*Journal of Discrete Mathematical Sciences & Cryptography*
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 27 (2024), No. 2-A, pp. 465–476
DOI : 10.47974/JDMSC-1902

# Implementing end-to-end encryption strategist for strengthening data security across network communication

Vijaya Parag Balpande *
*Department of Computer Science and Engineering*
*Priyadarshini College of Engineering*
*Nagpur*
*Maharashtra*
*India*

Ujjwala Bal Aher †
*Department of Computer Engineering*
*Government Polytechnic*
*Nagpur*
*Maharashtra*
*India*

Pradnya Borkar §
*Symbiosis Institute of Technology, Nagpur*
*Symbiosis International (Deemed University)*
*Pune*
*Maharashtra*
*India*

Parul Bhanarkar ‡
*School of Computer Science and Information Technology*
*Symbiosis Skills and Professional University*
*Pune*
*Maharashtra*
*India*

* *E-mail:* **vpbalpande15@gmail.com (Corresponding Author)**
† *E-mail:* **ujjwalaaher@gmail.com**
§ *E-mail:* **pradnyaborkar2@gmail.com**
‡ *E-mail:* **bhanarkar1111@gmail.com**

Gurunath T. Chavan [@]
*Department of Information Technology*
*Vishwakarma Institute of Information Technology*
*Pune*
*Maharashtra*
*India*

Raju M. Sairise [#]
*Department of Electronics and Telecommunication*
*Yadavrao Tasgaonkar College of Engineering & Management*
*Bardi*
*Maharashtra*
*India*

**Abstract**

In order to improve data security in network communication, this project suggests using End-to-End Encryption (E2EE). E2EE protects against unwanted access by guaranteeing that private information is kept secret for the duration of its travel. In order to mitigate potential vulnerabilities at intermediate points, cryptographic procedures are implemented to safeguard data from the point of origin to its final destination. The method makes use of strong encryption algorithms and key management procedures in order to strengthen the confidentiality and integrity of data that is transferred. This method tackles the growing worries about data breaches, privacy violations, and cyberthreats. By emphasizing E2EE, communication networks' security posture is strengthened and user and stakeholder trust is increased. By implementing this proactive security solution, organizations may protect sensitive data, comply with changing regulatory requirements, and provide a strong basis for safe data sharing in the digital sphere.

***Subject Classification:*** *Primary 93A30, Secondary 49K15.*

***Keywords:*** *Encryption, Data security, Confidentiality, Network communication.*

## 1. Introduction

The quick development of digital communication has shaped how people, companies, and organizations communicate and is now a fundamental part of modern society. Digital communication channels are being used more and more due to the growth of internet platforms, cloud services, and connected gadgets. But because sensitive data is moving through multiple networks, this dependency creates an increased risk of

[@] *E-mail:* **gt.chavan@gmail.com**
[#] *E-mail:* **rsairise566@gmail.com**

security lapses. Strong security measures are vital in the field of network communication, as evidenced by the rise in the scope and frequency of cyber threats in recent years. Sensitive data is frequently transferred across complex networks, leaving it open to interception and illegal access. Consequently, maintaining data integrity and confidentiality has become a top priority for both individuals and organizations.Many weaknesses in today's network communication infrastructures jeopardize the security of data transmission. Data eavesdropping, man-in-the-middle assaults, and unauthorized access are common dangers that erode the reliability of digital communication. The shortcomings of conventional security protocols, such virtual private networks (VPNs) and transport layer security (TLS), reveal crucial openings that malevolent actors take advantage of to jeopardize the integrity and confidentiality of important data.

The objective of this study is to tackle the previously described obstacles by presenting an all-encompassing execution of end-to-end encryption (E2EE) in network communication. The principal objective is to enhance the security stance of digital communication channels by guaranteeing that information stays private and intact during its journey from the source to the recipient. This study aims to improve data security by reducing the risks associated with different cyberthreats by implementing E2EE.

## 2. Proposed Method

A strong technique combining the Advanced Encryption Standard (AES) and the Elliptic Curve Cryptography (ECC) key generation algorithm is presented to strengthen data security via an End-to-End Encryption (E2EE) strategy.AES is the symmetric encryption algorithm used in this procedure, which guarantees the integrity and secrecy of the data being transported. A distinct symmetric key is dynamically produced for every communication session, lowering the danger of long-term key exposure. AES is a popular and efficient method for encrypting huge amounts of data because it strikes a balance between security and computing performance, as shown in Figure 1.

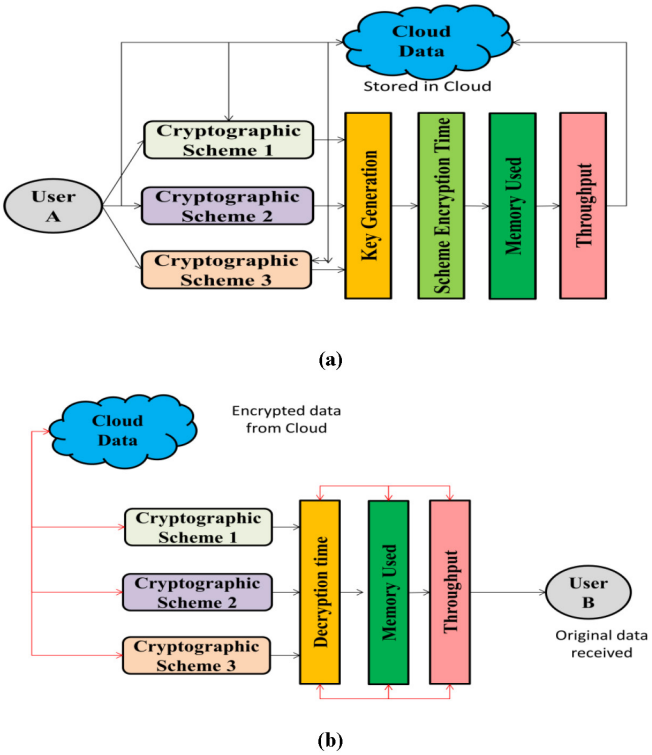


**Figure 1**

**Proposed Model (a) Encryption and (b) Decryption for E2E data security**

Concurrently, the asymmetric keys used in the safe transfer of session keys between communication parties are managed by the ECC key generation process. Because it may provide strong security with lower key lengths, ECC is a strong option for contexts with limited resources, including mobile devices and Internet of Things applications. While keeping a high level of security, the usage of ECC improves the encryption process's overall efficiency.

The suggested E2EE approach combines the benefits of PFS for continuous key refreshing, ECC for key generation and exchange, and AES for symmetric encryption. This all-encompassing method seeks to offer a robust and effective means of protecting data while it is being transmitted, guaranteeing that integrity and confidentiality are preserved throughout a variety of communication routes. By combining the

advantages of both AES and ECC, a comprehensive solution to satisfy the needs of contemporary data security regulations is provided.

## 3. AES Algorithm

The symmetric key encryption model that AES uses the same key for both encryption and decryption. This algorithm's efficiency is enhanced by its simplicity, which makes it ideal for protecting massive amounts of data in real-time communications. For data security in End-to-End Encryption systems, AES is a fundamental component. Its symmetric key concept, algorithmic resilience, key management features, performance attributes, and broad use make it a dependable and efficient alternative for securing confidential data while it's being transmitted. The usage of AES greatly aids in the establishment of secure communication channels, promoting trust and privacy in the digital sphere even as E2EE remains a crucial data protection technique.

---

**Algorithm:**

a. Round Constant (Rcon) Calculation:

$Rcon[i] = (x^{(i-1)}, 0, 0, 0)$ in GF($2^8$), where x is a primitive polynomial in GF($2^8$).

b. SubWord:

Substitutes each byte of the word with the corresponding byte from the S-Box.

c. RotWord:

Rotates the word cyclically.

d. Round Key Calculation:

$$W[i] = W[i-4] \oplus SubWord(RotWord(W[i-1])) \oplus Rcon\left[\frac{i}{4}\right] for\ i\ mod\ 4 = 0$$

$$W[i] = W[i-4] \oplus W[i-1] for\ i\ mod\ 4 \neq 0.$$

e. SubBytes:

Substitute each byte in the state with the corresponding byte from the S-Box.

$$S[i, j] = S - Box[S[i, j]]$$

f. ShiftRows:

Shift the rows of the state array.

$$S[i, j] = S[i, (j + i) \bmod 4]$$

g. MixColumns:

Mix the columns of the state array using matrix multiplication in GF($2^8$).

$$S'[i, j] = (02 \cdot S[i, j]) \oplus (03 \cdot S[i, (j+1) mod 4]) \oplus S[i, (j+2) mod 4] \oplus S[i, (j+3) mod 4]$$

h. AddRoundKey:

XOR the state with the round key.

$$S[i, j] = S[i, j] \oplus RoundKey[i, j]$$

---

## 4. Key Generation and Key exchange for E2EE Security

4.1 *Key Generation*

a. Elliptic Curve Equation:

Choose an elliptic curve defined over a finite field Fp with equation:

$$E : y^2 \equiv x^3 + ax + b (mod\, p)$$

b. Base Point Selection:

Choose a base point G on the elliptic curve E, where,

$$G = (x_G, y_G).$$

c. Select Private Key:

Choose a random integer d as the private key, where 1 < d < n (n is the order of the base point G).

d. Compute Public Key:

Compute the public key Q, where $Q = d * G$.

$$Q = [d]G$$

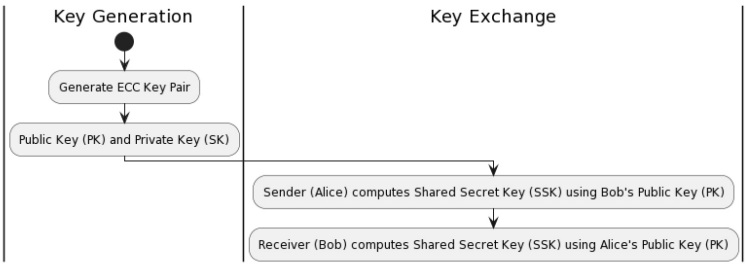


**Figure 2**

**Flowchart for Key generation and Key exchange for ECC**

4.2 *Key Exchange*

a. Elliptic Curve Parameters:
The communicating parties agree on the elliptic curve parameters (a, b, p) and the base point G.

b. Private Key Generation:
Each party generates its private key.

$$Party\ A : d_A$$

$$Party\ B : d_B$$

c. Public Key Computation:
Each party computes its public key.

$$Party\ A : Q_A = [d_A]G$$

$$Party\ B : Q_B = [d_B]G$$

d. Key Exchange:

$$Party\ A\ sends\ Q_A\ to\ Party\ B.$$

$$Party\ B\ sends\ Q_B\ to\ Party\ A.$$

Figure 2 shows the flow of the Key Generation and Exchange of ECC.

e. Shared Secret Calculation:
Both parties independently compute the shared secret using the received public key and their private key.
Shared Secret for Party $A : S_A = [d_A]Q_B$
Shared Secret for $Party\ B : S_B = [d_B]Q_A$

The calculated shared secrets S_A and S_B will be the same due to the properties of elliptic curve scalar multiplication. These shared secrets can then be used as symmetric keys for encryption and decryption in secure communication between the parties.

## 5. Experimentation and Analysis

Table 1 compares the encryption timings for three different algorithms: AES, ECC, and RSA, for a range of data sizes, from 20 KB to 200 KB. The information highlights how well each encryption method handles various information volumes. Interestingly, AES outperforms both RSA (80 ms) and ECC (60 ms) in terms of encryption speed for the smallest data size of 20 KB, with a time of 30 ms. The disparities in encryption times between the approaches become more noticeable as the size of the data rises.

**Table 1**

**Comparison of Encryption for different methods**

| Data Size (kb) | RSA (ms) | AES (ms) | ECC (ms) |
|---|---|---|---|
| 20 | 80 | 30 | 60 |
| 80 | 97 | 38 | 75 |
| 120 | 102 | 62 | 82 |
| 150 | 110 | 85 | 100 |
| 180 | 128 | 115 | 145 |
| 200 | 149 | 125 | 165 |

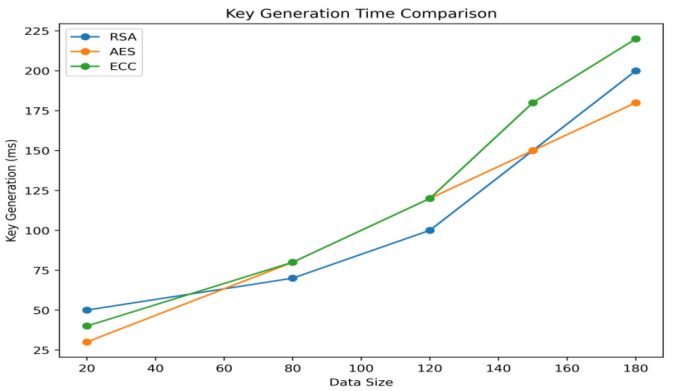


**Figure 3**

**Key generation and time taken for different model**

The RSA encryption times show a consistent rise, with a 200 KB dataset requiring 149 ms, highlighting the encryption's susceptibility to bigger data volumes. With its symmetric key technique, AES is well-known for maintaining continuously lower encryption times than RSA and demonstrating scalability as data sizes increase. Conversely, the asymmetric key algorithm ECC exhibits competitive performance across the spectrum, with progressively longer encryption durations. Interestingly, ECC records 165 ms at 200 KB, making it a competitive option to RSA and AES. Figure 3 Shows the comparison for key generation time consumption.

Decryption timings for RSA, AES, and ECC are compared in Table 2 for data sizes ranging from 20 KB to 200 KB. AES is the most efficient at 40 ms at 20 KB, followed closely by RSA and ECC at 82 ms and 60 ms. AES retains efficiency as data size grows, scaling to 125 ms for 200 KB. On the other hand, RSA decryption times increase steadily and reach a maximum of 158 ms for 200 KB. The asymmetric approach ECC maintains its competitiveness; hitting 147 ms. Decision-makers can choose an algorithm based on application requirements by using the table, which highlights variances in decryption efficiency. It emphasizes the necessity of striking a compromise between computing efficiency and security when making cryptographic decisions, taking into account the unique advantages of each method for various data size scenarios. Figure 4 shows the time comparison of Encryption and Decryption of Different Models.

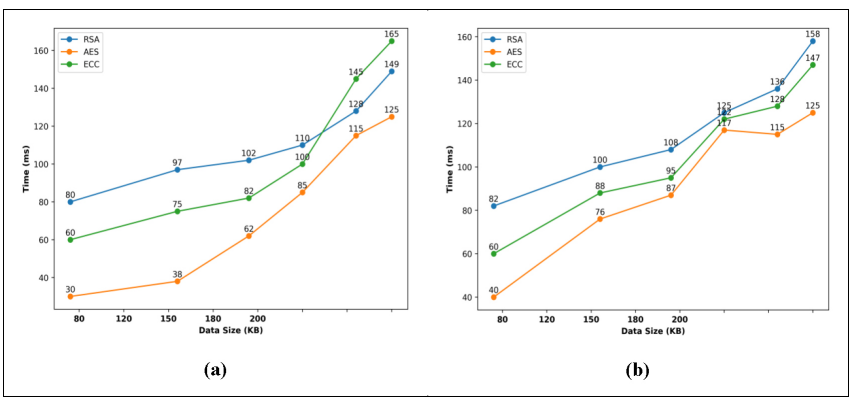


**Figure 4**

**Comparison of (a) Encryption and (b) Decryption for different methods**

**Table 2**

**Comparison of Decryption for different methods**

| Data Size (kb) | RSA (ms) | AES (ms) | ECC (ms) |
|---|---|---|---|
| 20 | 82 | 40 | 60 |
| 80 | 100 | 76 | 88 |
| 120 | 108 | 87 | 95 |
| 150 | 125 | 117 | 122 |
| 180 | 136 | 115 | 128 |
| 200 | 158 | 125 | 147 |

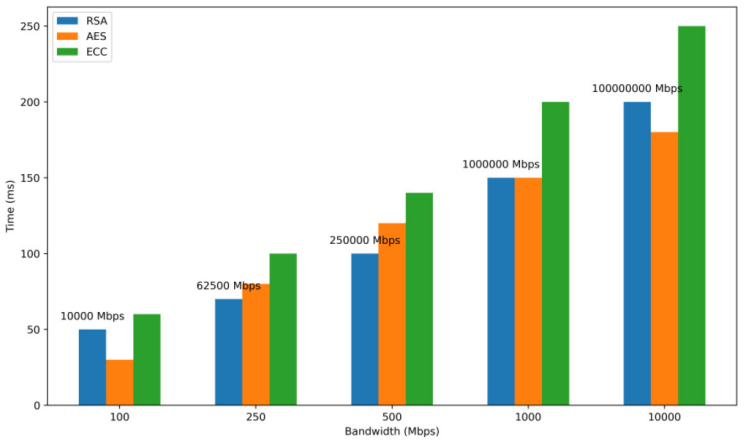


**Figure 5**

**Encryption Time Comparison**

A thorough comparison of the encryption times for the three cryptographic techniques RSA, AES, and ECC is shown in Figure 5. The graph gives a clear picture of how well each algorithm works with varying amounts of data, revealing important details about the advantages and disadvantages of each approach in terms of computational efficiency.By looking at the graph, one can see that AES consistently shows faster encryption speeds than RSA and ECC for a range of data sizes.

## 6. Conclusion

A strong and all-encompassing approach has been adopted in the implementation of an End-to-End Encryption (E2EE) plan to strengthen

data security across network communication. Information is protected throughout its trip with E2EE, which secures data at both ends of the communication channel—from the sender to the recipient. By reducing the possibility of illegal access and interception, this calculated deployment promotes increased privacy and confidence. The encryption procedure serves as a strong defense against potential attacks since it frequently uses sophisticated algorithms like RSA, AES, and ECC. Because it offers a secure communication environment, E2EE is effective in helping organizations comply with strict data protection regulations. Adopting E2EE becomes essential for protecting sensitive data as cyber threats change because it provides a strong barrier against illegal access and data compromise.

## References

[1] Y. Ma, Q. Cheng and X. Luo, "2PCLA: Provable Secure and Privacy Preserving Enhanced Certificateless Authentication Scheme for Distributed Learning," in *IEEE Transactions on Information Forensics and Security,* vol. 18, pp. 5876-5889 (2023).

[2] Q. Zhang et al., "Efficient Blockchain-Based Data Integrity Auditing for Multi-Copy in Decentralized Storage," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 12, pp. 3162-3173, Dec. (2023).

[3] T. Mahjabin, A. Olteanu, Y. Xiao, W. Han, T. Li and W. Sun, "A Survey on DNA-Based Cryptography and Steganography," in IEEE Access, vol. 11, pp. 116423-116451 (2023).

[4] Y. Jiang, K. Zhang, Y. Qian and L. Zhou, "Anonymous and efficient authentication scheme for privacy-preserving distributed learning", IEEE Trans. Inf. Forensics Security, vol. 17, pp. 2227-2240 (2022).

[5] R. Xu and Q. Ren, "Cryptoanalysis on a cloud-centric Internet-of-Medical-Things-enabled smart healthcare system", IEEE Access, vol. 10, pp. 23618-23624 (2022).

[6] H. Shu, P. Qi, Y. Huang, F. Chen, D. Xie and L. Sun, "An efficient certificateless aggregate signature scheme for blockchain-based medical cyber physical systems", *Sensors*, vol. 20, no. 5, pp. 1521, Mar. (2020).

[7] Y. Chen and J. Chen, "An efficient and privacy-preserving mutual authentication with key agreement scheme for telecare medicine information system", Peer-Peer Netw. Appl., vol. 15, no. 1, pp. 516-528, Jan. (2022).

[8] R. Vinoth, L. J. Deborah, P. Vijayakumar and B. B. Gupta, "An anonymous pre-authentication and post-authentication scheme assisted by cloud for medical IoT environments", IEEE Trans. Netw. Sci. Eng., vol. 9, no. 5, pp. 3633-3642, Sep. (2022).

[9] L. Wu, J. Wang, K. R. Choo and D. He, "Secure key agreement and key protection for mobile device user authentication", IEEE Trans. Inf. Forensics Security, vol. 14, no. 2, pp. 319-330, Feb. (2019).

[10] Anil Kumar & Sandeep Kumar Sharma. Information cryptography using cellular automata and digital image processing, *Journal of Discrete Mathematical Sciences and Cryptography*, 25:4, 1105-1111 (2022), DOI: 10.1080/09720529.2022.2072437.

[11] M. A. Saleem, K. Mahmood and S. Kumari, "Comments on 'AKM-IoV: Authenticated key management protocol in fog computing-based Internet of Vehicles deployment", IEEE Internet Things J., vol. 7, no. 5, pp. 4671-4675, May (2020).

[12] T. Alladi, S. Chakravarty, V. Chamola and M. Guizani, "A lightweight authentication and attestation scheme for in-transit vehicles in IoV scenario", IEEE Trans. Veh. Technol., vol. 69, no. 12, pp. 14188-14197, Dec. (2020).

[13] S. K. Sharma, A. Chaurasia, V. S. Sharma, C. L. Chowdhary and S. Basheer, "GEMM, a Genetic Engineering-Based Mutual Model for Resource Allocation of Grid Computing," in IEEE Access, vol. 11, pp. 128537-128548 (2023), doi: 10.1109/ACCESS.2023.3333278.

[14] X. Feng, Q. Shi, Q. Xie and L. Wang, "P2BA: A privacy-preserving protocol with batch authentication against semi-trusted RSUs in vehicular ad hoc networks", IEEE Trans. Inf. Forensics Security, vol. 16, pp. 3888-3899 (2021).