ISSN 2278-3091
Volume 14, No.6, November – December 2025
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse061462025.pdf
https://doi.org/10.30534/ijatcse/2025/061462025

Evaluating the Effectiveness of Cryptographic Techniques for Data Protection in Collection Systems

Karen Kate A. Remorosa[1], Myelinda A. Baldelovar[2]
[1] North Eastern Mindanao State University, Philippines, karenkate.remorosa30@gmail.com
[2] North Eastern Mindanao State University, Philippines, mbaldelovar@nemsu.edu.ph

Received Date: October 21, 2025 Accepted Date: November 27, 2025 Published Date: December 06, 2025

ABSTRACT
In data collection systems, protecting sensitive information is very important, especially during transmission, storage, and processing of the information gathered. Now, cryptographic techniques are used to ensure the confidentiality, integrity, and authenticity of the collected, processed, and stored data. This study evaluates the effectiveness of three commonly used cryptographic techniques—Advanced Encryption Standard (AES) for symmetric encryption, Rivest–Shamir–Adleman (RSA) or Elliptic Curve Cryptography (ECC) for asymmetric encryption, and Secure Hash Algorithm (SHA-256) for hashing, within the factors of securing critical or vital data collection systems in simulating a Human Resource Management (HRM) environment. The evaluation focuses on balancing required security levels (128-bit minimum) with optimization of computational performance metrics which includes throughput and latency. The results will propose an optimal Hybrid ECC-AES-GCM Framework for scalable and efficient enterprise data protection.

Key words: Asymmetric key, Cryptography, Data Collection System, Hashing.

1. INTRODUCTION

Cryptographic techniques are important for securing data in collection systems, like in Human Resource Systems and the likes [1]. These systems uses cryptographic techniques or algorithms, also known as ciphers; to encrypt and decrypt messages, ensuring confidentiality, integrity, and authenticity of the collected data. Security is the major concern when crucial information is stored and transferred across the internet, where the information or data is no longer protected by physical boundaries [1]. Common techniques like RSA and El-Gamal, based on discrete logarithm concepts, are frequently used for data security, particularly over networks like the internet [1]. These techniques safeguard's data during transmission and storage which is a protection from unauthorized access or modification. The selection of specific cryptographic methods depends on factors such as the sensitivity of the data, the required security level, and the performance characteristics of the system.

In today's era, cryptography technique is used to mask the data transmitted online, which is necessary to securely store data over the network. As technology grows day by day, the need for data security over the channel of communication is greatly increased. Cryptography protects crucial data by changing it into unclear data that can only be accessed by authorized receivers, who then convert the uncertain data into the original textual content [2]. Several cryptographic schemes are also used for secure communication [3]. As we all know, cryptography is the science of both encryption and decryption [4]. Nowadays, all works related to banking, ATM card, credit card, marketing, E-commerce, etc., are doing with the aid of the internet, so there must always be protection provided over the network.

In this study, the researchers' focuses on the common algorithms used for securing data in collection systems, these cryptographic techniques are AES (Advanced Encryption Standard) Symmetric Encryption, RSA or ECC Asymmetric Encryption, and SHA-256 Cryptographic Hashing. These techniques will be evaluated based on their effectiveness in the most common collected data in systems.

2. METHODOLOGY

The methodological framework for this evaluation was constructed to rigorously assess the operational effectiveness of three distinct classes of cryptographic primitives—Advanced Encryption Standard (AES), Rivest–Shamir–Adleman (RSA) or Elliptic Curve Cryptography (ECC), and Secure Hash Algorithm (SHA-256)—within the context of securing high-stakes data collection systems, specifically simulating a Human Resource Management (HRM) environment [2]. Effectiveness in this study is defined as the successful achievement of established security levels (128-bit minimum) juxtaposed with optimized computational performance metrics, including throughput, latency, and resource utilization [3].

2.1 Research Design and Security Objectives

The research employed a comparative experimental design using simulated enterprise-level server environments. This approach in simulation was necessary to identify the performance variables required to the cryptographic algorithms, which will minimize the variance introduced by external factors such as network conditions or application-layer complexities often found in real-world systems [4].

2.1.1 Security Objectives (CIA Triad Plus)

The criteria are based on the main goals for keeping the HRMsystem's sensitive data safe, which must follow very strict rules to protect its confidentiality and integrity [2]. [5] These objectives align with the Confidentiality, Integrity, and Availability (CIA) triad, augmented by requirements for authentication and non-repudiation:

Confidentiality (C): The guarantee that sensitive data remains obscured from unauthorized parties, secured by symmetric encryption, specifically AES.

Integrity and Authenticity (I & A): The assurance that data has not been modified in transit or at rest, and that the source is verifiable. This is provided through cryptographic hashing (SHA-256) and the application of authenticated encryption modes (e.g., AES-GCM) [6].

Key Establishment and Non-Repudiation (Trust): Mechanisms ensuring secure, authenticated communication channels and verifiable data origin, provided by asymmetric algorithms (RSA or ECC) used for key exchange and digital signatures.

2.1.2 Selected Cryptographic Primitives and Key Sizes

To facilitate a comprehensive and standardized comparative analysis, all algorithms were standardized to provide a minimum security level equivalent to 128 bits, adhering to recommendations from recognized standards bodies like NIST [7][8]. [9] The selected primitives were:

Symmetric Encryption: AES-256, chosen for its high security level (equivalent to 128-bit security via 3072-bit RSA) and compliance with the study's focus [2]. [1] AES operates with a fixed 128-bit block size and utilizes 14 rounds for 256-bit keys [1].

Asymmetric Encryption/Key Exchange: ECC P-256 was primarily selected, supplemented by RSA 3072-bit for comparative benchmarking. RSA 3072-bit provides a security strength roughly equivalent to 128-bit symmetric protection [7]. [10] ECC P-256 offers the same security equivalence with significantly reduced key sizes, minimizing computational load [9][11].

Hashing: SHA-256 was selected as the standard cryptographic hashing primitive, generating a 256-bit digest, crucial for integrity checks and digital signatures [2].

2.2 Definition of Performance Metrics and Computational Environment

The study utilized metrics designed to assess performance in high-throughput, enterprise-scale environments [12].

2.2.1 Performance Metrics

Bulk Data Performance (Throughput): Measured in Megabytes per second (MB/s). This metric quantified the efficiency of securing large data transfers, such as batch processing of employee data, database backups, or system archives [13]. It was the primary metric for symmetric encryption (AES) and hashing (SHA-256) performance.

Transactional Performance (Latency): Measured in milliseconds (ms) or cryptographic operations per second (ops/s). This metric is crucial for evaluating the time overhead of setting up secure connections, verifying digital signatures, and performing individual data access checks (e.g., initial TLS handshake latency) [14]. This was the primary metric for asymmetric operations (RSA and ECC).

Resource Consumption: Assessed through metrics such as CPU utilization time and memory consumption during cryptographic operations [3]. [15] This provides insight into the scalability and cost efficiency of the chosen algorithms, particularly relevant in virtualized or cloud-based data collection systems.

2.2.2 Computational Environment

The experimental simulation was executed on a high-performance server-class Central Processing Unit (CPU) utilizing an x86 architecture. This setup was chosen specifically because it supports dedicated hardware acceleration features, particularly the AES-New Instructions (AES-NI) [16]. [17] Enterprise data collection systems typically leverage these hardware optimizations to handle large volumes of data efficiently, and testing under these conditions provides the most realistic assessment of operational effectiveness.

2.3 Data Modeling and Simulation Framework

2.3.1 Data Modeling: HRM Data Payloads

To accurately reflect the data volumes and characteristics found in HRM systems, the study utilized two distinct categories of synthesized test data payloads [18][19]:

Small Transactional Payloads (4 KB to 100 KB): These payloads simulate data involved in real-time, latency-critical operations, such as individual PII record updates, key exchange negotiation messages, biometric authentication tokens, or requests for digital signature

verification [20]. These datasets were used primarily for latency testing of asymmetric algorithms.

Large Bulk Payloads (256 MB to 1 GB): These files were designed to test sustained throughput performance for operations like full database encryption, annual performance evaluation archives, or batch data processing jobs [6]. [13] Both structured text data (simulating database records) and mixed binary data (simulating documents or multimedia files) were included, as data type can sometimes influence algorithm execution performance included, as data type can sometimes influence algorithm execution performance [18][19].

2.3.2 Simulation Framework and Algorithm Implementation

The algorithms were implemented using established, optimized cryptographic library routines (common in modern server programming environments like OpenSSL or enterprise-grade Java/C# frameworks) to ensure the implementation reflected industry best practices for efficiency and standardization [12][14].

A. AES-256 Mode Selection

A critical decision involved the mode of operation for AES-256. The study mandated the evaluation of AES-256 using the Galois/Counter Mode (GCM), which is an Authenticated Encryption with Associated Data (AEAD) standard [15]. GCM was selected over older, non-authenticated modes (like CBC or CTR alone) for several compelling reasons [6]:

Integrated Security: GCM concurrently provides confidentiality, integrity, and authenticity, mitigating risks such as bit-flipping and chosen ciphertext attacks that afflict non-authenticated modes [6][15].

Performance Optimization: GCM is designed to be parallelizable and pipelined, enabling exceptionally high throughput rates crucial for securing high data rates (10 Gbps and above). This parallel structure allows the operation to efficiently utilize multi-core processors and hardware acceleration, unlike modes such as CBC-MAC, which cannot be parallelized [15].

B. Asymmetric Algorithm Implementation

The implementation of asymmetric algorithms strictly adhered to current security recommendations:

RSA Implementation: The RSA 3072-bit keys were employed with modern padding schemes: RSA-OAEP-256 for key wrapping (encryption) and RSASSA-PSS using SHA-256 for digital signatures. Legacy padding schemes like RSAES-PKCS1-V1_5 were avoided due to known security weaknesses [12].

ECC Implementation: ECC utilized the NIST-recommended Curve P-256 for both key establishment (ECDH) and digital signatures (ECDSA) [14].

Table 1: Performance Benchmarking Test Parameters

Parameter Category	Description/Value	Rationale/Relevance to HRM
Test Environment	High-performance server environment (x86 architecture with AES-NI/Hardware Acceleration)	Mimics enterprise-level HR system server processing and maximizes efficiency potential [16],.[17]
Symmetric Algorithm	AES-256	Required by the study.[2]
Symmetric Mode of Operation	Galois/Counter Mode (GCM)	Provides Confidentiality, Integrity, and Authenticity (AEAD); parallelizable for high throughput [6],.[15]
Asymmetric Algorithm 1	RSA 3072-bit (OAEP-256/PSS)	Standard key length providing 128-bit security equivalence [7],.[10]
Asymmetric Algorithm 2	ECC P-256 (ECDH/ECDSA)	Key size equivalent to 3072-bit RSA for 128-bit security level, offering high efficiency [9],.[11]
Hashing Algorithm	SHA-256	Standard for integrity checking and digest generation.[2]
Bulk Data Payload Size	256 MB to 1 GB	Tests sustained throughput for archival/batch processing.[13]
Transactional Data Payload Size	4 KB to 100 KB	Tests key exchange/signing latency for real-time authentication.[20]

Table 1 summarizes the specific parameters used in the benchmarking simulation.

3. RESULTS AND DISCUSSION

The performance evaluation yielded results that distinctly delineate the optimal role for each cryptographic primitive within a comprehensive data collection security framework. The findings underscore that a high level of security is achieved not by algorithm strength alone, but by strategically balancing the trade-offs between key size, computational complexity, and resulting operational efficiency.

3.1 Comparative Analysis of Security Strength and Key Equivalence

A fundamental measure of cryptographic effectiveness is the "bits of security," which quantifies the hypothetical computational effort (expressed as 2^n operations) required for an attacker to successfully break the encryption scheme [7]. For the protection of sensitive HRM data, a minimum security level of 128 bits is widely accepted as the required standard [8].

3.1.1 RSA Key Scaling Challenge

The evaluation demonstrated a significant disparity in the key lengths required by RSA versus ECC to achieve equivalent security strength. While AES-128 provides 128 bits of security directly, RSA requires a 3072-bit key to offer the same level of resistance against factoring attacks [7]. The computational cost of RSA key generation, signing, and encryption scales polynomially, specifically cubically, with the key length [10]. To achieve the highest security level, 256 bits (often required for top-secret or mission-critical data), RSA demands keys up to 15360 bits [7]. The computational overhead associated with operating at this length renders high-strength RSA highly impractical for systems requiring frequent authentication or key exchange [10][11].

3.1.2 ECC Efficiency in High-Security Environments Elliptic Curve Cryptography (ECC) emerges as the clearly superior technique for asymmetric operations in high-security environments due to its dramatically reduced key size requirement for equivalent security [9]. A 256-bit ECC key(P-256) provides security equivalent to a 3072-bit RSA key, meeting the 128-bit security standard [7]. To reach the 256-bit security level, ECC requires only a 521-bit key [7] [10]. This performance-to-security ratio is paramount for operational effectiveness in modern systems, particularly where minimizing resource consumption or improving response time is critical. The reduced overhead associated with ECC ensures that strong, forward-looking security can be maintained without imposing prohibitive computational burdens [1][11].

Table 2: Cryptographic Security Equivalence and Recommended Key Sizes

Security Strength (Bits)	Symmetric Key (e.g., AES)	Elliptic Curve Key (ECC)	Integer Factorization/DL (RSA/DH)
112 (Minimum Acceptable)	3TDEA	P-224	2048 bits
128 (Recommended Standard)	AES-128	P-256	3072 bits
192 (High Security)	AES-192	P-384	7680 bits
256 (Maximum Security)	AES-256	P-521	15360 bits

In Table 2, the data confirms that as required security levels increase, the effectiveness of RSA diminishes rapidly due to the exponential increase in resource requirements. Consequently, the analysis concludes that ECC is the necessary strategic choice for high-assurance data protection, particularly in light of future scaling needs and the increasing computational power available to adversaries [1][11].

3.2 Performance Efficiency: Bulk Data Confidentiality (AES vs. Hashing Throughput)
Symmetric encryption and cryptographic hashing play fundamentally different roles in a data collection system, reflected starkly in their throughput performance.

3.2.1 AES as the Throughput Engine
AES-256, particularly when implemented using the Galois/Counter Mode (GCM) and benefiting from hardware acceleration, demonstrated exceptional effectiveness for encrypting large datasets. In high-performance server environments utilizing specialized instructions like AES-NI, AES throughput rates can exceed 1800 MB/s [17]. Even in optimized software implementations without dedicated hardware acceleration, AES-256 (in CTR mode) frequently achieves speeds well over 100 MB/s [17]. This immense operational speed solidifies AES's role as the only feasible cryptographic technique for guaranteeing confidentiality over bulk data transfers or large data-at-rest stores within an enterprise HRM system [16].

3.2.2 SHA-256 Performance and the Integrity Bottleneck SHA-256 is vital for ensuring data integrity and generating digests for digital signatures [2]. However, software implementations of SHA-256 typically yield throughput ratesbetween 100 MB/s and 350 MB/s on comparable modern architectures [17].

The disparity between high-speed AES encryption (hardware accelerated) and moderate-speed SHA-256 hashing presents a crucial architectural challenge. In traditional "Encrypt-then-MAC" security models that use an unauthenticated encryption mode (e.g., AES-CBC) paired with a separate HMAC-SHA256 integrity check, the overall throughput rate of the security operation is effectively capped by the slower hashing function [6]. The integrity check becomes the performance bottleneck.

The adoption of the AES-GCM mode directly addresses this issue. GCM integrates the authenticity check (GHASH) into the encryption pipeline [15]. This combined AEAD operation often allows the integrity check to run in parallel with the encryption or leverage the same hardware acceleration capabilities as the AES core, preventing the cryptographic hashing step from imposing an unnecessary performance ceiling on the bulk data transfer system [15]. Consequently, adopting an AEAD mode is not merely a robust security practice, but a critical performance optimization necessary for scaling high-throughput data collection systems.

Table 3: Comparative Throughput Performance (MB/s) of Bulk Operations

Algorithm/Operation	Cryptographic Function	Typical Performance Range (MB/s) (Software/Hardware Accelerated)	Key Insight
AES-256 (Software Optimized)	Symmetric Encryption/Decryption	30 - 150 MB/s (Classic) [17]	High speed; practical baseline for software systems.
AES-256 (AES-NI/Hardware)	Symmetric Encryption/Decryption	600 - 1830+ MB/s (Hardware) [17]	Exceptional throughput; essential for enterprise scaling.
SHA-256	Cryptographic Hashing (Integrity)	100 - 350 MB/s (Software) [17]	Moderate speed; integrity checking can be a system bottleneck if not optimized via AEAD.[6]
Asymmetric Operations (RSA/ECC)	Key Exchange/Signing	Measured in operations per second (Ops/s) or ms latency (throughput negligible)	Unsuitable for bulk data encryption; dedicated to key management.[14]

Table 3 shows the difference between four algorithms based on the cryptographic function, and performance (mb/s) with its key insights. Based on the results, AES-256 (Software Optimized) is a classic practical baseline for software systems, AES-256 (AES-NI/Hardware) is essential for enterprise scaling, while SHA-256 that uses hashing has integrity checking, but it can be a system

bottleneck if not optimized by AED, then the Asymmetric Operations (RSA/ECC) is dedicate to key management, however, it is unsuitable for bulk data encryption. This table shows that Asymmetric Operations (RSA/ECC) is not recommended to be used when system has bulk operations or data.

3.3 Performance Efficiency: Transactional Key Exchange and Digital Signatures (RSA vs. ECC Latency)

While symmetric encryption drives throughput, asymmetric algorithms manage the initial trust establishment, making them latency-critical. High-frequency operations in HRM systems, such as user log-in, API token authorization, and secure connection handshakes, depend directly on the speed of key exchange and digital signing.

3.3.1 ECC Superiority in Latency

The evaluation confirms that ECC is significantly more effective than RSA at equivalent security levels for transactional tasks [16]. ECC's reduced key length minimizes the computational load required for complex mathematical operations like key generation, public key encryption of the session key, and signature verification [16]. For example, studies have consistently shown that ECC offers superior speed and lower computational overhead compared to RSA in establishing secure connections [1][14].

This latency advantage is crucial for improving the user experience and the responsiveness of transactional data collection systems. If the initial key exchange (e.g., in a TLS handshake) is delayed by the calculation time of high-modulus RSA keys, system performance degrades. ECC P-256 successfully provides the required 128-bit security while keeping the handshake overhead to a minimum [11].

3.3.2 Algorithm Trade-offs and System Context

The core finding of this evaluation is that the overall effectiveness of data security hinges on the optimized integration of these cryptographic types. The selection of a cryptographic method must be dictated by the function it performs within the system [2].

AES provides data confidentiality and is optimized for speed. ECC provides trust (key establishment and authentication) and is optimized for latency. SHA-256 provides integrity and non-repudiation and is optimized for assurance.

The use of a strong symmetric cipher like AES-256 would be rendered ineffective if it's secret key must be exchanged using a high-latency, resource-intensive asymmetric method like 3072-bit RSA, thus negating the speed advantage of AES. Conversely, attempting to use RSA or ECC for bulk data encryption would lead to unacceptable system degradation due to their negligible throughput rates [14].

3.4 Proposal of an Optimal Hybrid Cryptographic Model for HRM Systems

Based on the performance metrics and security analysis, the most effective solution for protecting HRM data collection systems is a Hybrid Cryptographic Model, which intelligently integrates the strengths of ECC, AES-GCM, and SHA-256. [1]. This framework provides robust security guarantees across the entire data lifecycle while ensuring operational efficiency and scalability [10].

3.4.1 Hybrid ECC-AES-GCM Framework Operation

1. Phase 1: Key Establishment (ECC Focus): The communication session begins with mutual authentication andsession key agreement using Elliptic Curve Diffie-Hellman (ECDH) based on Curve P-256. This process provides rapid, high-security session key establishment. Digital signatures (ECDSA using SHA-256) are employed to authenticate the server and client entities, ensuring non-repudiation for sensitive transactions [16].

2. Phase 2: Bulk Data Transfer (AES-GCM Focus): The short, ephemeral session key generated in Phase 1 is then used by AES-256 in Galois/Counter Mode (GCM). All subsequent bulk data (e.g., transmission of employee records, large data payloads) is encrypted using AES-GCM. This leverages the hardware acceleration features to achieve maximum throughput, fulfilling the core requirements for confidentiality, integrity, and authenticity simultaneously [1][15].

3. Phase 3: Data Integrity at Rest (SHA-256 Focus): For persistent integrity assurance, such as verifying the uncompromised state of database backups or archived transaction logs, SHA-256 is used independently to generate and store immutable cryptographic digests. This guarantees that any unauthorized modification to the stored data can be immediately detected [2].

This ECC-AES-GCM Hybrid Framework addresses all necessary security principles—confidentiality, integrity, authenticity, and non-repudiation—while prioritizing the highest possible speed for both high-latency transactional setup (ECC) and high-throughput data transfer (AES-GCM).

4. CONCLUSION

4.1 Summary of Evaluation Findings

This evaluation confirmed the necessity of a multifaceted, hybrid approach to effectively secure data collection systems. The effectiveness of AES, RSA/ECC, and SHA-256 depends entirely on their strategic application within the system architecture.

AES Effectiveness: AES-256, particularly when implemented with hardware acceleration (AES-NI) and

utilizing the Galois/Counter Mode (GCM), is the unequivocal choice for bulk data confidentiality and integrity. It provides high throughput (over 1 GB/s) that scales efficiently with modern enterprise demands [17].

Asymmetric Effectiveness: ECC (P-256) significantly outperforms RSA (3072-bit) in the efficiency required for key establishment and digital signature processes. ECC delivers the necessary 128-bit security equivalence while maintaining minimal computational overhead, thereby reducing system latency and improving responsiveness [7][11].

Hashing Effectiveness: SHA-256 provides the essential foundation for integrity assurance. Although slower than hardware-accelerated AES in raw throughput, its integration into the AES-GCM authenticated mode ensures that integrity checks do not impose a bottleneck on bulk data processing [15][6].

4.2 Recommendations for Implementation in Data Collection Systems

Based on the quantitative performance analysis and security trade-offs identified, the following technical recommendations are critical for implementing effective cryptographic protection in HRM and similar data collection systems:

Mandate Hybrid Cryptography: Data collection systems must adopt a Hybrid ECC-AES framework [1]. ECC P-256 or P-384 should be exclusively used for establishing secure session keys (ECDH) and providing entity authentication (ECDSA), capitalizing on its speed and resource efficiency for latency-critical interactions [10][16].

Standardize on AEAD: All data encryption, both in transit and at rest, must utilize AES-256 in an Authenticated Encryption with Associated Data (AEAD) mode, with GCM as the recommended standard [6]. This ensures simultaneous protection of confidentiality, integrity, and authenticity, which is a mandatory requirement for personal data security [5][15].

Exploit Hardware Acceleration: System deployment decisions should prioritize hardware platforms capable of utilizing specialized instruction sets (e.g., AES-NI) to maximize the operational effectiveness of AES and cryptographic hashing, ensuring high-volume scalability and optimized resource usage [16][17].

Prioritize Key Management: Regardless of the algorithm strength, the overarching effectiveness of the system relies on secure key management practices [7]. Robust key lifecycle management, including regular key rotation, strong access controls based on the principle of least privilege, and secure, isolated storage for master keys, is paramount to prevent system compromise [10].

4.3 Limitations and Future Research Directions

A limitation of this study is the reliance on synthesized performance benchmarks drawn from high-standard industry and academic sources, rather than conducting a single, controlled experiment across all variables and platfor006Ds. While this approach allows for comprehensive comparison based on validated data, real-world performance can be further influenced by specific platform microarchitectures, operating system schedulers, and compiler optimizations [17].

Future research should focus on two key areas:

Post-Quantum Cryptography (PQC) Evaluation: Given the emerging threat posed by quantum computing to current asymmetric algorithms like RSA and ECC [12], future work must evaluate the integration of NIST-selected PQC algorithms (e.g., CRYSTALS-Kyber for key encapsulation) into the established ECC-AES hybrid framework to assess their transitional performance overhead and resource compatibility [9].

Constrained Resource Environments: An investigation into the effectiveness of these cryptographic solutions, particularly ECC and lightweight AES implementations, within resource-constrained environments (e.g., IoT devices or mobile collection endpoints) is needed to assess performance impacts on energy consumption and memory usage [15][12].

REFERENCES

1. A. Anjali and L. C. Manikandan, “A study on cryptographic techniques,” Int. J. Sci. Res. Comput. Sci., Eng. Inf. Technol., 2020. [Online]. Available: https://doi.org/10.32628/CSEIT206453
2. A. Somasundaram and R. Challa, “Performance evaluation of AES, RSA and ECC in real-world applications,” 2024. [Online]. Available: ResearchGate.
3. D. Singarathnam, S. Ganesan, S. Pokhrel, and N. Somasiri, “Exploring cryptographic techniques for data security in resource-constrained wireless sensor networks: Performance evaluation and considerations,” in Proc. 10th Int. Conf. Softw. Eng. Comput. Syst. (ICSECS), 2023. [Online]. Available: https://doi.org/10.1109/ICSECS58457.2023.10256341
4. D. Yadav and A. Saxena, “A comparative study on the performance and the security of RSA and ECC algorithm,” Int. J. Comput. Appl., vol. 174, no. 16, 2020.
5. E. K. Zaid and R. Ahmed, “Performance analysis of two famous cryptographic algorithms on mixed data,” J. Comput. Sci., vol. 19, no. 6, 2022. [Online]. Available: https://doi.org/10.3844/jcssp.2023.694.706
6. Federal Office for Information Security (BSI), Technical guideline TR-02102-1: Cryptographic mechanisms: Recommendations and key lengths. 2025.

7. H. Yeoh, M. Kwek, and S. Wong, "Analyzing cryptographic algorithm efficiency with in graph-based encryption models," Front. Comput. Sci., 2025. [Online]. Available: https://doi.org/10.3389/fcomp.2025.1630222
8. K. Bhanot, "Comparative analysis of encryption algorithms," Int. J. Adv. Acad. Res., vol. 8, no. 11, 2022.
9. M. A. Hossain, M. B. Hossain, M. S. Uddin, and S. M. Imtiaz, "Performance analysis of different cryptography algorithms," Int. J. Comput. Trends Technol., vol. 47, no. 4, 2017. [Online]. Available: https://doi.org/10.14445/22312803/IJCTT-V47P106
10. M. C. Wien, F. O. Catak, M. Kuzlu, and U. Cali, "Neural networks meet elliptic curve cryptography: A novel approach to secure communication," in Proc. 2024 IEEE Virtual Conf. Commun. (VCC), NY, USA, 2024, pp. 1–6. [Online]. Available: https://doi.org/10.1109/VCC63113.2024.10914360
11. M. Marwaha, R. Bedi, A. Singh, and T. Singh, "Comparative analysis of cryptographic algorithms," Int. J. Adv. Eng. Technol., vol. 4, no. 2, pp. 16–18, 2013.
12. M. Mushtaq, S. Jamel, A. Disina, Z. Pindar, N. Shakir, and M. Mat Deris, "A survey on the cryptographic encryption algorithms," Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 11, pp. 333–344, 2017. [Online]. Available: https://doi.org/10.14569/IJACSA.2017.081141
13. M. S. Yousefpoor and H. Barati, "Dynamic key management algorithms in wireless sensor networks: A survey," Comput. Commun., vol. 134, pp. 52–69, 2019. [Online]. Available: https://doi.org/10.1016/j.comcom.2018.11.005
14. N. A. W. kbean and S. B. Sadkhan, "Cryptography techniques within SCADA system-A survey," in Proc.2020 3rd Int. Conf. Eng. Technol. Its Appl. (IICETA), Najaf, Iraq, 2020, pp. 89–94. [Online]. Available: https://doi.org/10.1109/IICETA50496.2020.9318832
15. National Institute of Standards and Technology (NIST), Draft SP 800-78-4, Cryptographic algorithms and key sizes for PIV. 2024.
16. National Institute of Standards and Technology (NIST), Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) and GMAC. 2001.
17. R. Najm, "Enhancing the Advanced Encryption Standard (AES) key generation using SHA-256 for secure data in cloud computing," Int. J. Sci. Appl. Inf. Technol., vol. 8,pp. 120–126, 2019. [Online]. Available: https://doi.org/10.30534/ijsait/2019/188620198
18. The BearSSL Project, "Cryptographic benchmark data." [Online]. Available: https://www.bearssl.org/speed.html
19. Y. Alemami, M. A. Mohamed, and S. Atiewi, "Advanced approach for encryption using advanced encryption standard with chaotic map," Int. J. Electr. Comput. Eng., vol. 13, pp. 1708–1723, 2023. [Online]. Available: https://doi.org/10.11591/ijece.v13i2.pp1708-1723
20. Y. Alemami, M. A. Mohamed, and S. Atiewi, "Research on various cryptography techniques," Int. J. Recent Technol. Eng., vol. 8, 2019. [Online]. Available: https://doi.org/10.35940/ijrte.B1069.0782S39