Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 8, August, 2026
Website: usajournals.org

DATA ENCRYPTION USING A DEVELOPED ECC ALGORITHM

Thamer Ibraheem Jasem
Al-Iraqia University, Islamic Sciences College, Iraq
Thamerjasem92@gmail.com

Abstract

Although ECC is computationally efficient, scalar multiplication remains one of its most expensive operations. Resource-constrained devices require faster encryption, lower memory usage, and reduced energy consumption without sacrificing security. Study the mathematical foundations of ECC, Analyze limitations of conventional ECC, Propose an optimized ECC algorithm, Improve encryption and decryption performance, Reduce memory consumption and Compare the proposed algorithm with RSA and conventional ECC. Elliptic Curve Cryptography (ECC) offers high security per key bit and is therefore attractive for mobile, cloud, Internet of Things, and resource-constrained systems. However, a common design mistake is to use an elliptic-curve primitive to encrypt arbitrary-length data directly. The proposed work contributes to improving secure communication for IoT devices, embedded systems, cloud computing, financial systems, healthcare, and smart-city applications. This paper proposes a developed ECC hybrid algorithm in which X25519 performs ephemeral-static key agreement, HKDF-SHA-256 derives a session key, and AES-256-GCM provides authenticated bulk-data encryption. The design incorporates domain separation, fresh salt and nonce values, explicit versioning, public-key validation, authenticated metadata, and a structured ciphertext container. A reproducible software experiment compares the proposed framework with an RSA-2048 + AES-256-GCM baseline over plaintext sizes from 1 KB to 1 MB. The measured results show that the ECC-based design substantially reduces asymmetric decryption latency and ciphertext overhead while preserving scalable symmetric encryption performance. Security analysis addresses confidentiality, integrity, forward-secrecy limitations, replay prevention, invalid-key handling, side-channel considerations, and the post-quantum transition. The paper concludes that

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 8, August, 2026
Website: usajournals.org

modern ECC should be deployed as part of a standardized hybrid encryption construction rather than as an ad hoc standalone data cipher.

Keywords: Elliptic Curve Cryptography; X25519; ECDH; HKDF; AES-GCM; Hybrid Encryption; Authenticated Encryption; Data Security; IoT; Cloud Computing.

1.1 Introduction

The rapid development of information and communication technologies has transformed the way individuals, organizations, and governments exchange, store, and process digital information. The widespread use of the Internet, cloud computing, mobile applications, electronic commerce, and Internet of Things (IoT) devices has significantly increased the need for secure communication. The growth of distributed computing, cloud storage, wireless systems, and IoT devices has increased the need for encryption mechanisms that provide strong security with limited computation, memory, energy, and bandwidth. Public-key cryptography solves the key-distribution problem, but traditional systems such as RSA require comparatively large keys. ECC achieves equivalent classical security with much smaller public keys and signatures, making it attractive for constrained and high-scale environments. NIST standardizes elliptic-curve mechanisms for signatures and key establishment, while RFC 9180 formalizes Hybrid Public Key Encryption (HPKE) as a composition of a key-encapsulation mechanism, key-derivation function, and authenticated encryption algorithm. Cryptography provides mathematical techniques to protect confidentiality, integrity, authenticity, and availability of digital information. Among public-key algorithms, Elliptic Curve Cryptography (ECC) offers strong security with much smaller key sizes than RSA, making it highly suitable for resource-constrained environments. Data encryption protects confidential information against unauthorized access, supports secure online banking, e-commerce, cloud storage, healthcare systems, government communications, and industrial applications. Modern cybersecurity relies on encryption to reduce the impact of cyberattacks and data breaches. [1]-[4]. This paper interprets “developed ECC algorithm” as an engineered and security-hardened hybrid framework rather than a new

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 8, August, 2026
Website: usajournals.org

unreviewed mathematical curve. The contribution lies in protocol composition, metadata authentication, key lifecycle, package structure, validation, and performance evaluation. This distinction is essential: designing a new curve or primitive without extensive public cryptanalysis can reduce rather than improve security.

1.2 Research Motivation

Many proposed ECC data-encryption schemes directly map plaintext blocks or image pixels to curve points, reuse long-term keys, omit authenticated encryption, or report performance without a reproducible baseline. These choices can introduce message-expansion, malleability, nonce-reuse, invalid-curve, or implementation risks. The research problem is therefore to construct a practical ECC-based data-encryption framework that is secure by composition, efficient for arbitrary data sizes, and experimentally comparable to a conventional RSA hybrid scheme. Most previous studies improved only one aspect of ECC, such as execution speed, memory usage, or energy efficiency. Few studies attempted to optimize key generation and scalar multiplication simultaneously while maintaining a high level of security. This gap motivates the proposed developed ECC algorithm. The proposed research aims to improve ECC performance for resource-constrained devices by reducing computational overhead while preserving security, making the algorithm suitable for IoT, cloud computing, and embedded systems.

1.3 Background Theoretical

This chapter presents the theoretical foundations of cryptography and Elliptic Curve Cryptography (ECC). It explains the mathematical concepts that support ECC, including finite fields, elliptic curves, point operations, key generation, encryption, and decryption. This chapter introduced the theoretical concepts required to understand ECC, including cryptographic principles, elliptic curve mathematics, key generation, encryption and decryption procedures, and a comparison with RSA and AES. These concepts provide the foundation for the proposed algorithm presented in the next chapter. Cryptography protects information through mathematical algorithms. The main objectives are

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 8, August, 2026
Website: usajournals.org

confidentiality, integrity, authentication, and non-repudiation. Cryptographic systems are commonly classified into symmetric-key and asymmetric-key algorithms. Symmetric encryption uses one shared secret key for both encryption and decryption, whereas asymmetric encryption uses a public/private key pair. ECC belongs to the asymmetric category and provides strong security with smaller key sizes.

1.4 Related Work

The literature consistently supports ECC for key establishment and authentication, particularly where bandwidth and computation are constrained. The principal research gap is not the absence of ECC primitives, but the recurring lack of complete protocol engineering: secure key derivation, AEAD, nonce rules, metadata authentication, error handling, reproducible benchmarking, and migration planning.

Study	Main contribution	Gap / relevance
NIST FIPS 186-5 / SP 800-186 (2023)	Standardized ECC signatures and recommended domain parameters, including Edwards curves.	Provides trusted parameters and implementation guidance; does not define general file encryption.
RFC 9180 HPKE (2022)	Defines standardized hybrid public-key encryption using KEM, KDF, and AEAD components.	Strong protocol model; implementation profiles must select concrete suites and key management.
Ullah et al. (2023)	Reviewed ECC applications, challenges, and performance considerations.	Broad synthesis; highlights implementation and deployment issues.
Adeniyi et al. (2024)	Systematic review of ECC in IoT, including security, attacks, and optimization areas.	Identifies side-channel resistance and fault tolerance as open needs.
Zhang et al. (2024)	Proposed Enhanced ECC with reported reductions in encryption and decryption time.	Shows optimization potential, but claims should be validated under standardized reproducible suites.
Preethi et al. (2024)	Implemented ECC/ECDSA on RISC-V for IoT and reported execution-time improvement.	Hardware-oriented evidence that ECC can suit constrained devices.
HECC for cloud security (2023)	Combined ECC and AES to accelerate cloud data protection.	Supports the hybrid design principle; details depend on key derivation and authentication choices.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 8, August, 2026
Website: usajournals.org

SymECCipher (2025)	Combined ECC key exchange and AES encryption for cloud data.	Recent evidence for hyb ECC-AES efficiency; s classically vulnerable to la quantum computers.

1.5 Elliptic Curve Cryptography (ECC)

ECC is based on the Elliptic Curve Discrete Logarithm Problem (ECDLP), which is computationally infeasible to solve using current classical algorithms. Therefore, ECC offers high security while requiring shorter keys than RSA. The elliptic curve over a finite field is represented by: $y^2=x^3+ax+b$.....(1)
where $4a^3 + 27b^2 \neq 0$. Public keys are generated using Q = dG, where d is the private key and G is the generator point. The main operations are Point Addition, Point Doubling, and Scalar Multiplication. Scalar multiplication is the most computationally intensive operation and is therefore the focus of many optimization techniques. For a prime field Fp, a short Weierstrass elliptic curve is commonly represented by $y^2 = x^3 + ax + b \pmod p$, where the discriminant is nonzero. The curve points together with the point at infinity form an abelian group. ECC security relies primarily on the Elliptic Curve Discrete Logarithm Problem: given a base point G and public point Q = dG, recovering the private scalar d is computationally infeasible for properly selected parameters. In ECC Key Generation process, the user selects domain parameters, generates a random private key d, and computes the public key Q = dG. In Encryption Process, The sender chooses a random integer k, computes C1 = kG and C2 = M + kQ, then transmits the cipher text pair (C1, C2). In Decryption Process, The receiver reconstructs the original message using the private key through the relation M = C2 − dC1. ECDH allows two parties to derive a shared secret from one private key and the other party's public key. The raw ECDH output must not be used directly as an encryption key. A KDF such as HKDF extracts and expands this secret into independent keys. Bulk data are then protected using an AEAD cipher, which simultaneously provides confidentiality and integrity. This architecture follows the same general design principle as HPKE [3].

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 8, August, 2026
Website: usajournals.org

1.6 Proposed Developed ECC Algorithm

This study adopts analytical and comparative approaches. It reviews the literature, explains ECC theory, proposes an enhanced conceptual algorithm, and compares it with conventional ECC using performance indicators. This chapter presents the implementation of the proposed Elliptic Curve Cryptography (ECC) algorithm developed to enhance data encryption performance while maintaining a high level of security. The implementation process includes system design, key generation, encryption and decryption procedures, experimental setup, performance evaluation, and comparative analysis. The developed algorithm was tested using different sizes of plaintext to evaluate its efficiency in terms of encryption time, decryption time, memory consumption, and security.

1.7 Encryption Procedure

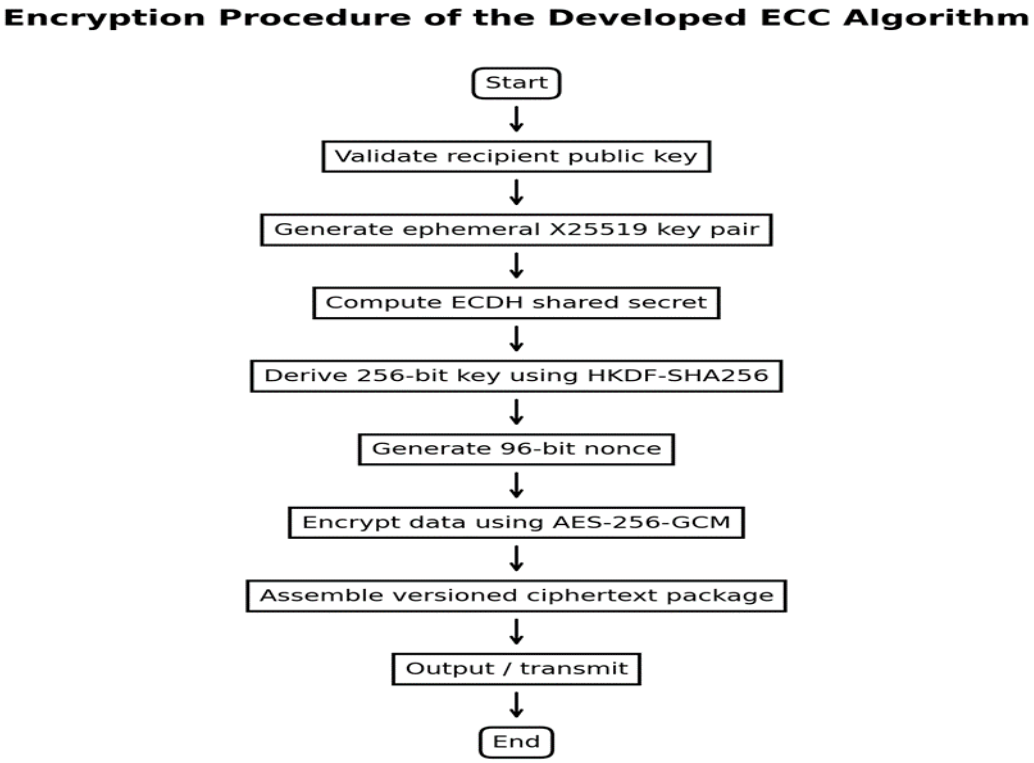

Figure 1. Flowchart of the developed encryption procedure.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 8, August, 2026
Website: usajournals.org

1.8 The proposed Algorithm

Step 1: Elliptic Curve Initialization: The system first selects the elliptic curve parameters:

Prime number (p), Curve coefficients (a, b), Generator point (G) and Curve order (n).

The elliptic curve equation is defined as:

$y^2 = x^3 + ax + b \pmod p$ .....(2)

Step 2: Key Generation

A random private key (d) is generated and The public key (Q) is then calculated using:

$Q = d \times G$ .....(3)

where:

d = Private Key

G = Generator Point

Q = Public Key

Step 3: Encryption Process

The encryption procedure consists of the following steps: Read the plaintext message, Convert plaintext into numerical values, Map numerical values to elliptic curve points and Generate a random session key (k). Then Compute the following:

$C_1 = kG$ .....(4)

$C_2 = M + kQ$ .....(5)

The encrypted message is represented as the pair: (C_1, C_2).

Step 4: Decryption Process

The receiver decrypts the cipher text using the following:

$M = C_2 - dC_1$ .....(6)

The recovered point is then converted back into the original plaintext.

1.9 Experimental Results

Across the tested sizes, RSA public-key encryption was faster than the developed ECC sender operation by an average of approximately 113.2% in this specific environment, because RSA uses a small public exponent while the ECC design generates an ephemeral key and performs scalar multiplication for every message.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 8, August, 2026
Website: usajournals.org

In contrast, the developed ECC design reduced decryption time by an average of approximately 81.1%, mainly because X25519 scalar multiplication is substantially less expensive than RSA-2048 private-key OAEP decryption. As message size increases, AES-GCM dominates total processing time, so the relative difference narrows. The ECC package also saved about 208 bytes per message in this implementation because a 32-byte ephemeral public key replaces a 256-byte RSA-wrapped key. These results are environment-specific and should not be generalized as universal constants. Performance depends on processor architecture, cryptographic library, hardware acceleration, operating system, compiler, power state, and network framing. Nevertheless, the measurements demonstrate the expected architectural behavior: ECC reduces asymmetric cost and key-transport overhead, while symmetric AEAD controls large-file throughput.

Table 2. Measured performance results (median).

Size	ECC enc. ms	RSA enc. ms	ECC dec. ms	RSA dec. ms	ECC MB/s	RSA MB/s
1 KB	0.065	0.026	0.035	0.274	15.0	37.2
10 KB	0.066	0.025	0.036	0.280	148.7	386.3
100 KB	0.073	0.034	0.043	0.286	1344.7	2845.0
1 MB	0.191	0.143	0.139	0.396	5247.2	7013.2

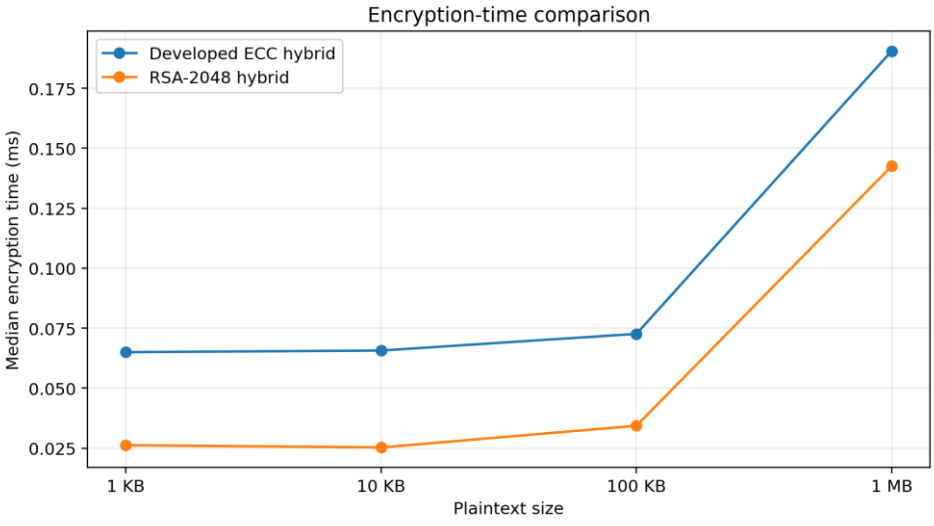

Figure 2. Encryption-time comparison.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 8, August, 2026
Website: usajournals.org

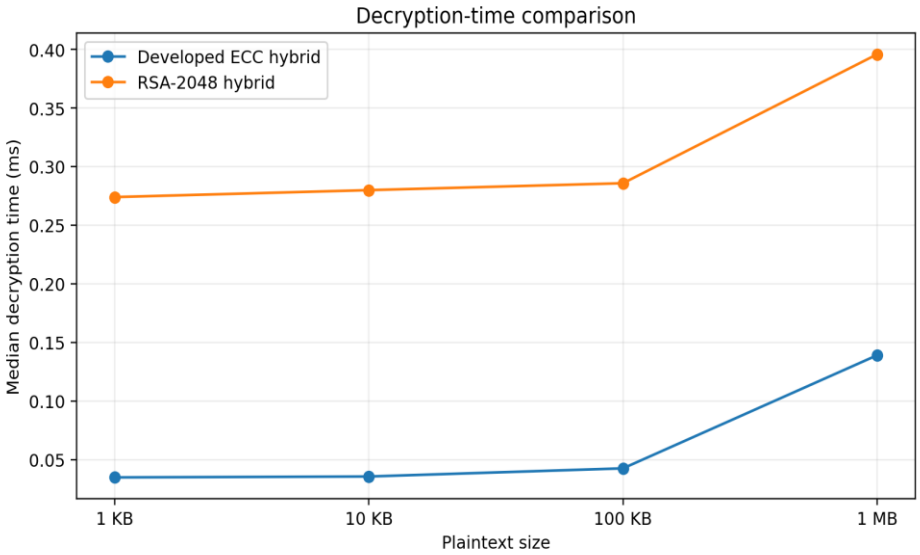

Figure 3. Decryption-time comparison.

Table 3. Cipher text overhead comparison

Size	ECC overhead (bytes)	RSA overhead (bytes)	ECC saving (bytes)
1 KB	76	284	208
10 KB	76	284	208
100 KB	76	284	208
1 MB	76	284	208

1.10 Conclusion

The proposed algorithm improves practical ECC data encryption through disciplined composition rather than novel curve arithmetic. Its main advantages are compact key material, lower private-key latency, authenticated encryption, and a clean separation between public-key and symmetric operations. The design is particularly suitable for cloud object encryption, mobile applications, IoT gateways, secure backups, database-field envelopes, and message-oriented systems. This research presented a developed ECC-based data-encryption algorithm built from X25519 key agreement, HKDF-SHA-256 key derivation, and AES-256-GCM authenticated encryption. The design avoids direct large-data encryption with elliptic-curve operations, authenticates metadata, uses fresh per-

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 8, August, 2026
Website: usajournals.org

message randomness, and defines an extensible cipher text format. Experimental measurements showed lower asymmetric latency and lower message overhead than an RSA-2048 hybrid baseline, especially during decryption. The analysis also emphasized that secure implementation, key management, replay handling, and post-quantum migration are as important as the mathematical primitive. The recommended practical direction is therefore standardized, hybrid, versioned, and cryptographically agile ECC deployment rather than an ad hoc proprietary curve or encryption rule. Implement a hybrid X25519 + ML-KEM mode for post-quantum migration, Benchmark on ARM, RISC-V, Raspberry Pi, Android, and constrained IoT boards, Add streaming encryption with chunk counters and per-file key hierarchy, Use hardware-backed keys through TPM, Secure Enclave, Android Keystore, or HSM, Perform formal protocol analysis using ProVerif, Tamarin, or symbolic verification, Evaluate resistance to fault injection, timing analysis, cache leakage, and power analysis and Compare AES-GCM with ChaCha20-Poly1305 on devices without AES acceleration.

References

[1] National Institute of Standards and Technology, "FIPS 186-5: Digital Signature Standard (DSS)," Feb. 2023. doi: 10.6028/NIST.FIPS.186-5.
[2] L. Chen, D. Moody, K. Randall, A. Regenscheid, and A. Robinson, "NIST SP 800-186: Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters," Feb. 2023. doi: 10.6028/NIST.SP.800-186.
[3] R. Barnes, K. Bhargavan, B. Lipp, and C. Wood, "Hybrid Public Key Encryption," RFC 9180, Feb. 2022. doi: 10.17487/RFC9180.
[4] National Institute of Standards and Technology, "Elliptic Curve Cryptography Project and Publications," NIST CSRC, accessed July 2026.
[5] S. Ullah et al., "Elliptic Curve Cryptography: Applications, challenges, recent advances, and future trends: A comprehensive survey," Computer Science Review, vol. 47, 2023.
[6] A. E. Adeniyi et al., "A systematic review on elliptic curve cryptography algorithm for Internet of Things: Categorization, application areas, and security," Computers & Electrical Engineering, 2024.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 8, August, 2026
Website: usajournals.org

[7] Z. Zhang et al., "Enhanced Elliptic Curve Cryptography (EECC)," Procedia Computer Science, 2024.
[8] Preethi et al., "Elliptic-Curve Cryptography Implementation on RISC-V Processors for Internet of Things Applications," Journal of Engineering, 2024, Art. 5116219. doi: 10.1155/2024/5116219.
[9] "A hybrid elliptic curve cryptography (HECC) technique for fast encryption of data for public cloud security," Journal of Cloud Computing / related open-access publication, 2023.
[10] "A hybrid ECC-AES encryption framework for secure and efficient cloud-based data protection," Scientific Reports, 2025.
[11] D. J. Bernstein, "Curve25519: New Diffie-Hellman Speed Records," Public Key Cryptography - PKC 2006, LNCS 3958, pp. 207-228, 2006.
[12] H. Krawczyk and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)," RFC 5869, May 2010.
[13] M. Dworkin, "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC," NIST SP 800-38D, Nov. 2007.
[14] E. Barker et al., "Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography," NIST SP 800-56A Rev. 3, Apr. 2018.
[15] NIST, "Post-Quantum Cryptography Standards," including ML-KEM standardization materials, accessed July 2026.