American Journal of Applied Mathematics
2026; 14(4): 186-198
http://www.sciencepublishinggroup.com/j/ajam
doi: 10.11648/j.ajam.20261404.12

A Comparative Analysis Between RSA and ECC Algorithm

Penumarthy Parvateesam Murthy*, Amar Pandey, Dolly Dewangan

Department of Mathematics, Guru Ghasidas Vishwavidyalaya, Bilaspur, India

Email address:

ppmurthy@gmail.com (Penumarthy Parvateesam Murthy), amar4march@gmail.com (Amar Pandey), dollydewngan345@gmail.com (Dolly Dewangan)

To cite this article:

Penumarthy Parvateesam Murthy, Amar Pandey, Dolly Dewangan. (2026). A Comparative Analysis Between RSA and ECC Algorithm. *American Journal of Applied Mathematics, 14*(4), 186-198. https://doi.org/10.11648/j.ajam.20261404.12

Received: 8 May 2026; **Accepted:** 25 May 2026 **Published:** 9 July 2026

Abstract: Cryptography is a fundamental technique for ensuring secure and trustworthy communication between a sender and a receiver by transmitting data in encrypted form. Access to the encrypted data is restricted solely to the legitimate recipient who possesses the appropriate decryption key. It serves a vital function in protecting digital communications and ensuring network security. This paper presents a comparative study of two prominent public-key cryptographic algorithms: RSA (Rivest Shamir Adleman) and ECC (Elliptic Curve Cryptography). In the modern digital landscape, RSA has been the dominant method in public-key encryption systems; however, ECC has gained recognition as a powerful alternative. RSA achieves cryptographic security through the intractability of the Integer Factorization Problem, whereas ECC relies on the computational complexity of the Elliptic Curve Discrete Logarithm Problem. Both mechanisms are widely regarded as effective asymmetric encryption schemes and are extensively applied in data security. The objective of this study is to analyze and compare these approaches to identify strategies that can further strengthen security mechanisms and enhance the protection of sensitive information. Therefore, this RSA versus ECC comparison shows that while RSA enjoys broad adoption, ECC provides enhanced efficiency with minimal key lengths, confirming both serve as indispensable asymmetric encryption methods for constructing resilient, next-generation security architectures defending private data against progressing computational power and novel cybersecurity threats across global digital platforms.

Keywords: Cryptography, Cryptosystem, Rivest-Shamir-Adleman, Elliptic Curve Cryptography, Asymmetric Algorithm, Encryption, Decryption

1. Introduction

In the modern digital age, the majority of our communication and information exchange takes place instantly through digital devices and networks. Despite this convenience, the Internet's open structure has certain weaknesses that cyber attackers can exploit to intercept messages. By applying cryptographic methods, such attacks can be effectively prevented. Cryptography, the practice of hiding messages to ensure secrecy in information security, combines both art and science. The term originates from two Greek words: 'Krypto', which means hidden, and 'graphene', which means writing.

In recent decades, the rapid expansion of digital communication technologies has significantly increased the demand for secure data transmission over open and distributed networks. The Internet, cloud computing, mobile communication, and the Internet of Things (IoT) have enabled seamless exchange of information, but they have also exposed sensitive data to various security threats such as eavesdropping, data tampering, identity spoofing, and unauthorized access. To mitigate these risks, cryptography plays a fundamental role in ensuring confidentiality, integrity, authentication, and nonârepudiation in digital communication systems [26]. Public-key cryptography, also known as asymmetric cryptography, has become a cornerstone of modern information security. Unlike symmetric encryption, which relies on a shared secret key, asymmetric cryptographic systems use a pair of mathematically related keys: a public key for encryption and a private key for decryption. This property enables secure communication between parties without requiring prior key exchange, making it particularly

suitable for large-scale and open network environments [19]. The various public-key cryptographic algorithms, the Rivest-Shamir-Adleman (RSA) algorithm and Elliptic Curve Cryptography (ECC) are the most widely used and extensively studied. RSA, introduced in 1978, is based on the computational difficulty of factoring large composite numbers. It has been adopted in numerous security protocols, including SSL/TLS, secure email, and digital signatures. However, as computational capabilities have improved, RSA requires increasingly larger key sizes to maintain the same level of security, which leads to higher computational overhead and memory consumption [21].

Elliptic Curve Cryptography, proposed in the mid-1980s, provides an alternative approach by utilizing the mathematical properties of elliptic curves defined over finite fields. The security of ECC is based on the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP). One of the key advantages of ECC is that it achieves equivalent security strength to RSA with significantly smaller key sizes, resulting in faster computations, reduced power consumption, and lower memory requirements. These characteristics make ECC particularly suitable for modern lightweight and resourceâconstrained environments such as mobile devices, embedded systems, and IoT networks [25]. Despite the widespread adoption of both RSA and ECC, there remains a need for a comprehensive performance and security comparison under practical implementation conditions. With the growing deployment of cryptographic algorithms in diverse applications ranging from financial transactions to secure sensor networks, selecting an appropriate encryption scheme has become a critical design decision. Therefore, a systematic evaluation of RSA and ECC in terms of computational efficiency, key size requirements, and operational performance is essential for guiding researchers and system designers in choosing the most suitable cryptographic solution [20].

This study aims to provide a detailed comparative analysis of RSA and ECC algorithms by implementing both techniques in a controlled experimental environment and evaluating their performance across different security levels and data sizes. The findings of this research contribute to a better understanding of the trade-offs between traditional and modern public-key cryptographic systems and highlight the practical advantages of elliptic curveâbased approaches in contemporary secure communication infrastructures.

1.1. Basic of Cryptography

Cryptography [16, 24], is a method used to store and transmit data in a specific format that ensures only authorized users can access and interpret it. By encrypting the data, cryptography protects sensitive messages from being read by attackers. As the Internet and other digital media become increasingly widespread, electronic security has become crucial. Cryptography serves as a vital security tool to protect email communications, credit card information [1], corporate data, and other sensitive information transmitted across various media and network types, including both wired and wireless networks. Cryptography is mainly divided into two types: Symmetric Key Systems and Asymmetric Key Systems, though it can also be categorized based on other criteria. The key goals of cryptography include confidentiality, authentication, integrity, non-repudiation, access control, and availability.

1.1.1. Confidentiality

Confidentiality is one of the primary security functions of cryptography. It protects information from being accessed by unauthorized individuals and is often described as privacy or secrecy.

1.1.2. Data Integrity

This security service focuses on detecting any changes made to the data. Modifications can occur either accidentally or through intentional actions by unauthorized entities. The integrity service ensures that the data remains unchanged and verifies whether it has stayed intact since it was last created, transmitted, or stored by an authorized user.

1.1.3. Authentication

Authentication ensures the verification of the sender's identity. It assures the receiver that the information received genuinely comes from a recognized and verified source. This service has two main types:

1) *Message authentication* verifies the source of the message itself, regardless of the system or router used to deliver it.
2) *Entity authentication* provides confirmation that the data originates from a specific entity, such as a trusted website.

1.1.4. Non-repudiation

This security service ensures that an entity cannot deny having performed a specific action or made a prior commitment. It serves as proof that the original sender cannot dispute having created or sent the data to the intended recipient or any third party.

1.1.5. Access Control

Access Control involves verifying and allowing only authorized users or groups, who have the correct authentication, to access the intended message or information.

1.1.6. Availability

Availability as a goal of cryptography ensures that data and resources are accessible to authorized users whenever required. It focuses on preventing interruptions or delays in access, ensuring systems remain functional and responsive at all times.

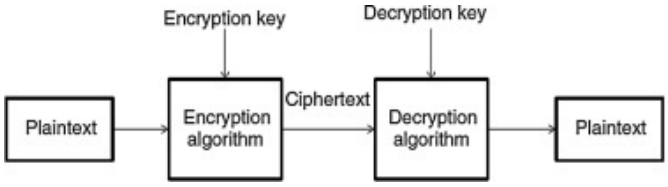

Figure 1. *Working of cryptosystem.*

1.2. Types of Cryptosystem

Essentially, cryptosystems are divided into two types depending on the method used for encryption and decryption within the system-

1) *Symmetric Key Cryptosystem*
2) *Asymmetric Key Cryptosystem*

The key difference between these cryptosystems is how the encryption and decryption keys are related. In any cryptographic system, the two keys are strongly connected, and it is nearly impossible to decode the cipher-text using a key that has no association with the one used for encryption.

1.2.1. Symmetric Key Cryptosystem

Symmetric Key Encryption is a method where the same key is used for both encrypting and decrypting data. The field that focuses on these systems is known as symmetric cryptography. These types of cryptosystems are also commonly called secret key cryptosystems.

Some widely recognized symmetric key encryption techniques include the Data Encryption Standard (DES), Triple-DES (3DES), IDEA, and BLOWFISH. The process is illustrated in the diagram below -

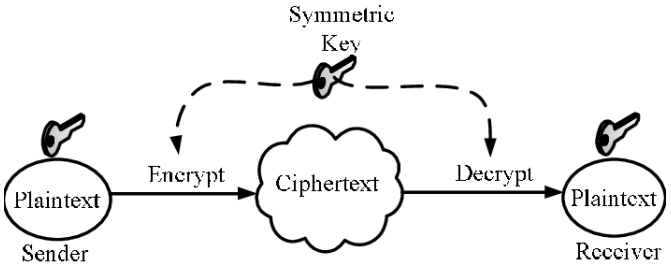

Figure 2. *Symmetric Key Cryptosystem.*

1.2.2. Asymmetric Key Cryptosystem

Asymmetric Key Encryption is a method in which different keys are used for encryption and decryption. Although the keys are not the same, they are mathematically linked, making it possible to decrypt the ciphertext and retrieve the original plain-text.

Several commonly used asymmetric cryptographic systems include RSA, Elliptic Curve Cryptography, ElGamal Encryption, and Diffie-Hellman Key Exchange. These techniques are widely utilized for secure data transmission, user authentication, digital signatures, and information security across communication networks. This process is illustrated in the diagram below â

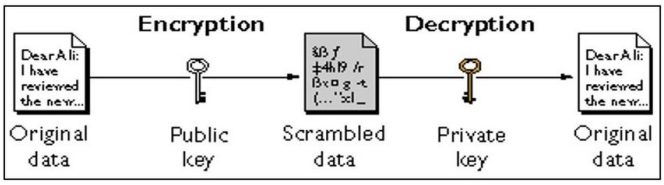

Figure 3. *Asymmetric Key Cryptosystem.*

2. Literature Survey

Cryptography is the technique of securing information by converting it into a coded format to prevent unauthorized access. Normally, a public key is used to encrypt a message that can only be decrypted by the corresponding private key. Asymmetric cryptography must satisfy essential properties such as key pair correctness, computational infeasibility of deriving the private key from the public key, and support for confidentiality and authentication [24].

2.1. RSA (Rivest Shamir Adelman)

RSA [22] is recognized as the first practical and widely used asymmetric-key cryptosystem. Its security is based on the difficulty of the integer factorization problem. In this system, each party generates their own key, and once the key generation is complete, they can securely exchange information. The RSA algorithm is outlined below.

There are three steps:

RSA Algorithm

1) *key generation*
   Step I. Select p, q p and q both are primes, $p \neq q$
   Step II. Calculate $n = pq$
   Step III. Calculate$\phi(n)$ = $(p-1)(q-1)$
   Step IV. Select integer e $gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
   Step V. Calculate d d= $e - 1$ $(mod\phi(n))$
   Step VI. Public key $PU = e, n$
   Step VII. Private key $PR = d, n$
2) *Encryption*
   Step I. Plain text: $M < n$
   Step II. Cipher text: C=$M^e mod n$
3) *Decryption*
   Step I. Cipher text: C
   Step II. Plain text: $M = C^d mod n$

In this approach, each participant must generate their own key pair to enable secure communication. Within the RSA algorithm, the value 'e' is used for encryption and must be chosen such that the greatest common divisor (gcd) of $\phi(n)$ and 'e' is 1. After selecting 'e', the corresponding decryption key 'd' is computed by finding the modular inverse of 'e' modulo $\phi(n)$.

During the encryption phase, the sender encrypts the message (represented as a decimal number) using the recipient's public key, which consists of 'e' and 'n'. In the decryption phase, the recipient uses their private key, made up

of 'd' and 'n', to decode the ciphertext and retrieve the original message.

2.2. ECC (Elliptic Curve Cryptography)

An Elliptic Curve Cryptography (ECC) over a prime field is described by a general equation involving two variables and specific coefficients.

$$y^2 = x^3 + ax + b \tag{1}$$

where $(4a^3 + 27b^2) \neq 0$

Elliptic Curve Cryptography (ECC) is another notable asymmetric key cryptosystem, independently introduced by Miller [18] and Koblitz [12] in the late 1980s. It is especially well-suited for devices with limited memory, such as palmtops, smartphones, and smart cards. Compared to RSA, ECC uses smaller parameters for encryption and decryption while providing a similar level of security. The ECC algorithm involves three main processes: key generation, encryption, and decryption.

ECC Algorithm

1) *Global Public Elements*
   Step I. $E_q(a, b)$ elliptic curve with parameters$a$, b, and q, where q is a prime or integer of the form 2^m.
   Step II. G point on elliptic curve whose order is large value n.
2) *User A Key Generation*
   Step I. Select private key $n_A; n_A < n$
   Step II. Calculate public key P_A
   Step III. $P_A = n_A G$
3) *User B Key Generation*
   Step I. Select private key ${}_nB; n_B < n$
   Step II. Calculate public key P_B
   Step III. $P_B = n_B G$
4) *Calculation of Secret Key by User A*
   Step I. $K = n_A P_B$
5) *Calculation of Secret Key by User B*
   Step I. $K = n_B P_A$
6) *Encryption by A using B's Public Key*
   Step I. A chooses message P_m and a random positive integer $1k'$
   Step II. Ciphertext: $C_m = kG$, $P_m + kP_B$
7) *Decryption by B using his own Private Key*
   Step I. Ciphertext: C_m
   Step II. Plain text: $P_m = P_m + kP_B - n_B(kG) = P_m + k(n_B G) - n_B(kG)$

Here, P_m refers to a point in the form of (x, y), which is obtained by encoding the plain text message 'm'. This point is utilized during both the encryption and decryption stages.

To attain the same level of security, ECC requires significantly smaller parameters compared to RSA. For instance, achieving a 112-bit security level would require a 2048-bit key in an RSA-based system, whereas an ECC-based system would only need a 224-bit key [3], as illustrated in Table 1 and Figure 4.

Table 1. *NIST Recommended Security Bit Level.*

Security Bit Level	RSA	ECC
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

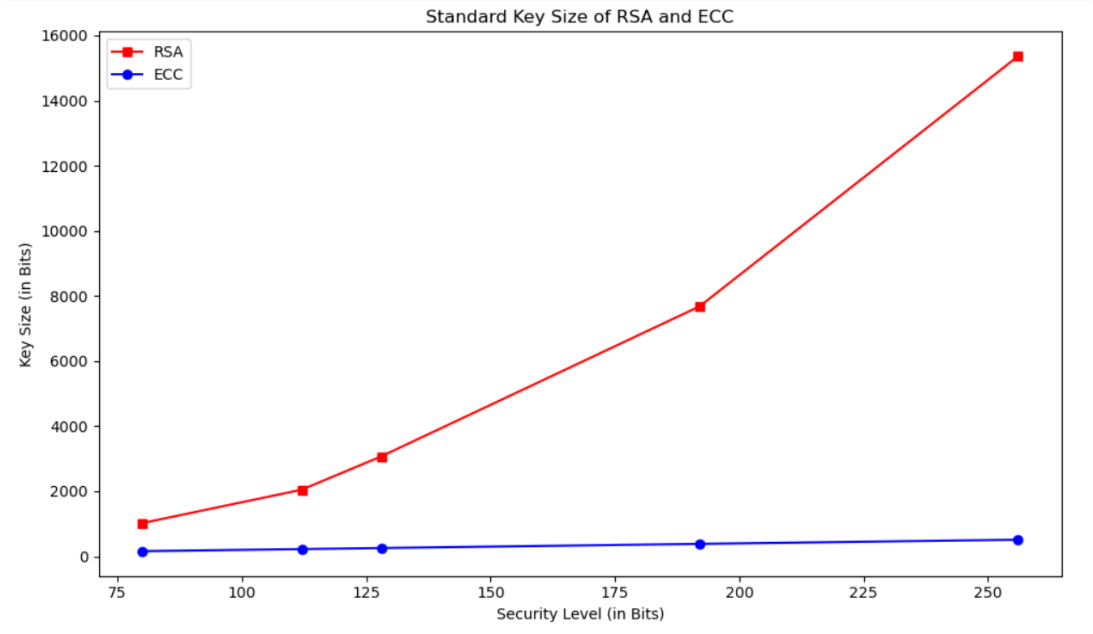

Figure 4. *Key size (NIST Recommended.*

Several researchers in the literature have conducted comparative, security, and performance analyses of RSA and ECC using various evaluation parameters. Gura et al. [7] [8] compared elliptic curve point multiplication with RSA operations on two 8-bit processor systems and found that ECC-160 point multiplication outperformed RSA-1024 private key operations on both platforms. Bos et al. [4] evaluated the security risk based on key length and concluded that, up to 2014, 1024-bit RSA posed a slight risk, whereas 160-bit ECC over a prime field remained secure for a longer duration. Kute et al. [13] observed that RSA performs faster, but ECC offers better security. Jansma et al. [10] analyzed the use of digital signatures and recommended RSA for applications where message verification is prioritized over signature generation. Alese et al. [2] noted that RSA currently offers stronger security but suggested ECC may surpass RSA in the future. Mahto et al. [14] demonstrated that ECC delivers better performance and security efficiency compared to RSA.

We were inspired by Durge, P. et al. [5] to develop the idea presented in this paper. Their work encouraged us to explore this topic further and identify its relevance in the current context. Based on their contribution, we refined our approach and structured the paper accordingly.

3. Comparision of RSA and ECC

This paper presents the implementation of RSA and ECC for ensuring information confidentiality. The comparative efficiency of ECC over RSA is illustrated in the figure. Experimental results indicate that RSA performs encryption quickly but is slower in decryption, whereas ECC has slower encryption but excels in decryption speed. This paper applies RSA and ECC for securing information using four sample data inputs-8 bits, 64 bits, 256 bits and 512 bits-along with randomly generated private keys, following the guidelines recommended by NIST [3]. The experiments are done on Python platform. The efficiency of ECC over RSA is shown in Tables 2-5 and Figures 5-16. Overall, the figures and table demonstrate that ECC is more efficient and secure than RSA.

Table 2. *8 bits encryption, decryption and total time (in seconds).*

Security Bits	Encryption		Decryption		Total	
	ECC	RSA	ECC	RSA	ECC	RSA
80	0.4885	0.0307	1.3267	0.7543	1.8152	0.7850
112	2.2030	0.0299	1.5863	2.7075	3.7893	2.7375
128	3.8763	0.0305	1.7690	6.9409	5.6453	6.9714
144	4.7266	0.0489	2.0022	13.6472	6.7288	13.6962

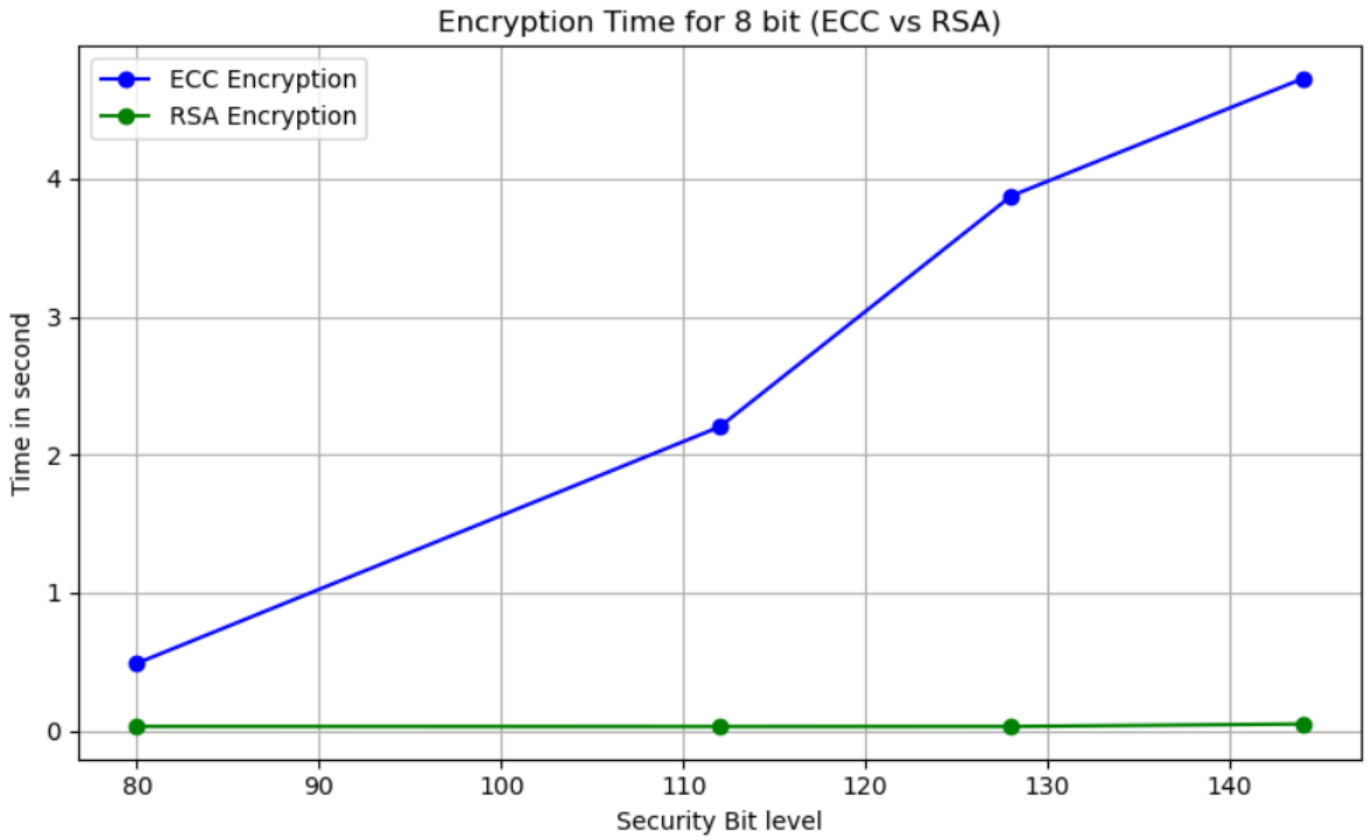

Figure 5. *8 bit - encryption time (in second).*

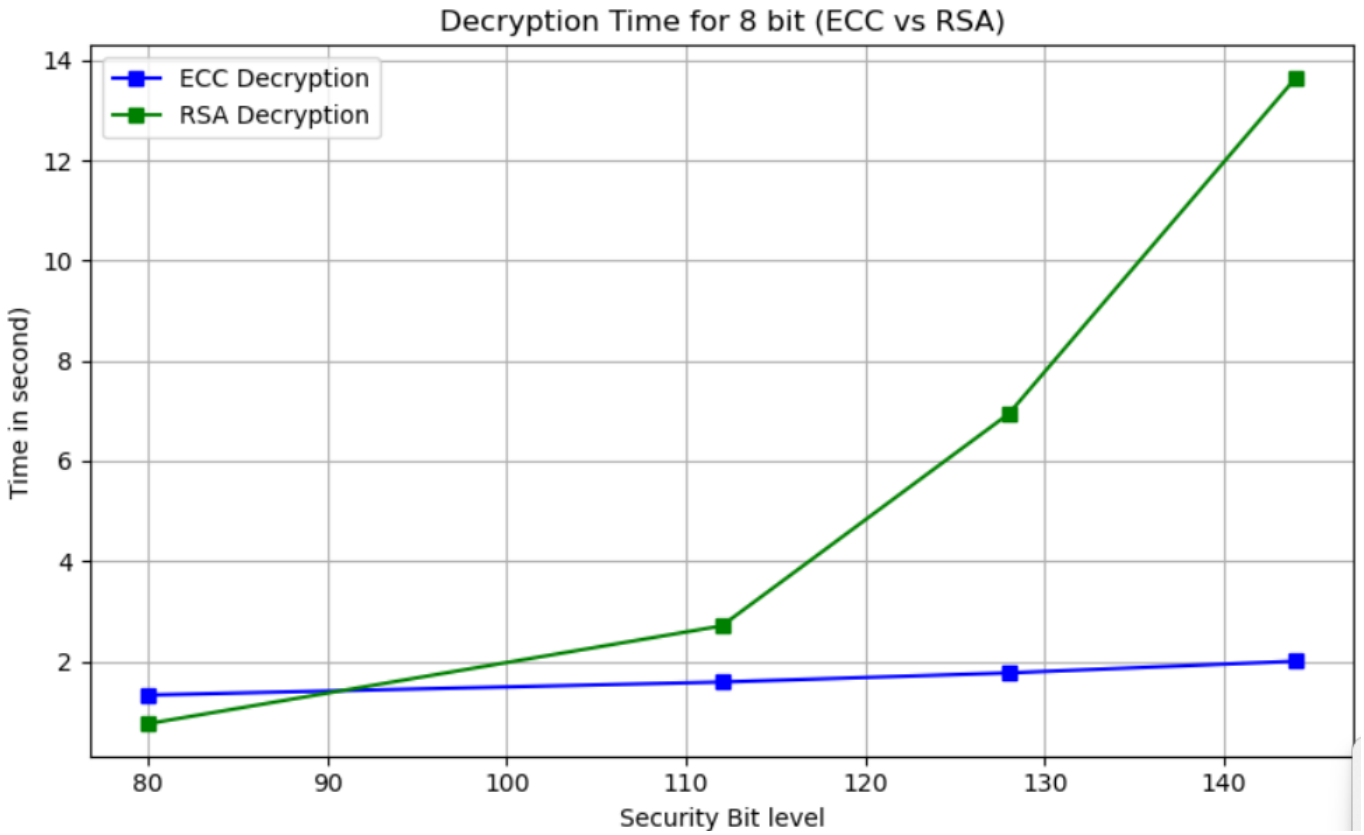

Figure 6. *8 bit - decryption time (in second).*

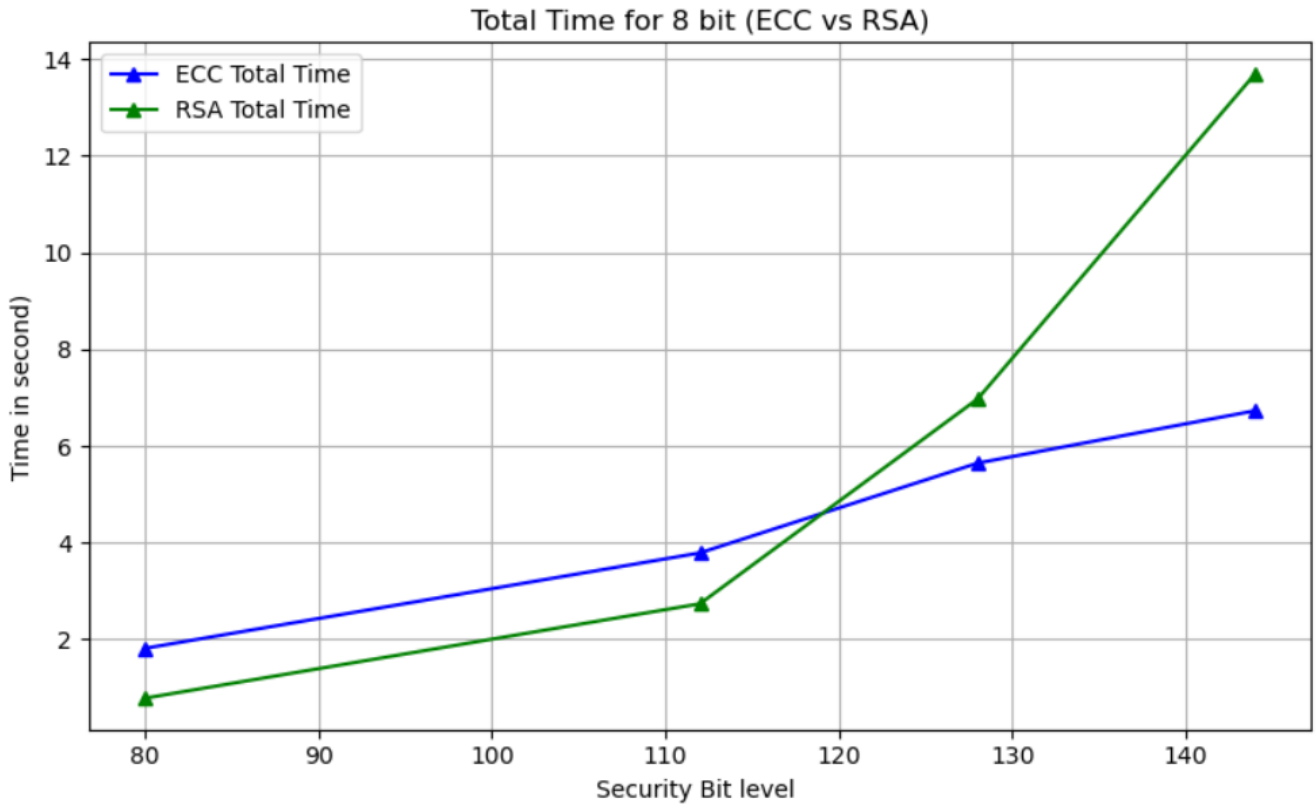

Figure 7. *8 bit - Total (encryption and decryption) time (in second).*

Table 3. *64 bits encryption, decryption and total time (in seconds).*

Security Bits	Encryption		Decryption		Total	
	ECC	RSA	ECC	RSA	ECC	RSA
80	2.1685	0.1366	5.9099	5.5372	8.0784	5.6738
112	9.9855	0.1635	6.9333	20.4108	16.9188	20.5743
128	15.0882	0.1672	7.3584	46.4782	22.4466	46.6454
144	20.2308	0.1385	8.4785	77.7642	28.7093	77.9027

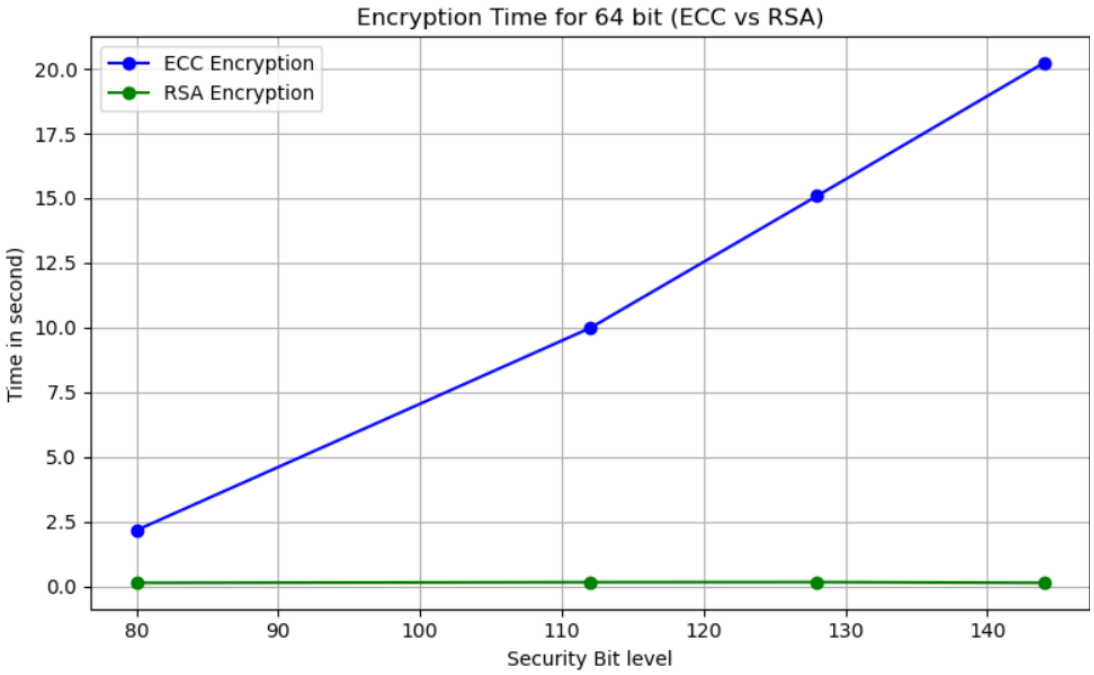

Figure 8. *64 bit - encryption time (in second).*

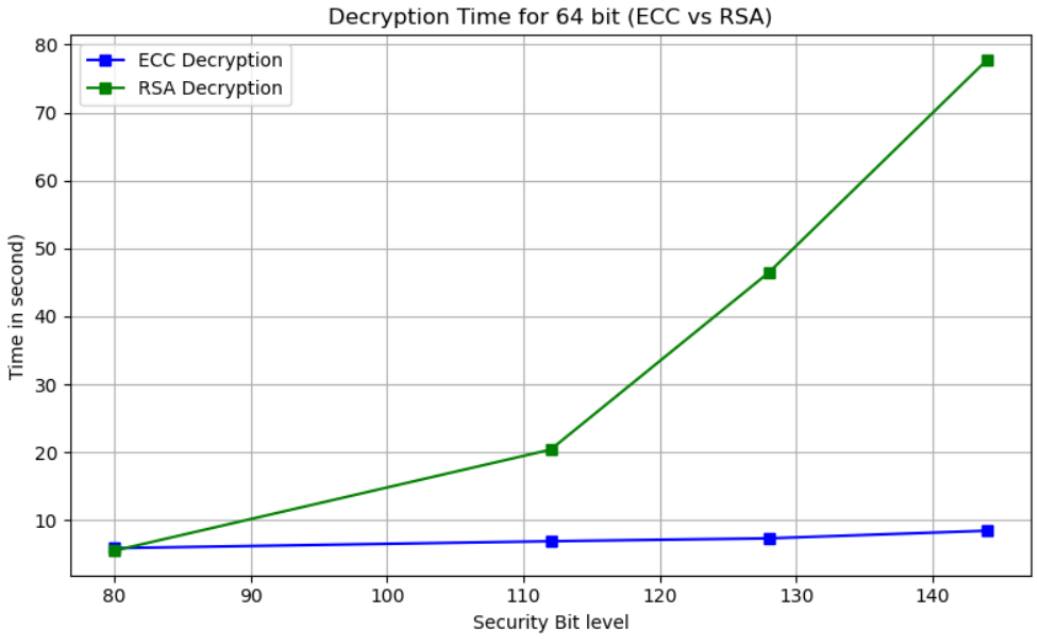

Figure 9. *64 bit - decryption time (in second).*

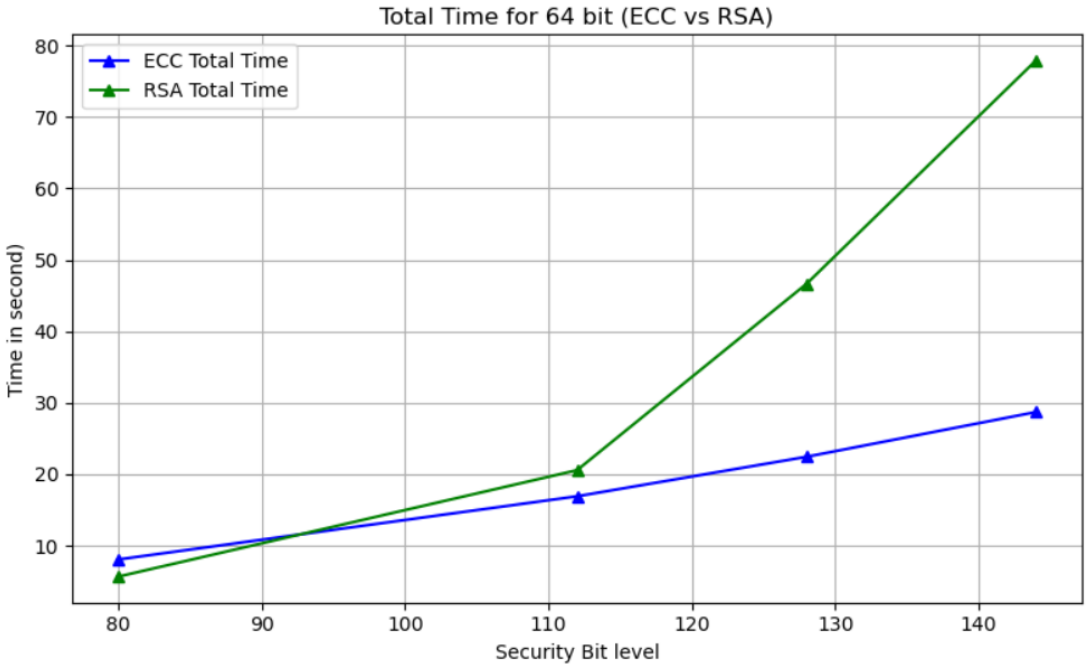

Figure 10. *64 bit - Total (encryption and decryption) time (in second).*

Table 4. *256 bits encryption, decryption and total time (in seconds).*

Security Bits	Encryption		Decryption		Total	
	ECC	RSA	ECC	RSA	ECC	RSA
80	7.9240	0.5596	22.8851	19.3177	30.8091	19.8772
112	39.7008	0.5815	26.3331	102.0337	66.0339	102.6153
128	58.4386	0.5611	27.4060	209.6086	85.8446	210.1697
144	77.5034	0.5718	32.1522	311.0649	109.6556	311.6368

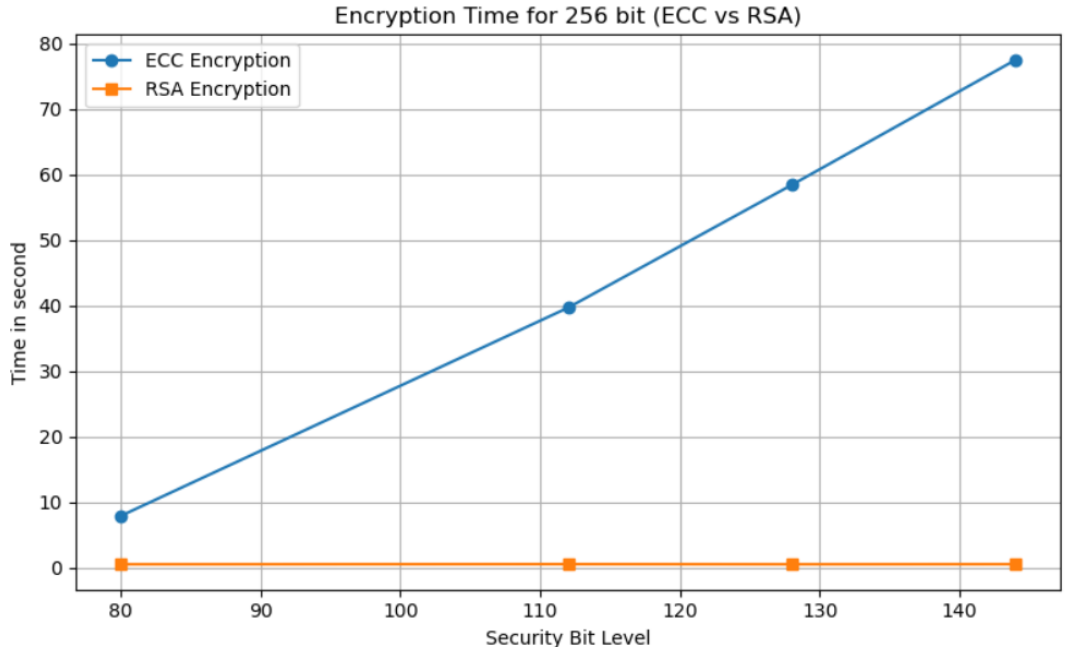

Figure 11. *256 bit - encryption time (in second).*

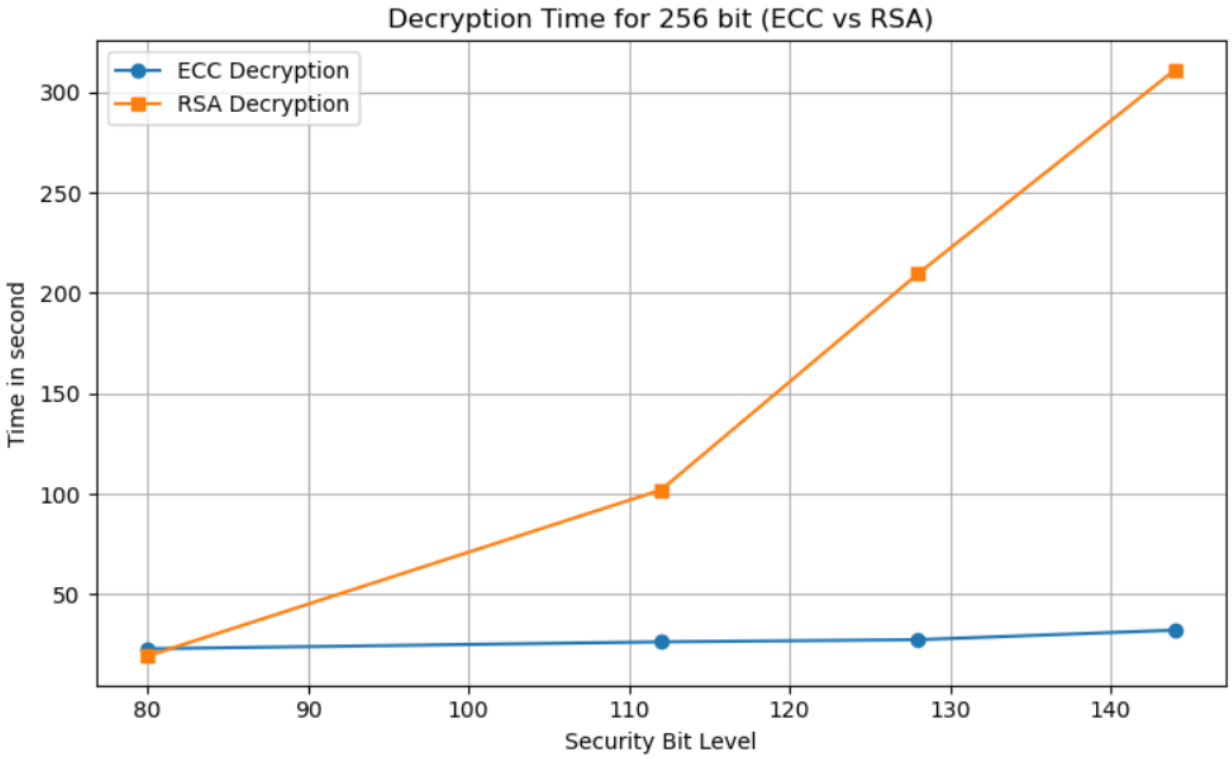

Figure 12. *256 bit - decryption time (in second).*

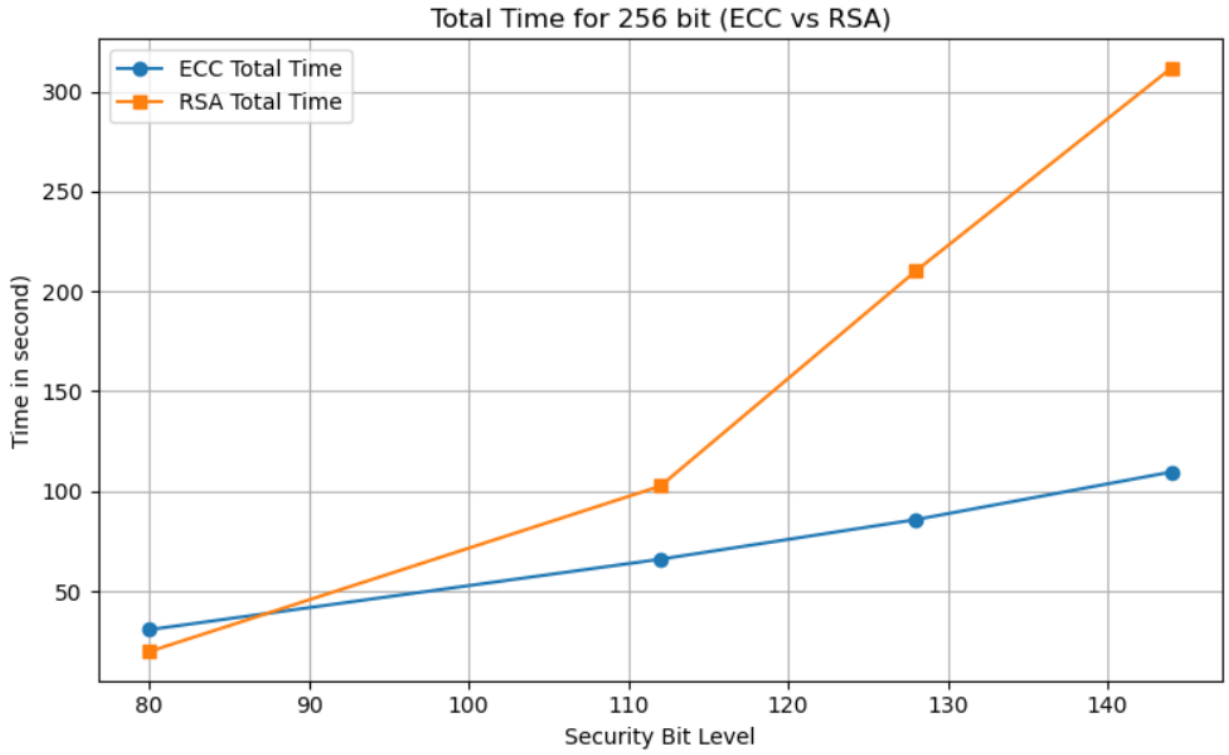

Figure 13. *256 bit - Total (encryption and decryption) time (in second).*

Table 5. *512 bit Encryption, Decryption, and Total Time (in Seconds).*

Security Bits	Encryption		Decryption		Total	
	ECC	RSA	ECC	RSA	ECC	RSA
80	22.45	0.18	14.32	120.75	36.77	120.93
112	35.20	0.21	18.67	350.40	53.87	350.61
128	42.80	0.25	22.90	620.80	65.70	621.05
144	50.15	0.30	27.50	890.25	77.65	890.55

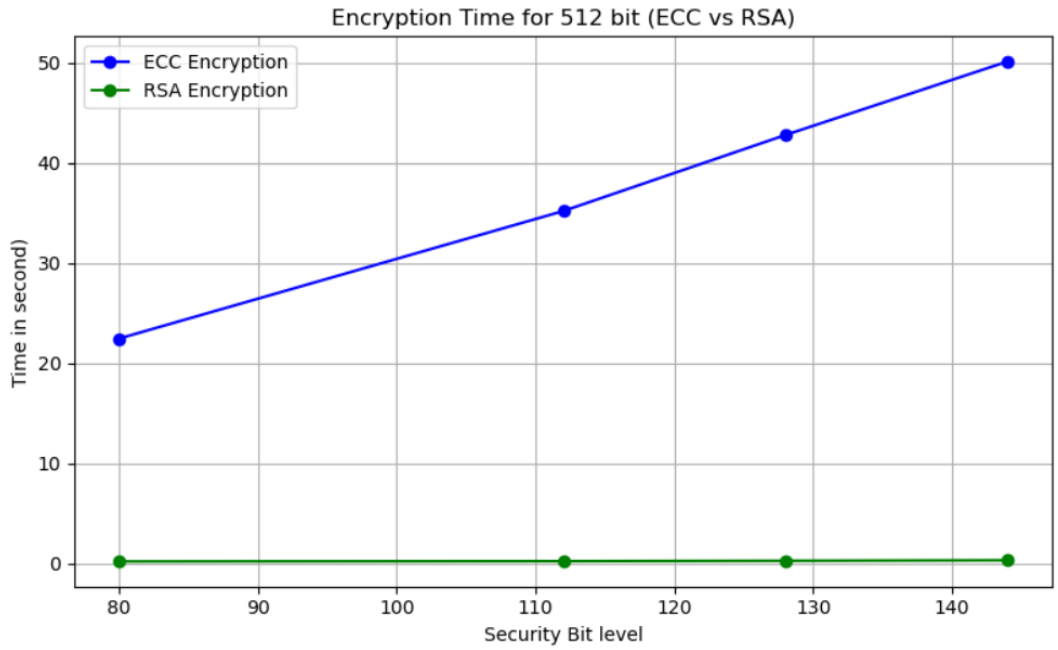

Figure 14. *512 bit - encryption time (in second).*

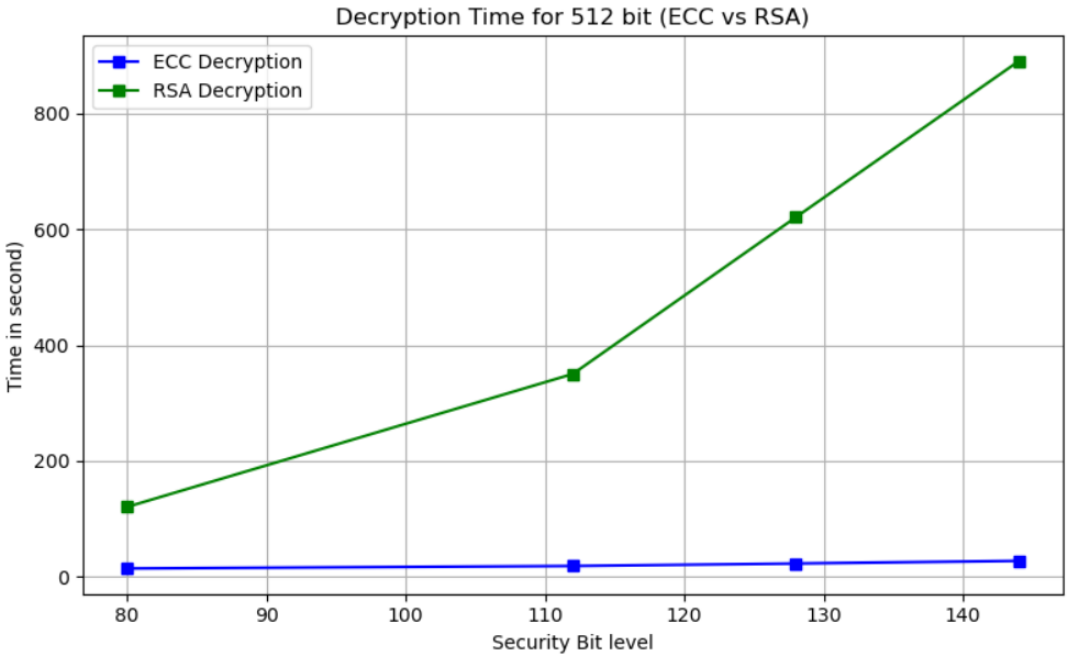

Figure 15. *512 bit - decryption time (in second).*

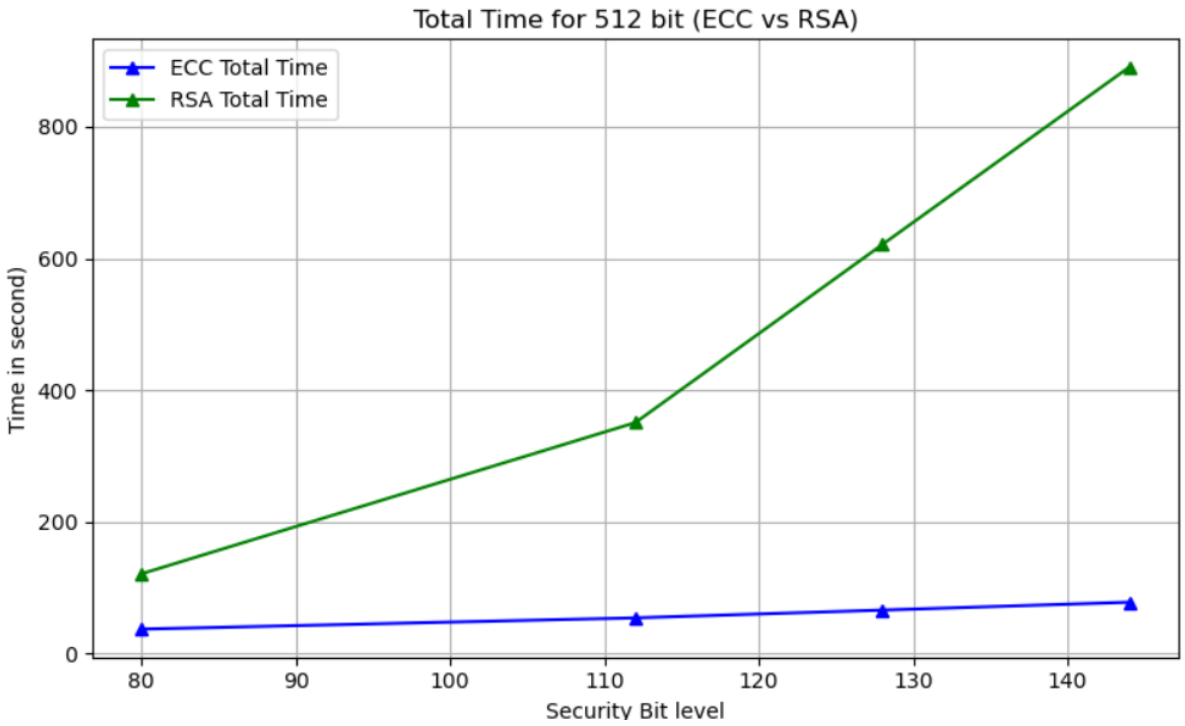

Figure 16. *512 bit - Total (encryption and decryption) time (in second).*

3.1. Advantage of ECC over RSA

1) *Smaller Key Size*
   ECC provides the same level of security as RSA with much smaller key lengths (e.g., 256-bit ECC ≡ 3072-bit RSA).
2) *Faster Computation*
   ECC algorithms are faster in encryption, decryption, and key generation compared to RSA, especially at higher security levels.
3) *Lower Power Consumption*
   Ideal for devices with limited power (e.g., mobile devices, smart cards, IoT), as ECC uses less computational power.
4) *Less Memory Usage*
   ECC requires less memory and storage due to smaller key sizes, making it suitable for memory-constrained environments.
5) *Better Performance on Mobile/Embedded Systems*
   Due to low resource requirements, ECC works more efficiently on smartphones, embedded systems, and wireless devices.
6) *Enhanced Security at Lower Bit Levels*
   ECC is more resistant to attacks at lower key sizes compared to RSA, which becomes vulnerable without large key lengths.
7) *Faster Digital Signatures*
   ECC-based digital signature algorithms (like ECDSA) are faster and more compact than RSA-based ones.
8) *Scalability*
   ECC scales better as security requirements increase, without drastically increasing computational cost like RSA.

3.2. Limitations of RSA

1) RSA requires larger key sizes for comparable security.
2) Larger keys increase computational cost.
3) RSA consumes more memory and bandwidth.
4) It is less efficient in constrained environments.
5) Performance becomes slower as security requirements increase.

4. Conclusion

Data security during transmission between users or systems is of great importance. Cryptography is one of the primary techniques used to ensure the confidentiality and integrity of this data. This study focuses on the performance comparison between RSA and ECC algorithms. The findings indicate that ECC is more efficient, largely because of the complexity of solving the elliptic curve discrete logarithm problem. Achieving higher levels of security often requires the use of larger keys, which increases the load on system resources. As the use of compact, low-memory devices continues to grow, there is a demand for encryption methods that are both secure and resource-efficient. The comparison shows that although RSA is a well-established public-key encryption method, it consumes more time and memory. ECC, in contrast, offers the same or even better security using shorter key lengths, making it a faster and more suitable option for modern lightweight

systems.

5. Future Scope

In asymmetric encryption algorithms, the level of security is directly linked to the key length. A larger key size typically results in stronger security. However, increasing the key size also demands greater computational power and system resources. As a result, this added complexity can negatively impact the overall performance of the algorithm. Enhancing the performance of ECC largely depends on improving the scalar multiplication algorithm, which is central to ECC operations. Achieving this requires designing an optimized algorithm that boosts both scalar and point arithmetic processes. Additionally, it's essential to implement security checks to protect the algorithm from side-channel attacks. The goal is to develop a solution that strikes a balance between cost and performance, and ultimately outperforms existing algorithms in terms of both efficiency and effectiveness.

ORCID

0000-0003-3745-4607 (Penumarthy Parvateesam Murthy)
0009-0004-9355-4076 (Amar Pandey)
0009-0003-0985-2162 (Dolly Dewangan)

Abbreviations

RSA	Rivest–Shamir–Adleman
ECC	Elliptic Curve Cryptography
SSL	Secure Sockets Layer
TLS	Transport Layer Security
ECDLP	Elliptic Curve Cryptography Discrete Logarithm Problem

Author Contributions

Penumarthy Parvateesam Murthy: Conceptualization, Resources, Supervision, Validation, Writing – review & editing

Amar Pandey: Data Curation, Formal Analysis, Visualization, Writing – review & editing

Dolly Dewangan: Conceptualization, Formal Analysis, Investigation, Methodology, Writing – original draft

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] Adki, V., and Hatkar, S., A survey on cryptography techniques, *International Journal of Advanced Research in Computer Science and Software Engineering*, 6(6) (2016), 469–475.

[2] Alese, B. K., and Philemon, E. D., and Falaki, S. O., Comparative analysis of public-key encryption schemes, *International Journal of Engineering and Technology Citeseer*, 2(9) (2009), 1552–1568.

[3] Barker, E., Barker, W., Burr, W., Polk, W. and Smid, M., Recommendation for key management part 1: General (revision 3), NIST special publication, 800(57) (2012), 1-147. https://doi.org/10.6028/nist.sp.800-57p1r3

[4] Bos, J., and Kaihara, M., and Kleinjung, T., and Lenstra, A. K., and Montgomery, P. L., *On the Security of 1024-bit RSA and 160-bit Elliptic Curve Cryptography*, (2009), http://lacal.epfl.ch/page81774.html

[5] Durge, P., and Hajare, H. R., Comparative Analysis of RSA and ECC Algorithm, *International Research Journal of Engineering and Technology*, 7(6) (2020), 2395–0056.

[6] Gobi, M., and Sridevi, R., and Rahini, R., A comparative study on the performance and the security of RSA and ECC algorithm, *International journal of advanced network and application*, (2015).

[7] Gura, N., and Patel, A., and Wander, A., and Eberle, H., and Shantz, S. C., Comparing elliptic curve cryptography and RSA on 8-bit CPUs, in Cryptographic Hardware and Embedded Systems - CHES 2004, ser. Lecture Notes in Computer Science, M. Joye and J. Quisquater, Eds. Springer Berlin Heidelberg, 3156 (2004), 119–132, https://doi.org/10.1007/978-3-540-28632-5_9

[8] Gura, N., and Patel, A., and Wander, A., and Eberle, H., and Shantz, S. C., Comparing elliptic curve cryptography and RSA on 8-bit CPUs, *Cryptographic Hardware and Embedded Systems-CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings 6, Springer*, (2004), 119–132. https://doi.org/10.1007/978-3-540-28632-5_9

[9] Hankerson, D., and Menezes, A. J., and Vanstone, S., *Guide to elliptic curve cryptography, Springer Science and Business Media*, (2004).

[10] Jansma, N., and Arrendondo, B., Performance comparison of elliptic curve and rsa digital signatures, *nicj. net/files*, (2004).

[11] Kak, A., Lecture notes on computer and network security, *Purdue University*, (2015). https://doi.org/10.1007/978-3-642-29280-4_1

[12] Koblitz, N., Elliptic curve cryptosystems, *Mathematics of computation*, 48(177) (1987), 203–209. https://doi.org/10.1090/s0025-5718-1987-0866109-5

[13] Kute, V. B., and Paradhi, P. R., and Bamnote, G. R., A software comparison of RSA and ECC, *Int. J. Comput. Sci. Appl*, 2(1) (2009), 43–59.

[14] Mahto, D., and Khan, D. A., and Yadav, D. K., Security analysis of elliptic curve cryptography and RSA, *Proceedings of the world congress on engineering*, 1 (2016), 419–422.

[15] Mahto, D., and Yadav, D. K., Rsa and ECC: A com-parative analysis, *International Journal of Applied Engineering Research*, 12(19) (2017), 9053–9061.

[16] Mallouli, F., and Hellal, A., and Saeed, N. S., and Alzahrani, F. A., A survey on cryptography: comparative study between RSA vs ECC algorithms, and RSA vs El-Gamal algorithms, *2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud), IEEE*, (2019), 173–176, https://doi.org/10.1109/CSCloud/EdgeCom.2019.00022

[17] Maqsood, F., and Ahmed, M., and Ali, M. M., and Shah, M., Cryptography: a comparative analysis for modern techniques, *International Journal of Advanced Computer Science and Applications, Science and Information (SAI) Organization Limited*, 8(6) (2017), https://doi.org/10.14569/ijacsa.2017.080659

[18] Miller, V. S., Use of elliptic curves in cryptography, In Conference on the Theory and Application of Cryptographic Techniques Springer, Berlin, Heidelberg (1985), 417-426. https://doi.org/10.1007/3-540-39799-x_31

[19] M. Rafeek Khan et al., Analysis of Elliptic Curve Cryptography & RSA *Journal of ICT Standardization*, (2023), https://doi.org/10.13052/jicts2245-800X.1142

[20] M. Sun, Application of Elliptic Curve Cryptography in the Security Field of Resource-constrained Internet of Things, (2025). https://doi.org/10.54254/2753-8818/2025.dl28336

[21] Raman Kumar, Design and Analysis of various computations using Elliptic Curve Cryptography and RSA, *International Journal of Intelligent Systems and Applications in Engineering*, (2024), https://doi.org/10.1063/5.0209083

[22] Rivest, R. L., and Shamir, A., and Adleman, L., A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM New York, NY, USA*, 21(2) (1978), 120–126. https://doi.org/10.1145/359340.359342

[23] Singh, S. R., and Khan, A. K., and Singh, S. R., Performance evaluation of RSA and elliptic curve cryptography, *2nd International Conference on Contemporary Computing and Informatics (IC3I), IEEE*, (2016), 302–306. https://doi.org/10.1109/ic3i.2016.7917979

[24] Stallings, W., Cryptography and Network Security: Principles and Practice, Prentice Hall, Sixth Edition, (1998).

[25] S. Alshammari et al., Elliptic Curve Cryptography: Applications, challenges, recent advances, and future trends, *Computer Science Review, Elsevier*, 47, (2023).

[26] V. N. Sokolov et al., Cryptographic encoding in modern symmetric and asymmetric encryption, *Procedia Computer Science*, vol. 207 (2022) 54–63, https://doi.org/10.1016/j.procs.2022.09.037