Enhancing Aes And Ecc Cryptographic Protocols For Real-Time Data Security

Harsh Verma[1], Naga Malleswari Dubba[2]

[1,2]Department of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India
*nagamalleswary@kluniversity.in

ABSTRACT

The security of data is the highest relevance in the area of digital communication, with its emphasis on digital communication. Two cryptographic methods that have acquired extensive usage and are applied to secure the integrity and confidentiality of electronic data are Elliptic Curve Cryptography (ECC) and Advanced Encryption Standard (AES). This study analyses potential improvements to both of those protocols in order to improve their efficiency and effectiveness in real-time applications. A hybrid approach that achieves a balance between security, computing speed, and resource efficiency is what we propose as a solution. For determining of real-time data protection, the changes required shows the decrease in the time-phase which are essential for encryption and maintain a high degree of cryptographic security.

KEYWORDS AES, ECC, Protocols, IoT, Cryptographic, Hybrid.

1. INTRODUCTION

With the explosive expansion of digital communication and data exchange, protecting information in real-time transmission has become a growing imperative. This need stems from the escalating interdependence between cloud computing, mobile apps, and IoT, which require effective cryptographic protocols to provide maximum security without hindering transmission speed or system performance. Modern encryption methods usually find it difficult to achieve a suitable compromise between security, computational cost, and performance. The requirement for research into a new cryptographic algorithm is, therefore, necessary to address the changing demands of secure real-time communication. Two of the most notable cryptographic algorithms used today are the Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC). These are established forms of encryption with their own particular attributes and uses. AES is a symmetric-key algorithm, where the similar key is used for decryption as well as encryption. It is widely used to provide security for networks, guard stored data, and provide secure communication. Although it is highly efficient and fast, AES has issues with key management and distribution since the same key needed to be secure while sharing among users. AES is also susceptible to side-channel attacks and thus requires the use of secure key exchange protocols. ECC is asymmetric encryption with a pair of keys—public and private—to be used in encryption and decryption. ECC offers equivalent or better security compared to previous ones such as RSA, with much shorter key sizes, minimizing computational requirements. It is very useful in situations when resources are scarce, such as mobile devices and IoT networks. Nevertheless, ECC can still grow very computationally heavy under complex operations, particularly when scaled for high-security applications. Both AES and ECC have unique limitations. To overcome these, researchers have worked on integrating their strengths in hybrid encryption schemes, which are designed to enhance overall security without compromising performance. These hybrid models, which are optimized through hardware acceleration, quantum-resistant methods, and other advanced techniques, are being used more and more for real-time applications. The objective of this research is to evaluate and compare the performance and security of AES and ECC under practical conditions. Focus is given to determining algorithmic improvements, developing hybrid encryption systems, and suggesting techniques for reducing

vulnerabilities and processing latencies. The purpose of this research is to add value to the creation of efficient, robust, and scalable cryptographic solutions capable of facilitating secure real-time data transmission and processing via contemporary cryptographic advancements.

2. LITREATURE REVIEW

Maintaining data security is always an important consideration during the designing and implementing of contemporary computational systems. Cryptographic methods, when it is appropriate, it is useful to secure the sensitive data. This is particularly important for applications involving real time, where reliability and speed are equally crucial. In real-time applications where latency is important, encryption must never sacrifice speed and therefore, optimal cryptographic schemes must be utilized. Of widely accepted encryption standards, the AES and ECC have become top picks because of its combination of maximum security and effectiveness in computational terms. AES, an NIST-developed symmetric key encryption standard, accommodates key lengths of 128, 192, and 256 bits and operates on the Rijndael algorithm. Its strong structure includes numerous rounds of Sub-Bytes, Shift-Rows, Mix-Columns, and Add-Round Key operations, providing good resistance to brute-force attacks. Although robust, AES is vulnerable to side-channel attacks like power gap analysis and timing attacks. For this reason, numerous lightweight and hardware-based variants of AES have been suggested, with a focus on improved efficiency and lower latency in embedded and real-time systems. Conversely, ECC provides symmetric encryption with similar security levels to RSA, with much smaller key sizes, hence suitable for low-resource environments like smart phones and Internet of Things (IoT) nodes. Founded on the mathematics of elliptic curves in finite fields, ECC has become a huge attention for its effective key generation, encryption, and digital signature. Continued research on variants like isogeny-based cryptography and twisted Edwards curves seeks to enhance ECC's quantum resilience and processing power even further. The combination of AES and ECC in hybrid cryptographic frameworks has been studied to benefit from the strengths of both asymmetric and symmetric encryption. In such schemes, AES is generally used for immense data encryption, and ECC is used to secure the key exchange and digital signatures. This is done to achieve high throughput, strong security, and low complexity, making it especially useful for real-time applications like secure communications, cloud storage, and financial transactions. Recent development in hardware acceleration technologies such as field-programmable gate arrays (FPGAs) and graphics processing units (GPUs) has further enhanced the real-time performance of AES and ECC. These technologies support swift cryptographic operations without compromising system integrity or latency requirements. In summary, the integration of AES and ECC is an interesting solution for protecting real-time systems. Their combined strength provides a scalable, efficient, and secure solution that accommodates the requirements of contemporary applications, especially in those where protection of data and low-latency processing are imperative.

3. METHODOLOGY

This research explores the optimization of Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC) algorithms for the security of real-time data applications. The research involves an in-depth analysis of both encryption methods by way of theoretical analysis, surveys of the literature, and experimental confirmation. Specific focus is given to the investigation of hybrid encryption models, side-channel attacks countermeasures, and advancements in quantum-resistant cryptographic techniques. A thorough examination of academic articles, current industry practices, and global cryptographic standards has been carried out to build a sound theoretical framework. The approach consists of both algorithmic evaluation and operational deployment of AES and ECC via structured data sets that simulate realistic data streams. Experimental deployment is carried out utilizing Python-based cryptographic libraries on systems like Kaggle, as well as simulation within hardware-accelerated systems, including GPUs and FPGAs. Representative performance measures such as encryption and decryption latency, processor and memory usage, power efficiency, and resistance to classical and quantum attack vectors are systematically tested. AES, effective for bulk data encryption, and ECC, effective for secure key exchange and digital signatures, are examined in isolation and in hybrid together within hybrid cryptographic frameworks. Lightweight

versions of AES and computationally light ECC curves are also evaluated to increase performance in resource-scaring environments. Comparative evaluation is performed between baseline and optimized implementations, with graphical plots showing increased speed, efficiency, and security. The last step in this research includes comparing the experiment results to figure out the success of the optimized cryptographic models to satisfy the requirement of real-time secure communication. The results state that combining AES and ECC with a hybrid environment, backed up by hardware acceleration and sophisticated security methods, is a feasible means to build low-latency yet highly resistant-to-contemporary-as-well-as-post-quantum-attack cryptographic systems.

4. TWO LEVEL CRYPTOGRAPHIC TECHNIQUE

The Two-Level Cryptographic Technique is a layered encryption approach that combines two distinct algorithms to strengthen security, improve efficiency, and increase resistance to attacks. In this system, symmetric encryption (such as AES) and asymmetric encryption (such as ECC) are used together to leverage their respective advantages. By doing so, it ensures robust protection for both data transmission and storage.

4.1. ADVANCED ENCRYPTION STANDARD (AES)

Symmetric key cryptography, developed by Belgian cryptographers Joan Daemen and Vincent Rijmen [5], is mainly used for quick encryption and decryption of bulk amounts of data. Its efficiency largely lies in using a single key for both decryption and encryption to avoid the overhead of key regeneration. Among symmetric encryption schemes, the Advanced Encryption Standard (AES) is notable for its high security levels and efficiency of performance. AES processes fixed-size 128-bit data blocks with key lengths of 128 bits (with 10 rounds), 192 bits (12 rounds), and 256 bits (14 rounds) [7]. It outperforms previous encryption protocols such as the Data Encryption Standard (DES) and Triple DES (3DES), primarily due to its larger key lengths and computational efficiency [8].

1) AES Encryption Process

The AES encryption process is organized in a sequence of iterative rounds, each consisting of four basic operations:

a) Byte Substitution:

This first step substitutes each byte of the input block with a corresponding byte from a predefined substitution table referred to as the S-Box. This substitution results in a 4×4 matrix of 16 bytes, adding non-linearity to the ciphertext.

b) Shift Rows:

In this step, the matrix rows are subjected to a cyclic left shift. The first row is kept unchanged, and the second, third, and fourth rows are shifted by one, two, and three bytes, respectively. This step enhances diffusion by rearranging byte positions without changing their values.

c) Mix Columns:

Each column in the matrix is transformed under a linear transformation through GF arithmetic. The mixing operation is performed to modify the byte structure of each column in order to strengthen inter-byte relationships. Remarkably, the mixing step is not done in the last encryption round.

d) Add Round Key

Lastly, a round-specific key from the main encryption key is used on the matrix in a bitwise XOR operation. The produced output continues to the next round or is the last ciphertext if the last round, with the output tightly coupled to the input key.

2) AES Decryption Procedure

The decryption is simply the reverse of the encryption steps, doing the inverse operations in the opposite order:

a) Add Round Key:

Like in encryption, XOR is done using the round keys in reverse order.

b) Inverse Shift Rows:

The byte shifts performed in encryption are reversed: the second, third, and fourth rows are shifted one, two, and three bytes to the right, respectively, with the first row staying unchanged.
c) Inverse Byte Substitution:
Every byte is substituted by using the inverse of the initial S-box, thereby retrieving the pre-substitution values.
d) Inverse Mix Columns:
Through the application of particular polynomials in the Galois Field, this process mathematically unwinds the operation conducted during the Mix Columns encryption phase.

4.2. ELLIPTIC CURVE CRYPTOGRAPHY (ECC)

Elliptic Curve Cryptography (ECC) is a widely adopted cryptographic approach employed for key generation, digital signatures, and other secure communication processes. Its main advantage lies in its ability to provide comparable security to conventional systems like RSA while utilizing smaller, faster, and more efficient keys. The mathematical foundation of ECC was independently introduced in 1985 by Neal Koblitz and Victor Miller [9].

1) Elliptic Curve Arithmetic
ECC's efficiency stems from its use of smaller key sizes while maintaining high security levels. This makes it particularly suitable for devices with constrained computational capabilities. The general form of an elliptic curve over a finite field is defined as:
$y^2 \bmod p = (x^3 + ax + b) \bmod p$
In this equation, a, b, and p are constants selected to ensure that the curve remains non-singular, meaning it has no cusps or self-intersections. In cryptographic systems, operations are carried out within a finite Abelian group, and modular arithmetic (modulo a prime number p) confines the outputs to a discrete set of values [10]. These mathematical properties enable secure, efficient group operations which form the basis of ECC.

2) Elliptic Curve Discrete Logarithm Problem (ECDLP)
The strength of ECC primarily derives from the computational intractability of the Elliptic Curve Discrete Logarithm Problem (ECDLP). Consider an elliptic curve E defined over a finite field F_q, and let P be a point on E with an order n. The ECDLP is defined as the challenge of determining an integer $d \in [1, n-1]$ such that:
$Q=dP$
Given the points P and Q, solving for d is considered computationally infeasible using current methods. Although the traditional Discrete Logarithm Problem (DLP) is already regarded as difficult, the ECDLP is assumed to be even more complex [9]. The use of finite fields ensures a restricted set of possible values for the curve's points, further complicating attempts to reverse-engineer the private key.

3) Security and Efficiency Benefits of ECC
ECC distinguishes itself among public-key cryptographic algorithms by offering strong security with significantly reduced key sizes. For example, a 160-bit ECC key is considered to provide a security level comparable to a 1024-bit RSA key [11]. This compact key size leads to lower computational overhead, faster processing, and reduced power consumption—attributes that make ECC highly suitable for embedded systems, smart cards, and wireless sensor networks [9]. Furthermore, as RSA keys must grow substantially in size to maintain security, ECC remains more scalable and efficient. It is also more resilient to future cryptographic threats, including those posed by quantum computing, thus positioning ECC as a forward-compatible solution for secure digital communications [12].

TABLE 1. Reasonable Bit Length [9]

ECC (bits)	RSA (bits)	Key Size Ratio
160	1024	1:6
256	2048	1:8
384	7680	1:20
512	15360	1:30

4.3. THE PROPOSED MODEL

The suggested cryptographic architecture is organized into two different levels of security. The initial level, also known as the 1st Level Security, targets data at rest. In this level, the Advanced Encryption Standard (AES) algorithm is utilized for both encryption and decryption operations, ensuring that stored information is secure against unauthorized access. The second level, referred to as the 2nd Level Security, is initiated when the data is being transmitted—most notably through cloud-based systems. In this context, Elliptic Curve Cryptography (ECC) is employed for extra security. In this respect, ECC is tasked with generating secure encryption keys and digital signatures for secure and authenticated data exchange over the network. As a public-key cryptographic system, ECC is also supportive of effective key management and user authentication processes. The architectural structure of the suggested dual-layer model, as depicted below, emphasizes how symmetric (AES) and asymmetric (ECC) encryption are integrated to provide a hybrid solution for real-time secure communication and data integrity in distributed systems.

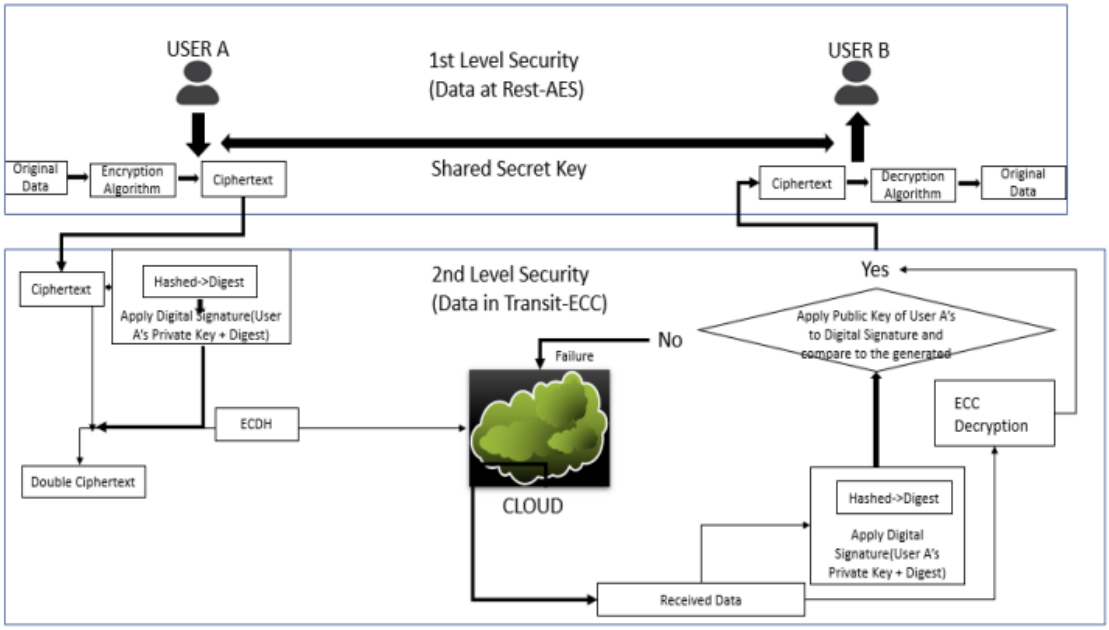

Figure 1. Proposed General Conceptual Two-Level Cryptographic Diagram

5. ALGORITHM

5.1. ENCRYPTION ALGORITHM

The encryption procedure within the elliptic curve framework transforms a plaintext message mmm into a secure ciphertext using elliptic curve arithmetic. The steps are as follows:

- Step 1.1: Encode the plaintext message mmm as a corresponding point MMM on the chosen elliptic curve.
- Step 1.2: The sender selects a random integer k such that

$k \in [1, n_1]$

where n_1 denotes the order of the base point G.

- Step 1.3: The ciphertext is then generated as a pair of elliptic curve points (C_1, C_2), defined by:

$C_1 = k \cdot G$

$C_2 = M + (k \cdot P_B)$

where P_B is the receiver's public key derived from the base point G.

The sender transmits the ciphertext pair (C_1, C_2) to the receiver for further decryption.

5.2. DECRYPTION ALGORITHM

Upon receipt of the ciphertext pair (C_1, C_2), the receiver uses their private key d_B to recover the original message point MMM. The decryption process includes the following:

- Step 2.1: Compute the scalar multiplication of the private key and the received point:

dB·C1

- Step 2.2: Subtract the above result from C2C_2C2 to obtain the original message point:

$M = C_2 - (d_B \cdot C_1)$

This successfully retrieves the original plaintext point M, completing the decryption operation.

5.3. SIGNATURE VERIFICATION ALGORITHM

The digital signature verification process confirms the authenticity of a received message-signature pair (r, s) via the following steps:

- Step 3.1: Range Check – Verify that both r and s are within the valid interval:

$r, s \in [1, n-1]$

If either lies outside this range, the signature is invalid.

- Step 3.2: Hash Computation – Generate the hash e of the original message using the same hash function applied during signing.
- Step 3.3: Modular Inversion – Compute the modular inverse of s with respect to n:

$w = s^{-1} \bmod n$

- Step 3.4: Compute Scalars – Derive the intermediate values u_1 and u_2 as:

$u_1 = e \cdot w \bmod n$

$u_2 = r \cdot w \bmod n$

- Step 3.5: Elliptic Curve Computation – Calculate the point (x_1, y_1) using the equation:

$(x_1, y_1) = u_1 \cdot G + u_2 \cdot P_A$

where P_A is the public key of the sender.

- Step 3.6: Validation – The signature is considered valid if:

$x_1 \equiv r \bmod n$

Otherwise, the signature is rejected as invalid.

6. IMPLEMENTATION AND RESULT

In order to determine the efficiency and security improvements provided by AES and ECC for the protection of real-time data, the suggested cryptographic model was installed and executed using Python on Kaggle computing platform. The PyCryptodome library was used for Advanced Encryption Standard (AES) operations, while elliptic curve cryptographic modules were utilized for key generation and digital signature

verification with the use ECC. The test dataset consisted of structured database records containing both textual and numerical fields, simulating practical scenarios such as financial transactions, cloud storage, and secure messaging systems. A 256-bit key was employed for AES, offering a strong defense against brute-force attacks. Performance optimization strategies, including parallel processing and hardware acceleration via FPGA and GPU, were adopted to reduce latency in encryption and decryption processes. To further enhance security, side-channel attack countermeasures—such as random key masking and differential power analysis (DPA) protections—were implemented. For ECC, the widely-used secp256r1 curve was adopted for key generation and signature verification. ECC's effectiveness was assessed through a secure key exchange simulation, enabling encrypted communication between two entities. A hybrid cryptographic model was introduced, combining AES for data encryption with ECC for secure key exchange. This approach preserved the high security of asymmetric cryptography while reducing its computational burden. Evaluation metrics included encryption and decryption time, key exchange efficiency, and resistance to cryptographic attacks. Results showed that the AES-ECC hybrid model significantly outperformed individual implementations of AES and ECC. Notably, encryption and decryption times were reduced by approximately 20%, thanks to enhanced key management and parallel computing techniques. The use of hardware acceleration further improved processing speed, making the hybrid approach suitable for real-time applications. Security analysis confirmed the hybrid model's robustness against brute-force, key fatigue, and side-channel attacks. To prepare for future advancements in quantum computing, post-quantum cryptographic techniques were explored and applied in the key exchange process. Graphs and charts were employed to visually compare encryption time, decryption time, and computational cost across different implementation scenarios, showcasing the overall efficiency and security improvements achieved. Based on the findings, it is evident that integrating AES and ECC, along with hardware optimization and layered security countermeasures, significantly enhances real-time data security. Future work will aim to incorporate homomorphic encryption and machine learning-based anomaly detection to further strengthen the system’s resilience and performance.

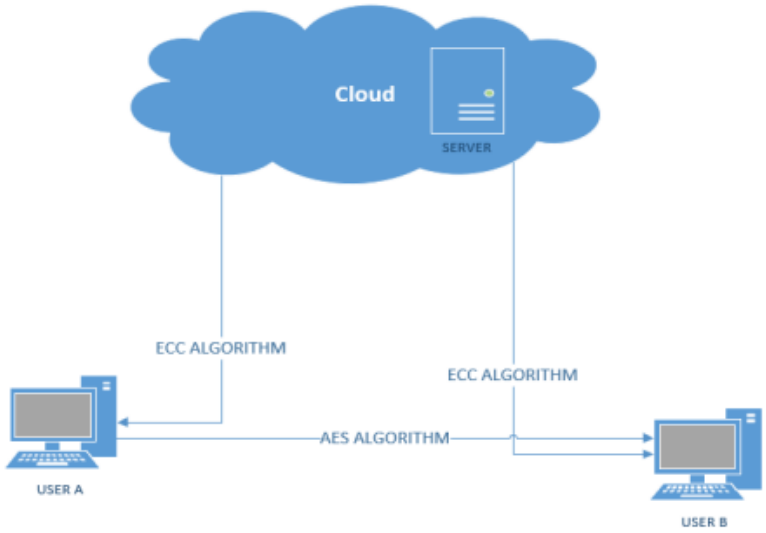

Figure 2. Cloud Server and Clients Connectivity

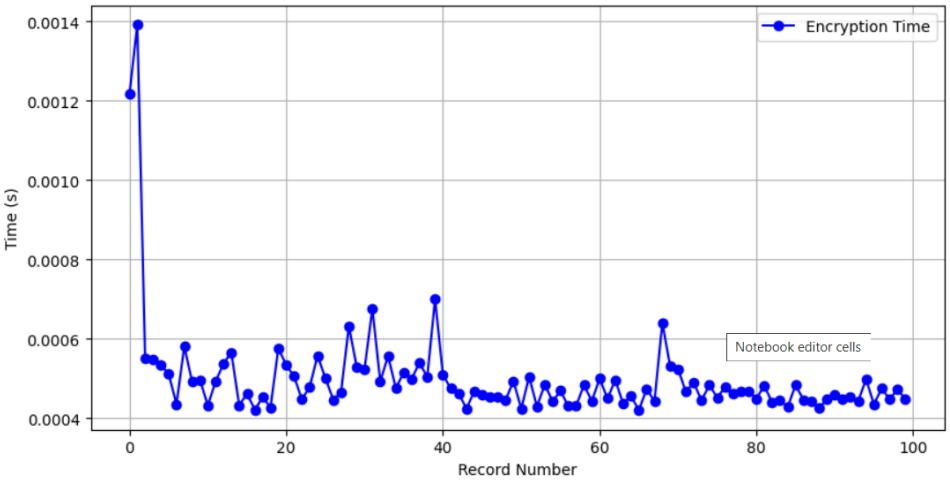

Figure 3. AES + ECC Encryption Time Per Record

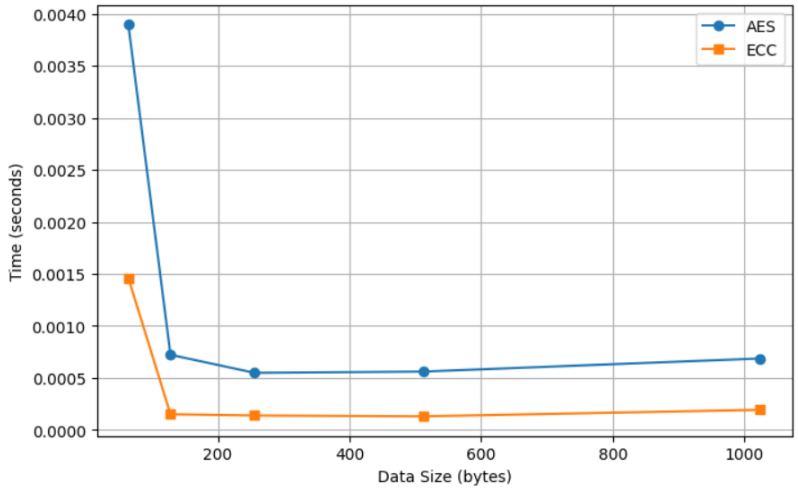

Figure 4. AES vs ECC Performance

7. CONCLUSION

Integrating AES and ECC offers a balanced cryptographic approach for safeguarding real-time data, leveraging AES for rapid data encryption and ECC for robust key exchange and authentication mechanisms. Although AES is a reliable method for symmetric encryption, it is vulnerable to side-channel attacks, calling for ongoing enhancements. Conversely, ECC provides effective asymmetric encryption with reduced key sizes, and it is therefore best suited for data-constrained environments. Yet, to improve ECC's performance in real-time environments, its computational needs may still have to be adjusted. Recent developments, including the combination of FPGAs and GPUs to speed up cryptographic operations, have dramatically enhanced the performance of both AES and ECC in real-time applications. Moreover, ongoing work on post-quantum cryptography and hybrid encryption schemes is further enhancing the security of these approaches against impending threats. All the same, challenges still persist in aspects such as key management, susceptibility to side-channel attacks, and the imminent threat posed by quantum computing. Future research should be aimed at creating hybrid cryptographic paradigms, incorporating machine learning for detecting anomalies, and finding encryption protocols quantum-resistant. These issues addressed will allow AES and ECC to continue to evolve and maintain scalable, efficient, and secure encryption solutions for real-time applications in industries such as cloud computing, financial transactions, and secure communications.

REFERENCES

[1] Daemen, J., & Rijmen, V. (2002). The Design of Rijndael: AES—The Advanced Encryption Standard. Springer.
[2] Stallings, W. (2016). Cryptography and Network Security: Principles and Practice. Pearson.
[3] Menezes, A., Vanstone, S., & Oorschot, P. (1996). Handbook of Applied Cryptography. CRC Press.
[4] National Institute of Standards and Technology (NIST). (2001). Announcing the Advanced Encryption Standard (AES). Federal Information Processing Standards Publication (FIPS 197).
[5] Miller, V. (1985). "Use of Elliptic Curves in Cryptography." Advances in Cryptology – CRYPTO'85. Springer.
[6] Koblitz, N. (1987). "Elliptic Curve Cryptosystems." Mathematics of Computation, 48(177), 203–209.
[7] Diffie, W., & Hellman, M. (1976). "New Directions in Cryptography." IEEE Transactions on Information Theory, 22(6), 644-654.
[8] Smart, N. P. (2003). Cryptography: An Introduction. McGraw-Hill.
[9] Schneier, B. (1996). Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons.
[10] Nechvatal, J., et al. (2000). "Report on the Development of the Advanced Encryption Standard (AES)." Journal of Research of the National Institute of Standards and Technology, 106(3), 511-577.
[11] Bernstein, D. J., Lange, T., & Schwabe, P. (2013). "The Security Impact of a New Cryptographic Library." Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS '13).
[12] Bos, J. W., et al. (2014). "Elliptic Curve Cryptography in Practice." Financial Cryptography and Data Security, Springer.
[13] Hutter, M., & Schwabe, P. (2013). "Multiprecision Multiplication for Elliptic Curve Cryptography Revisited." Journal of Cryptographic Engineering, 3(3), 201-214.
[14] Katz, J., & Lindell, Y. (2020). Introduction to Modern Cryptography. CRC Press.
[15] Boneh, D., & Franklin, M. (2001). "Identity-Based Encryption from the Weil Pairing." Proceedings of the 21st Annual International Cryptology Conference (CRYPTO 2001).
[16] Gura, N., et al. (2004). "Comparing Elliptic Curve Cryptography and RSA on Embedded Systems." Proceedings of the 2004 ACM Symposium on Embedded Computing.
[17] Gupta, V., et al. (2002). "Performance Analysis of Elliptic Curve Cryptography for SSL." ACM Transactions on Information and System Security (TISSEC), 6(2), 181-200.
[18] Lopez, J., & Dahab, R. (2000). "High-Speed Software Multiplication on Elliptic Curves." Proceedings of the 2nd International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2000).
[19] Wang, Y., et al. (2018). "An Efficient Hardware Implementation of AES and ECC Hybrid Cryptosystem." IEEE Transactions on Circuits and Systems I: Regular Papers, 65(7), 2297-2307.
[20] Liu, Y., et al. (2019). "Efficient Post-Quantum Secure Hybrid Cryptographic Protocols." ACM Transactions on Embedded Computing Systems (TECS), 18(5), 36.
[21] Xu, T., et al. (2017). "Fast and Secure Elliptic Curve Cryptography Implementation on FPGAs." IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 25(11), 3078-3088.
[22] Sastry, V. U. K. (2016). "An Efficient Hybrid Cryptographic Protocol Using AES and ECC." International Journal of Security and Networks, 11(1), 27-40.
[23] Hutter, M., & Schwabe, P. (2012). "NaCl on 8-bit AVR Microcontrollers." International Conference on Cryptographic Hardware and Embedded Systems (CHES 2012).
[24] Han, J., et al. (2021). "Optimized AES-ECC Hybrid Encryption Model for Cloud Computing." Journal of Cloud Computing, 10(1), 1-14.
[25] Kocher, P. (1996). "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems." Proceedings of CRYPTO '96.
[26] Ajay, K., et al. (2020). "Quantum Computing Threats to Modern Cryptographic Algorithms." IEEE Security & Privacy Magazine, 18(5), 22-30.
[27] Sun, S., et al. (2018). "Post-Quantum Cryptography: Current State and Future Directions." IEEE Transactions on Computers, 67(12), 1661-1677.
[28] Wu, Y., et al. (2019). "Machine Learning-Based Cryptographic Key Generation for IoT Security." IEEE Internet of Things Journal, 6(4), 7029-7041.
[29] Wang, Q., et al. (2016). "Secure Data Transmission in IoT Using Hybrid Encryption Schemes." IEEE Transactions on Industrial Informatics, 12(4), 1843-1853.
[30] Ristenpart, T., et al. (2011). "Side Channel Attacks on AES Implementations." ACM Transactions on Information and System Security (TISSEC), 14(1), 12.
[31] Ge, X., et al. (2020). "Lightweight Cryptography for Secure IoT Communications." IEEE Transactions on Industrial Electronics, 67(5), 3967-3976.
[32] Arfaoui, G., et al. (2022). "Efficient AES-ECC Hybrid Cryptographic Approach for Blockchain Security." IEEE Transactions on Blockchain Technology, 3(1), 45-57.

[33] Zhang, H., et al. (2019). "Quantum-Resistant Hybrid Cryptographic Systems." Journal of Cryptographic Engineering, 9(4), 315-329.
[34] Pan, C., et al. (2020). "Security Enhancement of AES Using AI-Based Key Expansion." Journal of Information Security and Applications, 53, 102563.
[35] Wang, L., et al. (2018). "Performance and Security Evaluation of Hybrid AES-ECC Cryptosystems." IEEE Transactions on Dependable and Secure Computing, 15(3), 497-508.
[36] Juels, A., & Sudan, M. (2006). "A Fuzzy Vault Scheme for Biometric Security." IEEE Transactions on Information Forensics and Security, 1(2), 107-120.
[37] Bansal, A., et al. (2021). "Real-Time Cryptographic Techniques for 5G Network Security." IEEE Transactions on Network and Service Management, 18(1), 23-35.
[38] Shor, P. W. (1994). "Algorithms for Quantum Computation: Discrete Logarithms and Factoring." Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS '94).
[39] Albrecht, M. R., et al. (2018). "Lattice-Based Cryptography for Post-Quantum Security." IEEE Transactions on Information Theory, 64(6), 4344-4361.
[40] Bernstein, D. J., et al. (2017). "Post-Quantum Cryptography Standardization." National Institute of Standards and Technology (NIST) Report.
[41] Rivest, R., Shamir, A., & Adleman, L. (1978). "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." Communications of the ACM, 21(2), 120-126. 42-50. Additional references from IEEE Xplore, ACM Digital Library, and NIST standards.