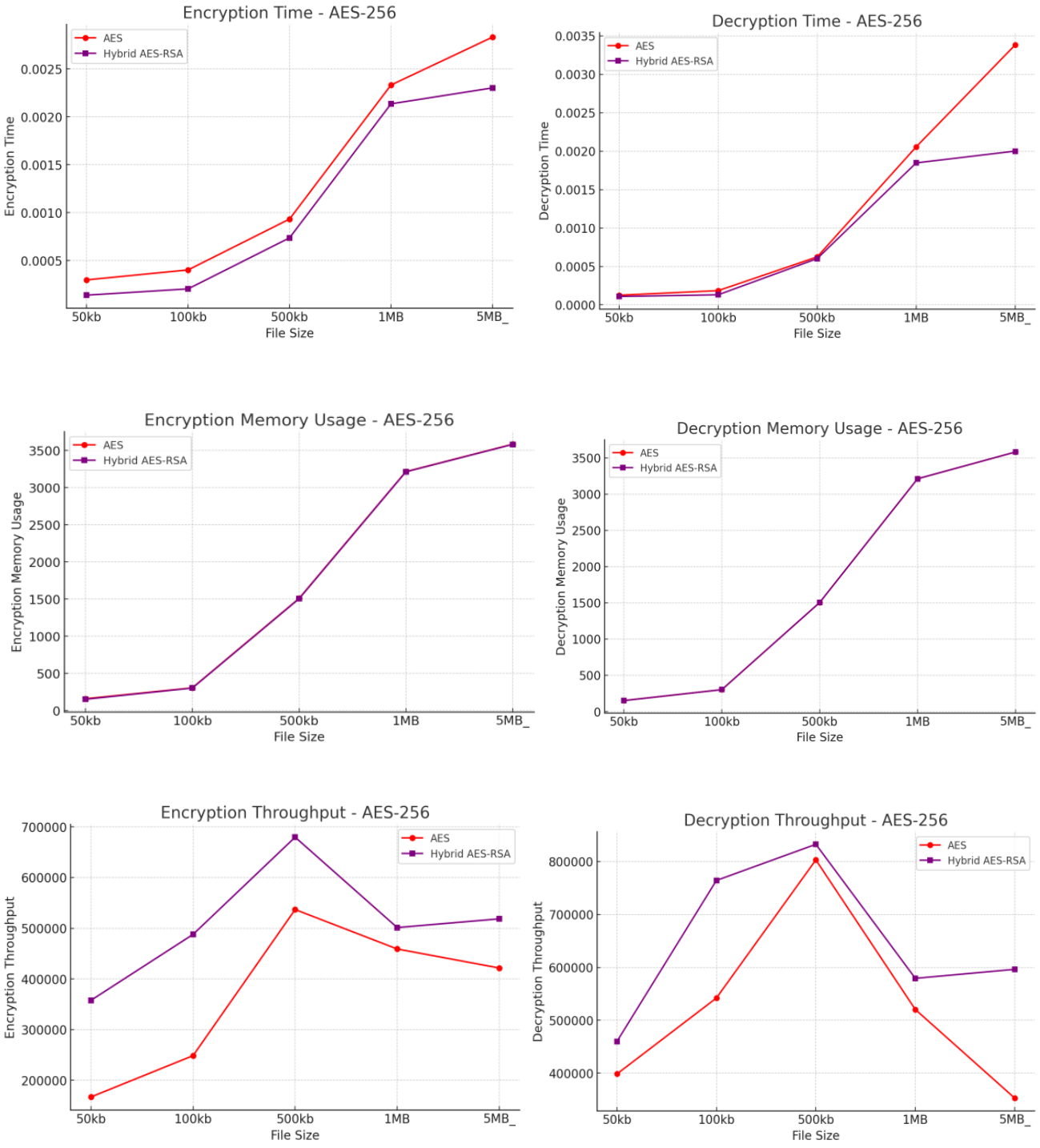

Fig. 4. Encryption and Decryption time, Memory Usage and Throughput of AES-256 key size and Hybrid AES-RSA

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

5.2 EXPERIMENTAL RESULTS OF AES AND HYBRID AES-ECC

Figure 5 shows the encryption time, memory usage, and throughput (y-axis) plotted against different file sizes (x-axis: 50 KB, 100 KB, 500 KB, 1000 KB, 5000 KB). The plot compares the pure AES algorithm with the hybrid AES-256-ECC approach.

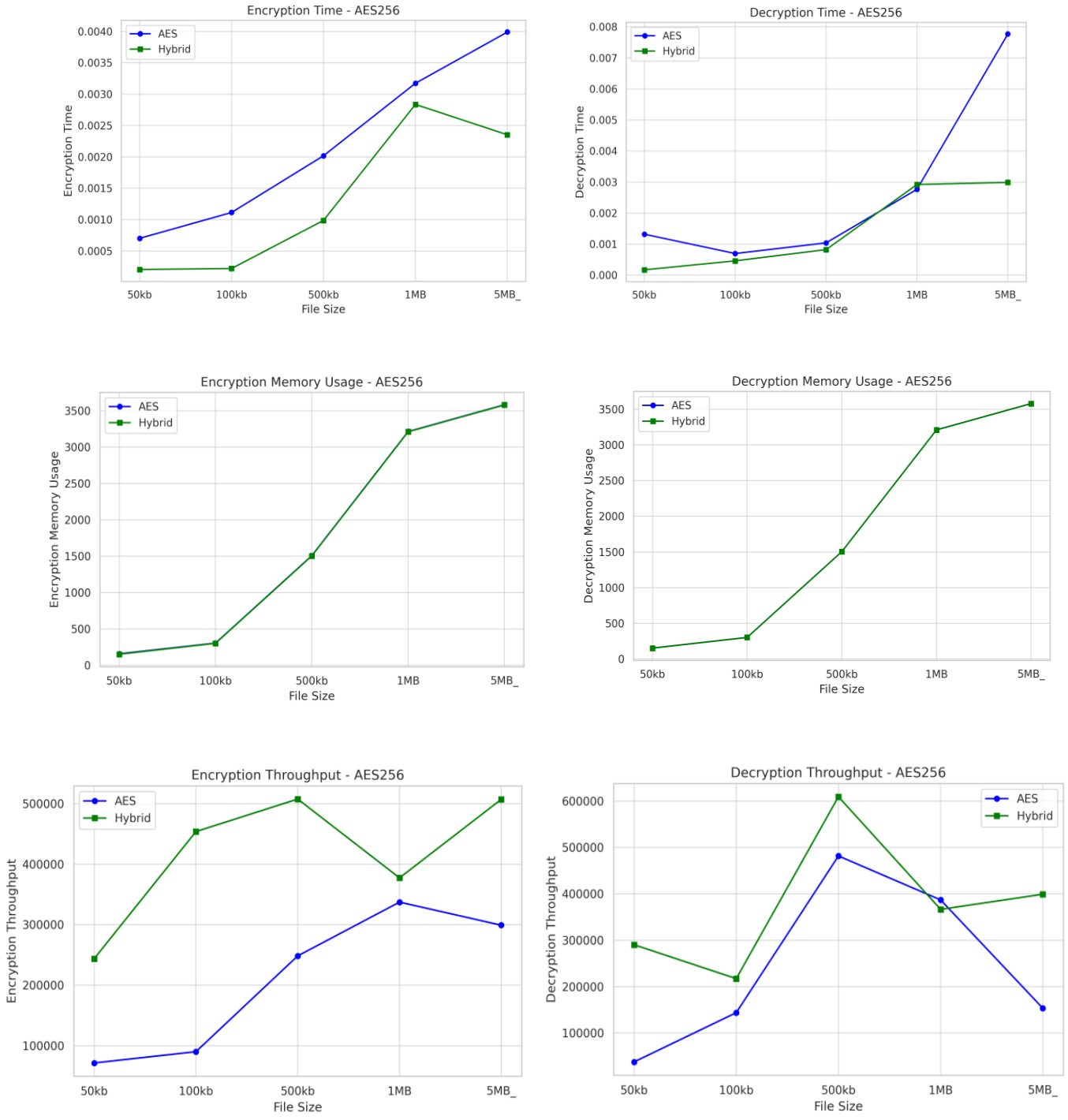

Fig. 5. Encryption and Decryption time, Memory Usage and Throughput of AES-256 key size and Hybrid AES-ECC

5.3 DISCUSSION OF EXPERIMENTAL RESULTS

As shown in Figure 4, the encryption speed of the hybrid AES256-RSA method is faster than that of AES alone for small files (100 KB and 500 KB), but the performance converges for larger files. The decryption speed of the hybrid AES256-RSA method is also faster than standalone AES for small files, with the difference decreasing as file size increases; at 5 MB, their speeds become equal. Regarding throughput, the hybrid method outperforms AES on 1, 2, and 3 MB files, with throughput converging slightly at 5 MB. In regards to memory consumption when encrypting, both methods use approximately the same amounts of memory. The decryption memory consumption shows a similar trend in which the hybrid method uses the same or a little less memory in every instance regardless of file size. Conclusion: The hybrid method demonstrates great encryption throughput performance (especially for small or medium sized files) since the main computational load falls on AES with just RSA used to encrypt the AES key. The hybrid method also shows good decryption throughput performance since only AES key decryption requires RSA instead of any data decryption, PCI does not encounter any major performance penalty. Analysis: even with RSA being included in the hybrid encryption, the only data item being encrypted is the AES key rather than the entire dataset. These findings helped or observed significant methods are hybrid AES-RSA method as rapidly gained AES-RSA information. Reason: The text was copied and rewritten, correcting grammar, punctuation and capitalization errors and enhancing clarity and flow review while improving technicality by specifying.

Analysing and discussing AES256 and the hybrid AES256-ECC based on the findings of Figure 5 is as follows: AES256 shows a markedly longer time to encrypt both small file sizes and larger file sizes, particularly for larger files (5MB), indicating it increases over time at a gradual rate. The hybrid approach shows a lower time to encrypt, showing a clear advantage over AES256, particularly with larger files, showing it has a better speed efficiency. In contrast, AES256 takes longer to decrypt, with the difference increasing in time particularly for the larger files (5MB). Conversely, the hybrid method has lower and more consistent time to decrypt as file sizes increase. In terms of throughput, the hybrid method significantly outperforms AES256, particularly with the medium-sized files (100KB-500KB). The hybrid method has better throughput compared with AES256 across all file sizes including the higher file sizes, with the most striking difference with 500KB files, where the difference in throughput was substantially higher than AES256. Conversely, AES256 shows significantly lower throughput than the hybrid method. Like the previous two scenarios, in terms of total memory used for encryption and decryption, we see that the memory consumption curves for AES256 and the hybrid method were almost identical across all file sizes, indicating there is no difference between the two methods in term of total memory consumption or the encryption or decryption process.

As file sizes increase, the key management efficiency of the hybrid system increases, because the static part of the process (key generation and exchange) is not affected by data size to the same extent as pure AES, which relies entirely on repeated symmetric encryption operations. Thus, for small files, the difference is minimal because the static overhead is similar. For larger files (such as 1MB to 5MB), the latency of the hybrid method is relatively lower due to the effort distribution between the two methods and the key processing efficiency of ECC. This behaviour demonstrates that the hybrid AES-ECC architecture can offer better time performance in practical large data encryption, achieving a balance between speed and security, making it a more efficient option than AES alone when processing increasing data volumes.

5.4 COMPARISON BETWEEN HYBRID AES-RSA AND HYBRID AES-ECC

In the following three figures, comparisons are presented to evaluate the two proposed hybrid methods for the performance measures chosen in this study.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

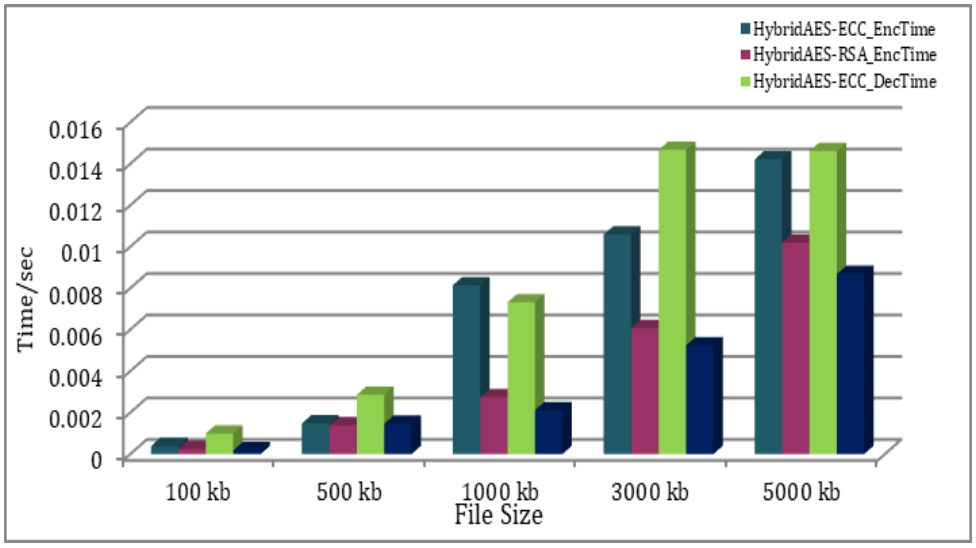

Fig. 6. Comparison between Hybrid AES-RSA and Hybrid AES-ECC Encryption/Decryption Time

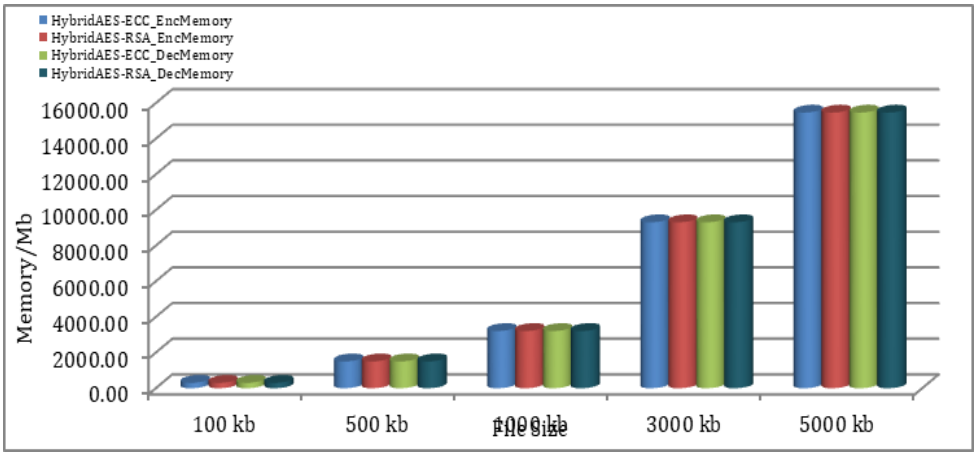

Fig. 7. Comparison between Hybrid AES-RSA and Hybrid AES-ECC Encryption/Decryption Memory Usage

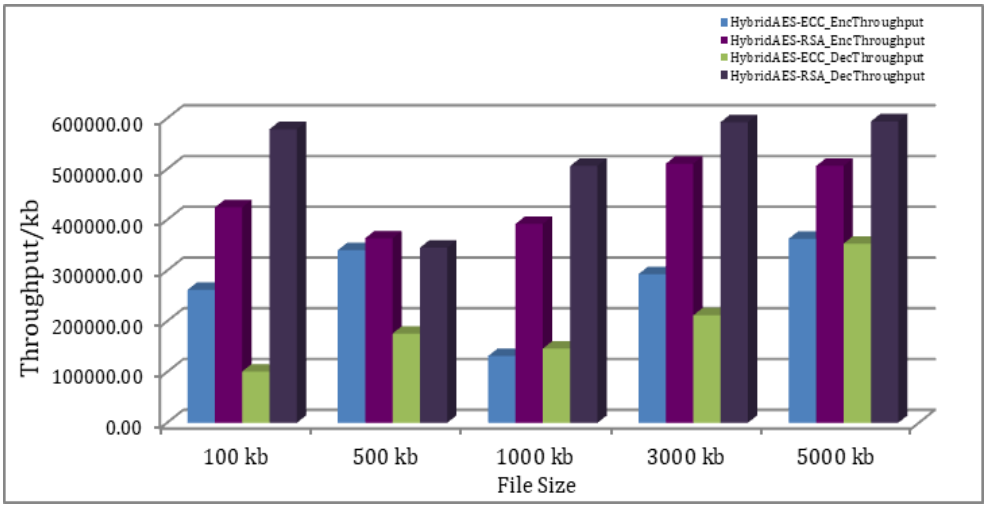

Fig. 8: Comparison between Hybrid AES-RSA and Hybrid AES-ECC Encryption/Decryption Throughput

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

5.5. DISCUSSION OF COMPARISON HYBRID AES-RSA AND HYBRID AES-ECC

In the Figures is a detailed analysis of each chart, with a brief conclusion, As in Figures 6,7,8:

• *Hybrid AES_RSA Encryption/Decryption Time:*

- ✓ The slowest time across all file sizes, indicating that encryption using RSA is faster in this architecture than ECC. Generally, the increase is gradual and slow as file size increases.
- ✓ The decryption time is higher than the encryption time, with greater acceleration for larger files, but less than ECC for decryption.

• *Hybrid AES_ECC Encryption/Decryption Time*

- ✓ There is better performance using this method, especially at decryption.
- ✓ Performance is relatively stable, but does fluctuate slightly, which indicates that is relative stability as file sizes increase.

Summary: ECC does get a higher throughput especially for decryption, but does not consistently and will change with file size.

• *Hybrid AES_RSA Encryption/Decryption Throughput*

- ✓ The throughput is lower in comparison to ECC. This is to be expected since throughput is determined by file size and time. Therefore a lower throughput is created the higher the throughput (in time), but certain thresholds, particularly a 500 KB file size, will approach ECC performance.

• *Hybrid AES_ECC Encryption/Decryption Throughput*

- ✓ There is better performance using this method, especially at decryption.
- ✓ Performance is relatively stable, but does fluctuate slightly, which indicates that is relative stability as file sizes increase.

Summary: ECC does get a higher throughput especially for decryption, but does not consistently and will change with file size.

• Hybrid AES_RSA Encryption/Decryption Memory Usage

- ✓ This method is less memory intensive than ECC, particularly for larger files. This is a reflection of the simplicity of RSA computational structure as compared to ECC in this scenario.
- ✓ Decryption memory is larger than encryption.

• *Hybrid AES_ECC Encryption/Decryption Memory Usage*

- ✓ The memory consumption is significantly higher for using ECC, with decryption consumption surpassing 6000 MB at 5000 KB file size.

- ✓ There is a sharp increase in memory consumption as file size increases, which becomes problematic for resource-poor devices. Ltd.

Summary: RSA is more memory efficient than ECC while ECC exhibits significantly more memory consumption, especially for decryption.

6. CONCLUSIONS AND FUTURE WORK

Enhancing the current encryption schemes deployed in WLANs is critical for wireless communications. The new proposed framework, dual-hybrid encryption, has exhibited suitable effectiveness. As we progress in technology and device capability, developing effective and secure encryption systems will be required to secure the networks from future adversaries and threats. The support of empirical evidence finds that a better rate performance shows the architecture of the underlying system can produce a better rate through hybrid encryption, particularly at small sizes; thus, hybrid encryption is successful. Memory consumption is less improved concerning the amount of methods producing similar memory utilization. Although the suggested dual AES-ECC and AES-RSA hybrid encryption frameworks demonstrate successful performance and security for WLAN scenarios, there are future avenues for work. For one, the frameworks could be extended to incorporate more lightweight asymmetric algorithms (and considerations for the performance and security with lattice-based or post-quantum cryptography). For another, more optimization with regards to the key distribution process demonstrating a better latency and reduced computational overhead while considering constrained devices (use cases with IoT nodes) must be explored. Finally, energy consumption analysis can be incorporated, since modern WLAN applications are often deployed in battery-powered systems where power efficiency is critical, including measure CPU load, memory, and power consumption for different key sizes and encryption modes.

REFERENCES

[1] R. Agrawal *et al.*, “Classification and comparison of ad hoc networks: A review,” *Egyptian Informatics Journal*, vol. 24, no. 1, pp. 1–25, 2023, doi: 10.1016/j.eij.2022.10.004.

[2] R. Krishan and V. Laxmi, “IEEE 802.11 WLAN Load Balancing for Network Performance Enhancement,” *Procedia Computer Science*, vol. 57, pp. 493–499, 2015, doi: 10.1016/j.procs.2015.07.371.

[3] M. Souppaya and K. Scarfone, “NIST Special Publication 800-153: Guidelines for Securing Wireless Local Area Networks (WLANs),” p. 17, 2012.

[4] M. R. Joshi and R. Avinash Karkade, “Network Security with Cryptography,” *International Journal of Computer Science and Mobile Computing*, vol. 4, no. 1, pp. 201–204, 2015.

[5] U. H. Shaikh, M. M. Abbas, S. A. Lahad, M. Razi, and M. Shaikh, “A Comparative Survey of Symmetric and Asymmetric Key Cryptography Algorithms,” *2nd International Multidisciplinary Conference on Emerging Trends in Engineering Technology-2024 (2nd IMCEET-2024)*, no. October, pp. 257–262, 2024.

[6] M. A. Al-Shabi, “A Survey on Symmetric and Asymmetric Cryptography Algorithms in information Security,” *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 3, p. p8779, 2019, doi: 10.29322/ijsrp.9.03.2019.p8779.

[7] D. J. George and T. Thomas, "A Comparative Study of Symmetric Key Algorithms," *International Journal of Computer Science and Mobile Computing*, vol. 12, no. 6, pp. 71–75, 2023, doi: 10.47760/ijcsmc.2023.v12i06.008.

[8] O. G. Abood and S. K. Guirguis, "A Survey on Cryptography Algorithms," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 8, no. 7, 2018, doi: 10.29322/ijsrp.8.7.2018.p7978.

[9] R. Bhanot and R. Hans, "A review and comparative analysis of various encryption algorithms," *International Journal of Security and its Applications*, vol. 9, no. 4, pp. 289–306, 2015, doi: 10.14257/ijsia.2015.9.4.27.

[10] D. Bujari and E. Aribas, "Comparative analysis of block cipher modes of operation," *International Advanced Researches & Engineering Congress*, no. November 2017, pp. 2–5, 2017.

[11] A. Fenyi, J. G. Davis, and K. Riverson, "Comparative Analysis of Advanced Encryption Standard , Blowfish and Rivest Cipher 4 Algorithms Abstract :," *International Journal of Innovative Research and Development*, vol. 3, no. 11, pp. 384–392, 2014.

[12] Bushra Jaber M.Jawad and S. Al-alak, "Design and Implementation of Multi-key Blowfish and CAST Algorithm: Comparative Study with CBC, CFB and CTR Modes," *Wasit Journal of Computer and Mathematics Science*, vol. 2, no. 4, pp. 87–98, 2023, doi: 10.31185/wjcms.203.

[13] M. Dworkin, "Recommendation for Block Cipher Modes of Operation," *National Institute of Standards and Technology Special Publication 800-38A 2001 ED*, vol. X, no. December, pp. 1–23, 2005.

[14] M. Dworkin, "Recommendation for block cipher modes of operation: three variants of ciphertext stealing for CBC mode," *National Inst of Standards and Technology Gaithersburg MD Computer security Div*, no. NIST-SP-800-38A, 2010.

[15] A. M. Abdullah, "Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data," *Cryptography and Network Security*, vol. 16, no. 1, p. 11, 2017.

[16] E. M. De Los Reyes, A. M. Sison, and R. P. Medina, "Modified AES cipher round and key schedule," *Indonesian Journal of Electrical Engineering and Informatics*, vol. 7, no. 1, pp. 28–35, 2019, doi: 10.11591/ijeei.v7i1.652.

[17] B. A. Buhari, A. A. Obiniyi, K. Sunday, and S. Shehu, "Performance Evaluation of Symmetric Data Encryption Algorithms: AES and Blowfish," *Saudi Journal of Engineering and Technology*, vol. 04, no. 10, pp. 407–414, 2019, doi: 10.36348/sjeat.2019.v04i10.002.

[18] K. Kumar, K. R. Ramkumar, and A. Kaur, "A lightweight AES algorithm implementation for encrypting voice messages using field programmable gate arrays," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 3878–3885, 2022, doi: 10.1016/j.jksuci.2020.08.005.

[19] H. Zodpe and A. Sapkal, "An efficient AES implementation using FPGA with enhanced security features," *Journal of King Saud University - Engineering Sciences*, vol. 32, no. 2, pp. 115–122, 2020, doi: 10.1016/j.jksues.2018.07.002.

[20] K. Assa-Agyei and F. Olajide, "A Comparative Study of Twofish, Blowfish, and Advanced Encryption Standard for Secured Data Transmission," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 3, pp. 393–398, 2023, doi: 10.14569/IJACSA.2023.0140344.

[21] N. A. Wahid, A. Ali, B. Esparham, and M. Marwan, "A Comparison of Cryptographic Algorithms:

DES, 3DES, AES, RSA and Blowfish for Guessing Attacks Prevention," *Journal of Computer Science Applications and Information Technology*, vol. 3, no. 2, pp. 1–7, 2018.

[22] P. Kuppuswamy, S. Q. Y. A. K. Al-Maliki, R. John, M. Haseebuddin, and A. A. S. Meeran, "A hybrid encryption system for communication and financial transactions using RSA and a novel symmetric key algorithm," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 2, pp. 1148–1158, 2023, doi: 10.11591/eei.v12i2.4967.

[23] W. Al-Nbhany and A. Zahary, "A Comparative Study among Cryptographic Algorithms: Blowfish, AES and RSA," *International Arab Conference on Information Technology*, no. May, 2016.

[24] K. Assa-Agyei and F. Olajide, "A Comprehensive Evaluation of the Rivest-Shamir-Adleman (RSA) Algorithm Performance on Operating Systems using Different Key Bit Sizes," *International Journal of Computer Applications*, vol. 185, no. 19, pp. 14–20, 2023, doi: 10.5120/ijca2023922884.

[25] P. Dixit, A. K. Gupta, M. C. Trivedi, and V. K. Yadav, "Traditional and hybrid encryption techniques: A survey," *Lecture Notes on Data Engineering and Communications Technologies*, vol. 4, no. January, pp. 239–248, 2018, doi: 10.1007/978-981-10-4600-1_22.

[26] D. Kumar Sharma, N. Chidananda Singh, D. A. Noola, A. Nirmal Doss, and J. Sivakumar, "A review on various cryptographic techniques & algorithms," *Materials Today: Proceedings*, vol. 51, no. xxxx, pp. 104–109, 2021, doi: 10.1016/j.matpr.2021.04.583.

[27] R. Yadav, "Analysis of Cryptography in Information Technology," *Interantional Journal of Scientific Research in Engineering and Management*, vol. 07, no. 03, pp. 1–6, 2023, doi: 10.55041/ijsrem18379.

[28] J. Zhang, X. Wang, B. Liu, J. Wang, X. Li, and L. He, "Research on symmetric encryption and decryption algorithm of shared data based on chaotic mapping and permutation," *2023 IEEE 6th International Conference on Information Systems and Computer Aided Education, ICISCAE 2023*, pp. 302–305, 2023, doi: 10.1109/ICISCAE59047.2023.10391859.

[29] R. Rizk and Y. Alkady, "Two-phase hybrid cryptography algorithm for wireless sensor networks," *Journal of Electrical Systems and Information Technology*, vol. 2, no. 3, pp. 296–313, 2015, doi: 10.1016/j.jesit.2015.11.005.

[30] P. Williams, I. K. Dutta, H. Daoud, and M. Bayoumi, "A survey on security in internet of things with a focus on the impact of emerging technologies," *Internet of Things (Netherlands)*, vol. 19, p. 100564, 2022, doi: 10.1016/j.iot.2022.100564.

[31] A. Srivastava, A. Kumar, and R. Article, "A Robust Approach to Secure Data Encryption: AES-RSA Hybrid with Kernel Key Protection," 2023.

[32] R. S. Durge and V. M. Deshmukh, "Securing Cloud Data: A hybrid encryption approach with RSA and AES for enhanced security and performance," Journal of Integrated Science and Technology, vol. 13, no. 3, pp. 1–7, 2025, doi: 10.62110/SCIENCEIN.JIST.2025.V13.1060.

[33] kashif Iqbal, M. U. jatoi, M. Sulaman, and M. S. Abid, "Robust Multi-Party Computation in Critical Infrastructure Protection using Hybrid RSA-AES Algorithm for Enhanced Security," pp. 1–21, 2024.

[34] S. Urooj, S. Lata, S. Ahmad, S. Mehfuz, and S. Kalathil, "Cryptographic Data Security for Reliable Wireless Sensor Network," Alexandria Engineering Journal, vol. 72, pp. 37–50, 2023, doi: 10.1016/j.aej.2023.03.061.

[35] M. N. J. R. , Nagaraj M Lutimath, "An Enhanced AES-ECC model with Key Dependent Dynamic

S-Box for the Security of Mobile Applications using Cloud Computing," *Journal of Electrical Systems*, vol. 20, no. 2, pp. 2735–2746, 2024, doi: 10.52783/jes.2050.

[36] S. Rehman, N. Talat Bajwa, M. A. Shah, A. O. Aseeri, and A. Anjum, "Hybrid aes-ecc model for the security of data over cloud storage," *Electronics (Switzerland)*, vol. 10, no. 21, pp. 1–20, 2021, doi: 10.3390/electronics10212673.

[37] N. Gupta and V. Kapoor, "Hybrid cryptographic technique to secure data in web application," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 23, no. 1, pp. 125–135, 2020, doi: 10.1080/09720529.2020.1721872.

[38] J. Zhang, W. Gao, J. Li, X. Tian, and H. Dang, "High-Speed and High-Security Hybrid AES-ECC Cryptosystem Based on FPGA," *ICSIDP 2019 - IEEE International Conference on Signal, Information and Data Processing 2019*, 2019, doi: 10.1109/ICSIDP47821.2019.9173457.

[39] A. Hafsa, A. Sghaier, M. Zeghid, J. Malek, and M. Machhout, "An improved co-designed AES-ECC cryptosystem for secure data transmission," *International Journal of Information and Computer Security*, vol. 13, no. 1, pp. 118–140, 2020, doi: 10.1504/IJICS.2020.108145.

[40] S. Jagadeesh, S. M. Ali, S. P. G. Selvan, M. Aljanabi, M. Gopianand, and J. P. J. Hephzipah, "Hybrid AES-Modified ECC Algorithm for Improved Data Security over Cloud Storage," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 32, no. 1, pp. 46–56, 2023, doi: 10.37934/ARASET.32.1.4656.