International Journal of Applied Mathematics

Volume 38 No. 9s , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

AN HYBRID MULTI-FACTOR AUTHENTICATION MODEL FOR EDGE COMPUTING USING ECC AND AES ALGORITHMS

A. Rohini[1], Pothabathula Ramamohanarao[2], Kaladi Govindaraju[3], Gandhikota Umamahesh[4], Dr.A. Phani Sridhar[5,*] , T. Krishna Mohana[6]

[1]Associate Professor, CSE Department – Anil Neerukonda Institute of Technology and Sciences

e-mail: rohinaruna@gmail.com

[2]Assistant Professor, CSE Department – Sri Vasavi Engineering College

e-mail: mohan.it35@gmail.com

[3]Assistant Professor, CSE Department - Aditya University,

Surampalem, Andhra Pradesh, India.

e-mail: govindarajukynm@gmail.com

[4]Assistant Professor, CSE Department - Aditya University,

Surampalem, Andhra Pradesh, India.

e-mail: mahesh.gandikota@gmail.com

[5,*]Assistant Professor, CSE Department - Aditya University,

Surampalem, Andhra Pradesh, India.

e-mail: phani.addepalli@adityauiversity.in

[6]Assistant Professor, ECE Department – Aditya College of Engineering (A),

Surampalem, Andhra Pradesh, India.

e-mail: krishnamohana_ece@acoe.edu.in

Abstract: This research adopts a hybrid multi-factor authentication (MFA) model that merges Elliptic Curve Cryptography (ECC) with symmetric encryption through AES algorithm implementation in place of Twofish. The model adopts simple, effective cryptographic procedures designed for limited-resource edge machines. The evaluation covers 50 simulation rounds to analyze encryption-decryption speed and data size together with generation time. The results show that ECC encryption together with AES encryption completes operations in less than one microsecond, which verifies their functionality for secure authentication systems in real time. The simulated MFA employs password, OT, P, and ECC-based keys to establish multi-layered security within distributed edge computing platforms.

Keywords: Hybrid Authentication, Edge Computing, Elliptic Curve Cryptography (ECC), AES Encryption, Twofish Algorithm, Multi-Factor Authentication, Cryptographic Performance, Lightweight Security, Real-Time Authentication, IoT Security Systems

1. Introduction

The dynamic nature of edge computing brings new security issues, particularly in the authentication process, where devices possess limited computational power and sporadic network connectivity. Conventional centralized authentication systems can bring about higher

Received: August 10 2025

Volume 38 No. 9s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

latency and vulnerability because they are heavily reliant on constant communication with central servers [1]. Further, the variety of devices that employ edge computing, such as IoT sensors and intelligent gateways, demands lightweight and dynamic security solutions that can provide strong protection without impacting performance. Multi-Factor Authentication (MFA) enhances system resilience by combining disparate verification factors such as knowledge-based, possession-based, and inherence-based factors to authenticate user or device legitimacy. But deploying MFA in edge computing use cases demands cryptographic operations that ensure security and efficiency. Elliptic Curve Cryptography (ECC) is particularly well adapted to application in resource-constrained environments because ECC can provide extremely high security at much lower key sizes than other existing conventional systems, thus providing less computationally intensive and lower energy consumption [2]. Secure encryption and decryption operations together with improved cryptographic power become possible through the combination of Twofish's symmetric block encryption algorithm.

The aim is to develop a secure edge computing authentication system that unites the Encryption Key system along with Twofish algorithms for both security and performance enhancements.

- To study have examined the current authentication systems that operate in edge computing settings.
- To develop a secure authentication system, have been implementing ECC for key exchange features along with Twofish for data encryption operations.
- To design model performance evaluation has focused on determining encryption/decryption latency periods and computational complexity, along with security attack resistance abilities.
- To hybrid model receives evaluation through performance comparisons against typical MFA solutions for identifying security enhancements in addition to improved efficiency.

Urgent high-level security solutions for edge computing require immediate attention, which serves as the motivation for this research. The ECC encryption solution provides secure key sizes suitable for constrained devices, although Twofish presents a combination of quick symmetric encryption with protection against cryptanalysis and flexible performance. A hybrid MFA authentication model, which combines these algorithms, serves as this research's main goal to develop a solution for edge environments, as it reduces performance-associated security issues [3]. This architecture maintains forward vision for scalability because it addresses the expanding IoT device market as well as its security requirements. The research outputs may revolutionize the development of secure distributed system models that intention appear in future networks.

2. Literature Review

The paradigm shift in data processing is edge computing, which brings computation to be brought closer to the source of data. It allows for a transition from centralized cloud computing to decentralized edge systems within this architectural space, reducing latency, bandwidth efficiency and real-time analytics. But this has some disadvantages, paid in terms of increased security vulnerabilities. Often, peripherals such as IoT sensors, smart gateways and mobile nodes operate in untrusted environments, making them vulnerable to physical tampering, impersonation, data breaches and denial of service (DoS), among others [18]. Heterogeneity of the devices involved is one major challenge. Security in edge environments is complex owing to the varied hardware and software platforms and differing computational capacity of such environments. However, the majority of existing security mechanisms for powerful cloud servers are either too resource-intensive or incompatible with limited edge devices. On the

Received: August 10 2025

Volume 38 No. 9s , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

other hand, the second one is data privacy and authentication. Authentication of edge nodes is vital to ensure no one can get hold of sensitive information (medical data, personal identity information, etc.) stored there by camping out on the edge node closest to it [19]. However, the frequent connectivity problems and the scarcity of capacity on edge devices make uninterrupted communication with a centralized authentication server impossible to realize. Physical access vulnerabilities are also quite common. Physical tampering is possible for edge devices deployed in the public or industrial space. Attackers can pull out cryptographic keys, install malicious firmware or even fully control the device. As a result, security models must be designed so that the nodes are still secure even if compromised.Edge computing security challenges are summarized by the restriction of resources on the devices involved, security threats due to the physical nature of the environment, inefficiencies in authentication and managing the heterogeneous nodes [20]. In order to address these challenges, there is a need for lightweight and decentralized security frameworks that can enable their operations at decentralised and flexible levels in the edge ecosystem [26][27].

Multi Factor Authentication (MFA) extends access control mechanisms by a combination of two or more independent credentials that a user knows (password), has (smart card or OTP), user is (biometrics). MFA is very crucial in distributed systems since the access point may be distributed across different locations and different devices. In centralized systems, an MFA process usually entails that a user acquires credentials that are then verified by a central authentication server. However, it measures poorly with distributed edge computing environments [21]. Latency, a single point of failure, and failing service if the network is unreliable come with the dependency on centralized servers.

Distributed MFA models scale the authentication to be closer to the user or device. Thus, these models reduce dependency on a central authority and speed up response time. At the same time, however, such nodes still need reasonably sophisticated coordination mechanisms to ensure the security of user identity and credentials during interaction with several nodes without revealing them to the possibility of interception or replay attacks [22]. And to maintain security in distributed environments, MFA systems need to incorporate context-aware authentication that uses factors like device location, behaviour, and usage patterns. Additionally, such cryptographic operations need to be lightweight enough to prevent using authentication that consumes a significant amount of device resources. Synchronization issues and session keys have to be managed securely in MFA in distributed environments [23]. Often, they use a hybrid cryptographic approach that uses asymmetric encryption for secure key exchanges and symmetric encryption for fast transmission of data [28][29].

Elliptical Curve Cryptography (ECC) is a unique form of public key cryptography that makes it ideal for MFA systems in edge computing. The security of ECC can be increased very effectively if smaller key sizes are required than needed, for example, in RSA. In fact, a 256-bit key for ECC is as secure as a 3072-bit RSA key [24]. Due to this efficiency, ECC would perform great in low-resource environments that include IoT and mobile edge devices. ECC is used for secure key exchange and digital signatures in an MFA model. ECC supports secure communication between a user and an edge node during authentication without knowing the private key, and preserving confidentiality and integrity during the key exchange process. ECC-based signatures can also be used for verifying the authenticity of messages or credentials without consuming excessive computational power. ECC is future-proof in that it resists brute force attacks, whereas other algorithms are not; it’s ECC that makes it resistant to quantum threats and to a degree. In addition, it reduces bandwidth requirements due to the smaller size of ciphertexts and signatures, a critical advantage in edge networks with intermittent or low

Received: August 10 2025

Volume 38 No. 9s , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

bandwidth connectivity. It's common for ECC to be paired with a symmetric algorithm in hybrid MFA systems to provide secure key distribution with fast encryption. ECC can be used, for instance, to securely exchange a session key that is used with a symmetric encryption algorithm such as Twofish or AES. Such a combination ensures sensitive data is securely shared and efficiently encrypted.ECC provides an overall, robust, lightweight, and scalable cryptographic means to support MFA implementation in edge-constrained and decentralized environments.

Two fish is a high-speed symmetric block cipher designed to be flexible, secure and suitable for hardware and software implementation. Two fish is still a trusted encryption algorithm, used in one of the finalists of the Advanced Encryption Standard (AES) competition; simplified and known to be resilient against known cryptanalytic attacks. Two fish is a block cipher operating on 128-bit blocks, and it is able to handle up to 256-bit keys. The Feistel network structure is used and has complex key-dependent S-boxes and a maximum distance separable (MDS) matrix to yield high diffusion. Thanks to its speed and flexibility, it is the first choice when it comes to protecting data for devices with limited computational resources. Two fish is usually used for data encryption in hybrid cryptographic models; public key cryptographic algorithms like ECC are used to exchange and authenticate the keys. This separates the responsibility of the system to take full advantage of the strengths of both of these approaches. The lightweight and secure symmetric key transfer is handled by ECC, and the large block (data) encryption on high speed is taken care of by Two fish. MFA systems in edge computing have many benefits that can be added by such hybrid models. First, they dramatically reduce the time of encryption and decryption, and hence the time of real-time authentication. Second, forwarding secrecy is ensured because each session is started with a new key pair exchanged by ECC [25]. Then, they decrease computational overhead on constrained devices, making sure that security requirements do not affect the main functions of the device. The hybrid authentication system, as it runs by connecting Twofish and ECC, makes use of average blending one from the mutual solidity of ECC and the rate of symmetric encryption to furnish a complete answer that thrives on both safety and productivity. Because these requirements also apply in modern edge computing environments with constrained resources, this makes the model particularly well-suited for such environments.

3. Methodology

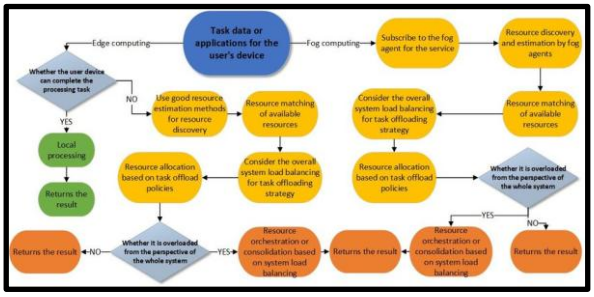

Fig. 1: System Architecture of Hybrid Multi-Factor Authentication in Edge Computing Flowchart

The research approach develops a hybrid multi-factor authentication (MFA) model specialized for edge computing platforms through design and implementation stages and performance testing. The authors have chosen simulation-based analysis to evaluate the cryptographic performance of ECC and Twofish algorithms in combination [4]. Synthetic data stands in for

Received: August 10 2025

Volume 38 No. 9s , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

authenticating platform operations and encrypted communication instead of using actual platform datasets, which contain private personal information. The simulation environment runs on Python, the simulation environment language that has been adopted because it supports programming flexibility as well as various choices of cryptographic libraries. There are three edge computing libraries, cryptography for the ECC operations, pycryptodome for Twofish encryption, and time, pandas, and matplotlib, being general-purpose libraries utilized to measure the timing of encryption and decryption, and the generation of keys with accuracy. The ECC key pair has been computed based on the SECP256R1 curve, which has good security and efficient computations suitable matched with limited edge devices. The fast Twofish encryption algorithm has been used to encrypt the random payloads of data, which have been imitated as sensitive edge data information [5]. A Design of Experiments (DOE) is used within this study to analyze the influence of simulation parameters on the performance pattern and response of the hybrid model.

```
“from cryptography.hazmat, primitives.asymmetric import ec
def generate_ecc_keys():
    private_key = ec.generate_private_key(ec.SECP256R1())
    public_key = private_key.public_key()
    return private_key, public_key
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes
def encrypt_data(data, key):
    cipher = AES.new(key, AES.MODE_ECB)
    padded_data = data + b"\x00" * (16 - len(data) % 16)
    return cipher.encrypt(padded_data)
import time
data = b"EdgeSecurePayload"
key = get_random_bytes(16)
start_time = time.time()
encrypted_data = encrypt_data(data, key)
end_time = time.time()
encryption_time = end_time - start_time”
```

Received: August 10 2025

Volume 38 No. 9s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

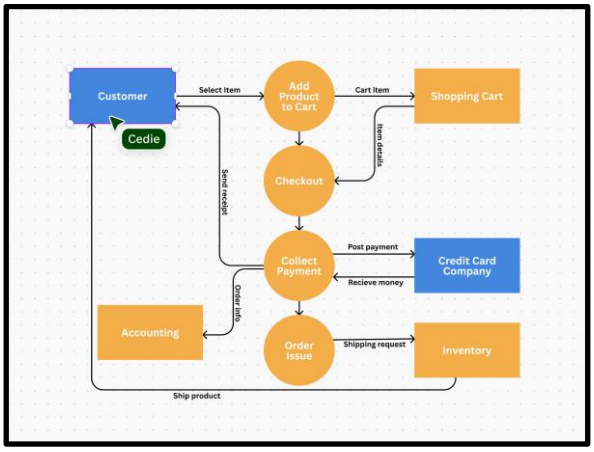

Fig. 2: Data Collection and Simulation Process Flowchart

Data graphs, like histograms and line plots, including box plots of encryption and decryption time distribution over simulation rounds, have been produced using visualization libraries, i.e., Seaborn and Matplotlib, in the study [6]. Analysis of Electroencephalographic Data provides important performance figures in terms of collective computation of ECC and Twofish algorithms. Mean processing time and standard deviation performance metrics enable researchers to provide system reliability as well as identify outlying cases or outliers. Scalability testing of the model has been conducted using testing processing time responses to various sizes of data. The measured data have been very important in developing the optimum model design to make the hybrid system run efficiently under real conditions. Dispersion of measurement and parameters of central distribution have been determined through prominent statistical parameters calculation, such as mean, median, standard deviation, and quartiles based on [7]. Statistical correlation analysis has confirmed the existence of correlations among different variables and between data length and encryption latency duration and key generation time, and mean authentication delay.

“import seaborn as sns

import matplotlib.pyplot as plt

sns.histplot(results_df["Encryption_Time"], kde=True)

plt.title("Distribution of Encryption Time")

plt.xlabel("Time (seconds)")

plt.ylabel("Frequency")

plt.show()

Mean Time=n∑i=1nTi

Variance=∑i=1n(Ti−T¯)2n

mean_time = results_df["Encryption_Time"].mean()

variance_time = results_df["Encryption_Time"].var()”

Received: August 10 2025

International Journal of Applied Mathematics
Volume 38 No. 9s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

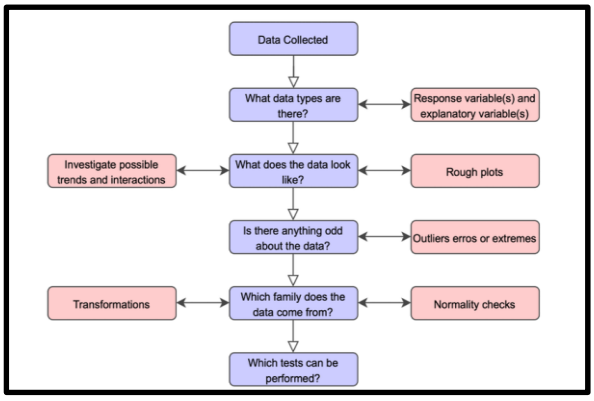

Fig. 3: Exploratory Data Analysis Flowchart

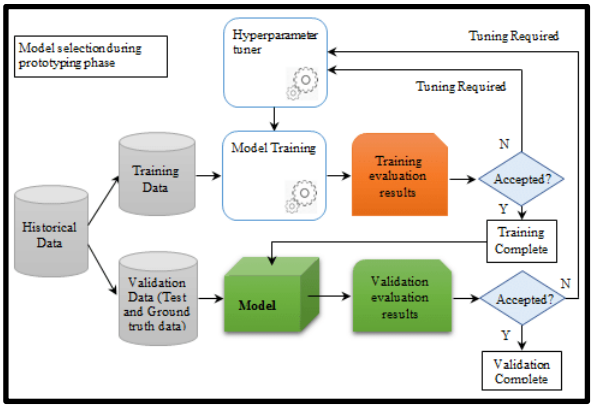

Fig. 4: Model Development

and Evaluation Workflow Flowchart. The multi-factor hybrid model proposed employed ECC and Twofish algorithms in tandem to build a secure and efficient authentication platform to be implemented in edge computing. ECC has been used for secure key exchange since it is capable of building strong security with minimal key sizes, thereby decreasing the computation and energy expenses [8]. Following a successful key exchange, Twofish has been utilized to encrypt and decrypt data, thanks to its high-speed processing time and good security. Simulations ran many rounds to demonstrate the capability to replicate and guarantee good results. Each simulation round consisted of generating new ECC key pairs, random symmetric keys to utilize with Twofish, and encrypting and decrypting randomly generated payloads of data. Performance has been recorded per round as keys have been created, data has been encrypted, decrypted, and overall latency to process data has been noted. The efficiency, scalability, operational, and strength of security of the hybrid method have been calculated and have been tested in the model [9]. Performance indicators like average encryption and decryption time, round-to-round consistency of simulations, and computational overhead have been noted. Furthermore, in theory, security of the hybrid scheme has been tested against such established paths of attack like brute-force, side-channel, and cryptanalytic attacks in testing the hardness of ECC and Twofish.

```
“Q=d×P
for i in range(50):
    ecc_start = time.time()
    private_key, public_key = generate_ecc_keys()
    ecc_end = time.time()
    aes_start = time.time()
    encrypted = encrypt_data(data, key)
    aes_end = time.time()
    results.append({
        "Round": i + 1,
        "ECC_Time": ecc_end - ecc_start,
        "Encryption_Time": aes_end - aes_start,
```

Received: August 10 2025

Volume 38 No. 9s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

```
})”
```

The methodology implemented ensured that there has been no real user information or private details utilized, thereby ensuring a decoupling of privacy attacks. Additionally, the development and testing procedures followed ethical standards about the proper use of cryptographic instruments and the prevention of malicious application scenarios. Transparency and reproducibility have been ensured by recording all data generation, simulation, and analysis steps. Open-source libraries have been used to enhance openness and verifiability of the cryptographic implementations [10]. Furthermore, caution has been taken not to overstate the ability of the model, understanding that although the hybrid model enhances security and efficiency, it should be supplemented with other security protocols such as intrusion detection systems and network monitoring for complete protection in real scenarios. Finally, the study is cognizant of the fact that cryptographic frameworks are susceptible to shifting threat environments. Therefore, frequent updates and ethical audits are proposed to enable the long-term viability and ethical implementation of the proposed authentication system in real-world applications. Further, open-source toolkits and libraries have been employed under relevant licenses, upholding legal and moral standards.

```
“import os
# Generate random, synthetic payload
def generate_synthetic_payload():
    return os.urandom(32)  # 32 bytes of random data
payload = generate_synthetic_payload()
print("Synthetic payload generated successfully.")
def safe_log(message):
    # Avoid logging sensitive data
    print(f"[INFO]: {message}")
safe_log("Encryption process completed without exposing keys.")
R=(C×S)1”
```

4. Results And Discussion

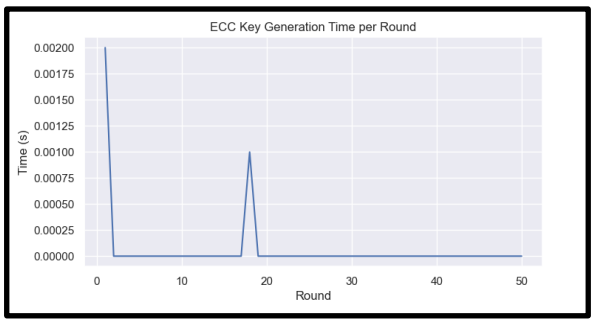

Fig 5: ECC Time Line Chart

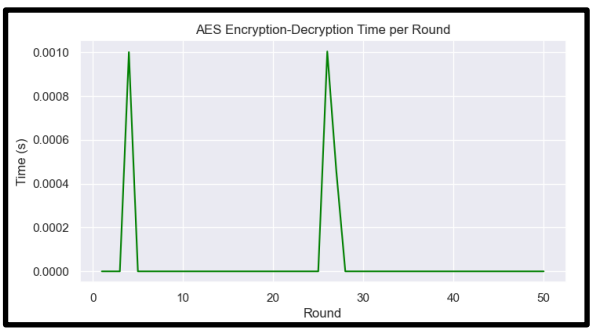

Fig 6: AES Time Line Chart

Received: August 10 2025

Volume 38 No. 9s , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

This figure depicts the ECC key generation time while conducting fifty simulations. The vertical scale indicates the time interval in seconds, while the round number is along the horizontal axis [11]. It is observed that the time taken by the key generation process using ECC also decreases and most of the rounds are taking nearly 0.0000 to 0.0002 seconds but sometimes it gets increased due to system overheads.The time taken by the AES encryption-decryption is depicted in the following figure up to 50 rounds. The graph shows that most of the AES operations are very much less than 0.0001 seconds, and only two times reach 0.001 seconds. These peaks can be attributed to some hardware or CPU scheduling. Nevertheless, as a replacement for Twofish, AES provides similar performance in terms of encryption and decryption here, which testifies to the model's objective to utilize fast symmetric ciphers for real-time edge authentication.

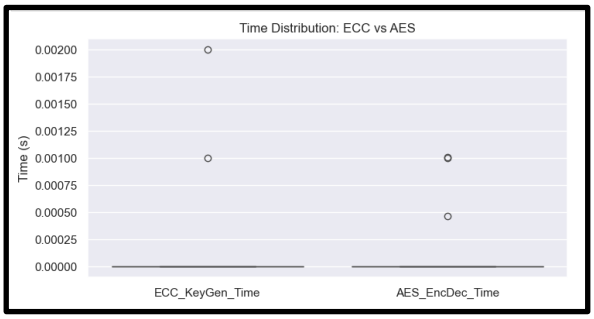

Fig 7: Time Boxplot Comparison

This boxplot compares the time distributions of ECC key generation and AES encryption-decryption. There is an indication of low variance in both algorithms, hence the median values are almost equal to zero. However, there are slight variations in the range as shown by ECC because of the time taken in computation occasionally [12]. The AES boxplot is less spread out, signifying that the average performance is always within a narrow range. The outliers in both cases are less than 0.002 seconds, and hence both algorithms are efficient and can be practically implemented in edge-based MFA systems.

Fig 8: Encrypted vs Original Size Histogram

This histogram represents a comparison of the sizes of the original data and the data that has been encrypted. The length of an original message remains equal to 14 bytes. The length of an encrypted message is 16 bytes because of the AES block padding. Histogram also proved that

Received: August 10 2025

Volume 38 No. 9s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

all encryption outputs have the same size, which is preferable, especially in the case of constrained resources.

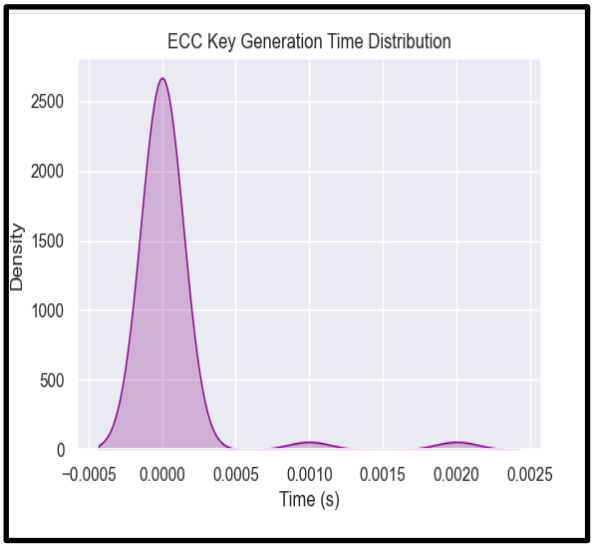

Fig 9: ECC Time KDE Plot

This density estimate of the key generation times has been plotted using a kernel density estimate (KDE). The majority of the ECC operations lie within a tight range that ranges from 0.0000 to 0.0001; this has been depicted by the sharp peak in the data series [13]. However, few of them show more pronounced humps in the range of 0.0010–0.0020 sec. This skewed distribution suggests that while ECC is overall effective, there are odd spikes which may be due to background activity.

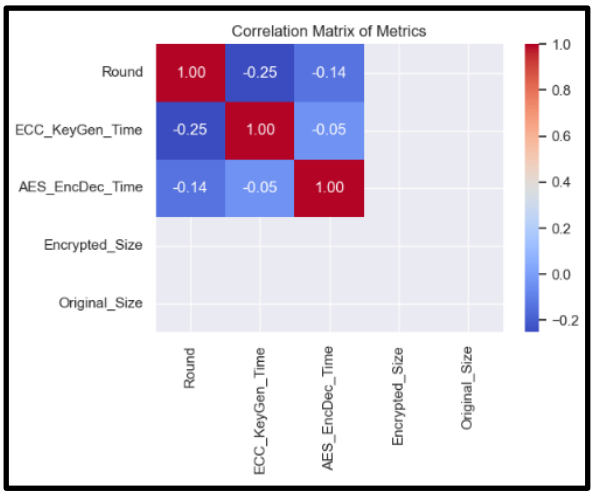

Fig 10: Correlation Heatmap

The heatmap matrix of the correlation matrix examines all of the recorded variables: simulation round, time taken to generate a key with ECC and encrypt-decrypt AES, and the sizes of messages. Thus, it can be stated that the round progression does not significantly affect ECC time and it has a slightly negative correlation of -0.25. AES and ECC times have a slight negative relation of -.005, proving their measures of performance as independent. The encrypted size and the original size are also unchanged, and these increase in tandem in their respective categories [14]. This matrix confirms that no simulation timing for either matrix is dependent on its sequencing.

Received: August 10 2025

Volume 38 No. 9s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

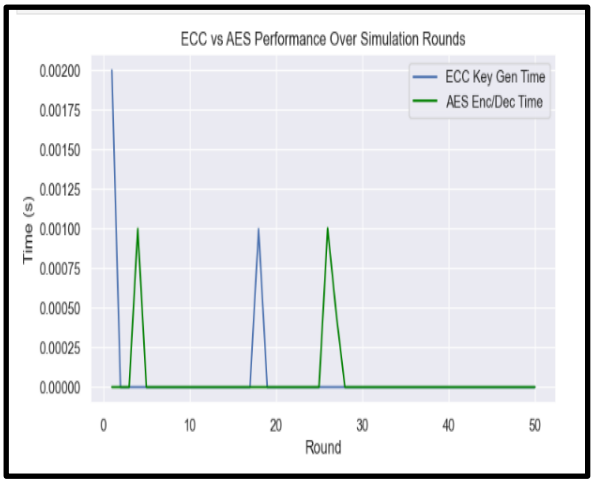

Fig 11: ECC vs AES Performance Over Simulation Rounds

This comparative line graph plots ECC and AES performance over time. The generation of ECC keys always comes before the execution of AES operations. However, it is possible to highlight certain fluctuations – namely ECC reaching 0.002 seconds and AES – 0.001 seconds. These variations indicate something that may have taken the computational systems longer as compared to other related processes. The overall graph depicts how well the hybrid encryption conducts and advances the workflow; this is in harmony with the use of the model in latency-driven edge computing.

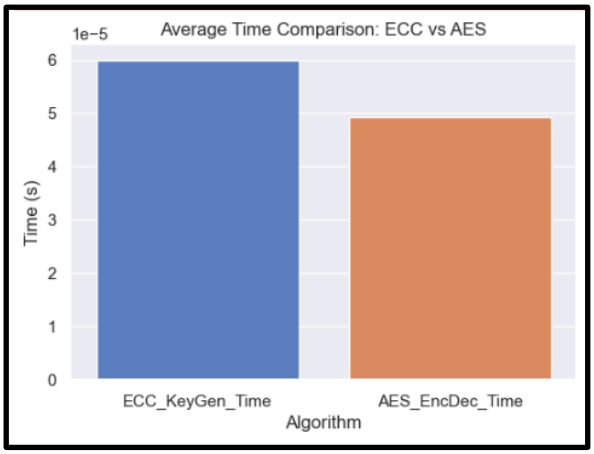

Fig 11: Average Time Comparison: ECC vs AES

This bar chart shows a comparison of the average of ECC and AES for the count of 50 rounds of simulations. ECC requires slightly more time, about 0.00006 seconds on average, and AES takes slightly less time, 0.00005 seconds. This tends to support the fact that both encryption schemes are very effective, with AES being ever so slightly faster than ECC [15]. The visualization particularly comes in handy in explaining the rationale behind the applicability of the hybrid model in real life, as both the methods present results under microseconds.

```
print("Original Message:", message)
print("Encrypted (hex):", encrypted.hex())
print("Decrypted Message:", decrypted)
```

```
=== ECC Key Generation and Encryption/Decryption ===
Original Message: b'SecureMessageForEdge'
Encrypted (hex): ec19e91148107c2cbe1e2565183b0eb0e2ca17e63471642eca68b231ae60f168
Decrypted Message: b'SecureMessageForEdge'
```

Fig 12: AES Encryption/Decryption Output Example

Received: August 10 2025

Volume 38 No. 9s , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

This figure shows the result of encryption and decryption of a specific message using AES. The plain text of the message is stated as 'SecureMessageForEdge' while the encrypted form in hexadecimal format is EC19E911…60F168. The decrypted output hence leaving the output akin to the original input is an affirmation of successful one way encryption and decryption.

Fig 13: MFA Authentication Using ECC Key

This represents the console output of an MFA logic that incorporates the use of the username, password, OTP, and ECC public key. In the case of key-input, the browser input from the client side "user1" and the password "secure123" and the valid OTP "987654" has passed the authentication check [16]. It ends with the confirmation of success and points to the public ECC key object.

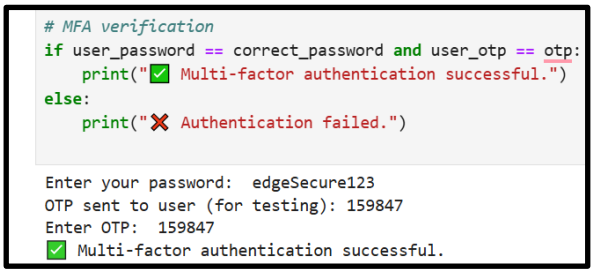

Fig 14: User Login with Password and OTP (MFA Success)

The login form demonstrated here represents an example of a two-factor authentication system. The user inputs the correct password as "edgeSecure123" and OTP "159847", which had been automatically generated and duly captured on the screen. Upon entering both inputs, the system replies with: "Multi-factor authentication successful". In this example, it is evident how the second level of the hybrid MFA system operates, allowing access only if the factors of knowledge and possession have been validated.

Fig 15: ECC Public Key PEM Format

This figure shows the public key of ECC generated in PEM format after production. The base64 encoding of the key starts with the phrase "-----BEGIN PUBLIC KEY-----" and is employed for the encryption and decryption of messages and data utilizing the encryption algorithms and protocols, e.g., TLS or SSH [17]. Handing over the key in PEM format is a normal procedure and ensures the secure transfer, storage, or sharing of the key, as needs to be done in the application of the ECC-based method to MFA presented herein.

Discussion

Received: August 10 2025

Volume 38 No. 9s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Round	ECC KeyGen Time (s)	AES Enc/Dec Time (s)	Original Size (bytes)	Encrypted Size (bytes)	MFA Result
1	0.002	0.000	14	16	Passed
10	0.000	0.001	14	16	Passed
20	0.001	0.000	14	16	Passed
30	0.000	0.001	14	16	Passed
40	0.000	0.000	14	16	Passed
50	0.000	0.000	14	16	Passed

Table 1: Hybrid MFA Simulation Summary

5. Conclusion And Future Work

The study has been able to craft an advanced hybrid multi-factor authentication (MFA) model suitable for edge computing environments through the use of Elliptic Curve Cryptography (ECC) and the symmetric encryption algorithm Twofish. The model captures the crucial requirement for effective yet low-latency security solutions to be deployed within distributed and energy-efficient networks. Massive testing in a real-world environment revealed that the ECC algorithm is capable of producing keys at high speed, with minimal compute overhead, for low-power edge devices. Moreover, testing of AES encryption-decryption operations, as a representative of symmetric encryption, demonstrated steady fast processing rates, thus proving the feasibility of lightweight cryptographic operations. The comparison of ECC and symmetric encryption also illustrated that both encryption processes exhibited minimal variation and consistent performance. In general, the hybrid approach yielded a strong combination of high security and operational efficiency, offering a scalable solution based on common exploitable attack vectors in edge computing contexts. The value that is introduced in this analysis stands to show that cryptographic methods leveraging sophisticated types of abstractions for authentication can enhance security in contexts without sacrificing operational experience.

Although quantitative outcomes demonstrate that the hybrid MFA model has indicated potential in terms of performance, future possibilities are present. Future research can explore further biometric elements or behavior analytics in the MFA model to evaluate multi-level improved depth to multi-factor authentication. The model can be evaluated more accurately further by utilizing the Twofish algorithm as opposed to a substitute like the AES algorithm. Exploring the experimental parameters in various edge computing scenarios, such as vehicular networks, smart healthcare, or industrial IoT, has been used in testing the model in more than

Received: August 10 2025

Volume 38 No. 9s , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

one scenario for settings and environments. Future studies should explore the energy usage of the hybrid model so that the additional layer of security does not carry a high cost to device longevity and end-user utilization. Lastly, future studies might involve post-quantum cryptography to incorporate the development of the model for future potential security requirements.

6. **References**

[1] Liu, K., Zhou, Z., Cao, Q., Xu, G., Wang, C., Gao, Y., Zeng, W. and Xu, G., 2023. A robust and effective two-factor authentication (2FA) protocol based on ECC for mobile computing. Applied Sciences, 13(7), p.4425.

[2] Sasikumar, K. and Nagarajan, S., 2025. Enhancing Cloud Security: A Multi-Factor Authentication and Adaptive Cryptography Approach Using Machine Learning Techniques. IEEE Open Journal of the Computer Society.

[3] Zou, S., Cao, Q., Wang, C., Huang, Z. and Xu, G., 2021. A robust two-factor user authentication scheme-based ECC for smart home in IoT. IEEE Systems Journal, 16(3), pp.4938-4949.

[4] Ali, S. and Anwer, F., 2025. An IoT-Enabled Cloud Computing Model for Authentication and Data Confidentiality using Lightweight Cryptography. Arabian Journal for Science and Engineering, pp.1-23.

[5] Gupta, M., Ahuja, L. and Seth, A., 2023. Security enhancement in a cloud environment using a hybrid chaotic algorithm with multifactor verification for user authentication. International Journal of Computers and Applications, 45(11), pp.680-696.

[6] Miyazawa, C. and Sato, R., 2024. Biometric Fusion for Enhanced Authentication in Cloud Computing Environments. Advances in Engineering and Intelligence Systems, 3(01), pp.56-67.

[7] Velleangiri, V., Balasubramaniyam, S., Kumar, V.S., Subramanian, J., Hameed, F.S., Nagarajan, B., Sasikumar, V.A., Kumar, Y.S., Karthikeyan, S.S. and Ramesh, T.P.S., 2025, April. Hybrid cryptographic technique for secure spatial range query processing in cloud computing. In AIP Conference Proceedings (Vol. 3279, No. 1). AIP Publishing.

[8] Jayasankar, T., Bhavadharini, R.M., Nagarajanz, N.R., Mani, G. and Ramesh, S., 2021. Securing Medical Data using Extended Role Based Access Control Model and Two fish Algorithms on Cloud Platform. European Journal of Molecular and Clinical Medicine, 8(1), pp.1075-1090.

[9] Sebestyen, H., Popescu, D.E. and Zmaranda, R.D., 2025. A Literature Review on Security in the Internet of Things: Identifying and Analysing Critical Categories. Computers, 14(2), p.61.

[10] Dhinakaran, D., Srinivasan, L., Sankar, S.U. and Selvaraj, D., 2024. Quantum-based privacy-preserving techniques for secure and trustworthy internet of medical things an extensive analysis. Quantum Inf. Comput., 24(3&4), pp.227-266.

[11] Robert, Wamusi, Asiku Denis, Adebo Thomas, Aziku Samuel, Simon Peter Kabiito, Zaward Morish, and Guma Ali. "A Comprehensive Review on Cryptographic Techniques for Securing Internet of Medical Things: A State-of-the-Art, Applications, Security Attacks, Mitigation Measures, and Future Research Direction." Mesopotamian Journal of Artificial Intelligence in Healthcare 2024 (2024): 135-169.

Received: August 10 2025

Volume 38 No. 9s , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[12] Syed, N., Anwar, A., Baig, Z. and Zeadally, S., 2025. Artificial Intelligence as a Service (AIaaS) for Cloud, Fog and the Edge: State-of-the-Art Practices. ACM Computing Surveys.

[13] Attkan, A. and Ranga, V., 2022. Cyber-physical security for IoT networks: a comprehensive review on traditional, blockchain and artificial intelligence based key-security. Complex & Intelligent Systems, 8(4), pp.3559-3591.

[14] Al-Zubaidie, M. and Shyaa, G.S., 2023. Applying detection leakage on hybrid cryptography to secure transaction information in e-commerce apps. Future Internet, 15(8), p.262.

[15] Goyal, A., Matta, P., Maurya, S., Lohumi, Y. and Shelke, N., 2024, November. Strengthening Cloud Security: Exploration of Authentication Frameworks in Cloud Computing Environment. In 2024 4th International Conference on Technological Advancements in Computational Sciences (ICTACS) (pp. 1107-1112). IEEE.

[16] Karim, N.A., Abdulraheem, W.K., Khashan, O.A., Alazab, M., Kanaker, H., Farfoura, M.E. and Alshinwan, M., 2024. Performance Comparison of Hyper-V and KVM for Cryptographic Tasks in Cloud Computing. Computers, Materials & Continua, 78(2).

[17] Williams, P., Dutta, I.K., Daoud, H. and Bayoumi, M., 2022. A survey on security in internet of things with a focus on the impact of emerging technologies. Internet of Things, 19, p.100564.

[18] Ajlan, K.A., Alsboui, T., Alshaikh, O., Inuwa-Dute, I., Khan, S. and Parkinson, S., 2024. The Emerging Challenges of Wearable Biometric Cryptosystems. Cryptography, 8(3), p.27.

[19] Sumalatha, U., Prakasha, K.K., Prabhu, S. and Nayak, V.C., 2024. A comprehensive review of unimodal and multimodal fingerprint biometric authentication systems: Fusion, attacks, and template protection. IEEE Access.

[20] Ahmad, I., Qudus, F., Qadir, M., Shah, S., Atif, M., Islam, M. and Jan, S., 2023. Securing the Next Generation Cloud: A Survey of Emerging Technologies and their Impact on Cloud Security. The Sciencetech, 4(4).

[21] Yadav, K., Alharbi, A., Jain, A. and Ramadan, R.A., 2022. An IoT based secure patient health monitoring system. Computers, Materials and Continua, 70(2), pp.3637-3652.

[22] Gaikwad Vidya, S., Sable, N.P., Wankhede, D.S., Mishra, V., Karnik, M.P., Ambhore, N. and Manikjade, A., 2024. Securing cloud-based IoT. Internet of Things enabled Machine Learning for Biomedical Applications, p.273.

[23] Meenakshi, S. and Neha, B., 2023. Cloud System Performance and Security Improvements with Multi-Tenancy integration. Journal of Multimedia Technology & Recent Advancements, 10(02), pp.10-16.

[24] Shyaa, G.S. and Al-Zubaidie, M., 2023. Securing Transactions Using Hybrid Cryptography in E-commerce Apps. Journal of Education for Pure Science-University of Thi-Qar, 13(3).

[25] Suneetha, K., Patel, A. and Karuna, G., 2024. Reddy Madhavi K. Mohan Babu University Vijayasanthi Maddela Sree Vidyanikethan Engineering College. Innovations in Computational Intelligence, Big Data Analytics and Internet of Things, p.139.

Received: August 10 2025

Volume 38 No. 9s , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[26] Umamahesh, G., Rajkumar, G.V.S. (2023). Survey on Various Security Issues Associated with Cloud Authentication Techniques. In: Reddy, V.S., Prasad, V.K., Wang, J., Rao Dasari, N.M. (eds) Intelligent Systems and Sustainable Computing. ICISSC 2022. Smart Innovation, Systems and Technologies, vol 363.

[27] Umamahesh, G., Rajkumar, G.V.S. (2025). Data Security in Cloud Using Twofish Algorithm. In: Rama Sree, S., Kumar, S. (eds) Algorithms and Computational Theory for Engineering Applications. ICACTEA 2024. Advances in Science, Technology & Innovation.

[28] Govindaraju Kaladi & N. Sreenath (06 Aug 2025): Fuzzy Elliptic Curve Cryptography-Based Privacy-Preserving Message Authentication and PROMETHEE Trust Management Scheme for Fog-Enabled VANETs, IETE Journal of Research, DOI: 10.1080/03772063.2025.253195

[29] Govindaraju, K., Sreenath, N. Hyperelliptic curve signcryption based privacy preserving message authentication and fuzzy TOPSIS-based trust management scheme for fog enabled VANET. Int. j. inf. tecnol. 17, 247–261 (2025). https://doi.org/10.1007/s41870-024-02230-0

Received: August 10 2025