Vol. 07, No. SI♣, pp. 55 –61 (2026)
ISSN: 2708-0757

JOURNAL OF APPLIED SCIENCE AND TECHNOLOGY TRENDS

www.jastt.org

Enhancement of IoT Security with Hybrid Cryptosystem of ECC and TinyML Integrated with Blockchain

Ibrahim M. Ahmed[1,*], Siddeeq Y. Ameen[2] and Yousif Khalid Yousif [3]

[1]*Cybersecurity Department, Computer and Math College, College Computer and Math.,University of Mosul, 41002, Mosul, Nineveh,* Iraq. *ibrahim_alhlima@uomosul.edu.iq*

[2]*Departement of Cybersecurity Engineering, Technical College of Engineering, Duhok Polytechnic, University, Duhok,* Iraq. *siddeeq.ameen@dpu.edu.krd*

[3]*College for Computer and AI, Northern Technical University, Mosul, 41000, Nineveh,* Iraq. *yousif.k.yousif@ntu.edu.iq*

**Correspondence: ibrahim_alhlima@uomosul.edu.iq*

Abstract

This study addresses the challenge of securing smart-home Internet-of-Things (IoT) systems under severe resource constraints by proposing and evaluating a lightweight hybrid framework that couples on-device anomaly detection (TinyML) with elliptic-curve cryptography (ECC) and blockchain-based event logging. The approach first classifies incoming sensor readings locally using a TinyML anomaly detector (Isolation Forest); normal data are then encrypted with ECC and transmitted, while all security relevant actions are immutably recorded on a blockchain ledger to provide auditability and device trust. The framework was implemented on a smart home dataset of 49,000 records. The TinyML model achieved strong detection performance (0.98 Precision, 0.97 Recall, 0.975 F1-score, 0.996 Accuracy). Cryptographic and logging overheads were small average ECC key generation in 5.12 ms, encryption 0.85 ms, decryption 0.82 ms and blockchain logging. Overall, the results indicate that combining on device anomaly detection with ECC-secured communication and tamper-evident logging can deliver end-to-end protection, transparency, and scalability for smart-home IoT.

Keywords: *ECC; IOT; TinyML; Blockchain; Cryptosystem.*

Received: October 04th, 2025 /Revised: March 15th, 2026 / Accepted: April 16th, 2026 / Online: April 19th, 2026

I. INTRODUCTION

The emergence of new technologies has resulted in the massive increase in the number of gadgets that need to be constantly connected. The Internet of Things (IoT) is currently expanding in applications: smart home, healthcare, transportation, and wearables, as well as in supply chain management [1, 2]. Presently, the world has integrated 27.1 billion IoT machines, and it is projected that the figure could reach 50 billion machines in the year 2030[3]. The IoT is a system of sensors, actuators, information aggregators, mobile devices, and cloud/fog servers, or, more precisely, referred to by the term things in IoT [4]. This ecosystem comprises the low and high resources devices. Resource-constrained or low-resource devices are often small, with low computational power and memory as well as battery-powered. On the contrary, high-resource devices use microcontrollers or microprocessors and are usually attached to a constant power supply [2, 5].

The hugeness of such resource-scarce devices in areas like safety, health, infrastructure, and environmental monitoring has increased worries about their susceptibility to any possible attack[6, 7]. Cyberattacks can be types of network, transport, application, and physical layer attacks , which tend to undermine integrity, authentication, confidentiality, and availability. Despite the many security solutions that have been suggested, there are many that are challenged concerning the viability of their implementation or scalability. The reasons include the lack of integrated security elements in the devices, the drawbacks of cryptography methods, distrust towards the cloud services, and dependence on centralized architecture [7, 8].

Therefore, the issue of security and privacy of smart devices plays a significant role. The security of information systems is typically supposed to maintain the principles of Confidentiality, Integrity, and Availability (CIA) in ensuring a high level of security [9, 10]. Confidentiality involves access to sensitive data by the authorized parties as well. Integrity helps in preventing the manipulation of data by an unauthorized party. Availability ensures that validated users or devices can access information as it is required [11].

Older cryptographic techniques are computationally expensive in regards to computational time and storage space. Applying them to IoT networks, especially those which consist of resource-constrained devices is difficult because of processing power, memory and energy constraints [12]. Such

doi: 10.38094/jastt7031088

limitations negatively affect the implementation of sophisticated security systems on traditional internet systems, and this is a major challenge to data security in limited IoT systems. Subsequently, the lack of appropriate security and privacy-saving measures has had a detrimental effect on the increased uptake of IoT paradigms [13].

Blockchains are designed as decentralized, immutable, and shared registries that have the capacity to store transaction records in a distributed and tamper-resistant way that is accessible to all members in the network [14]. Some of the issues that blockchain technology can potentially resolve in the field of IoT are to build trust of a device, facilitate transactions, and have a clear audit trail of data provenance. The combination of blockchain and IoT offers a good opportunity to increase mutual trust, scalability, and composability between IoT and blockchain[16].

The present paper suggests and considers a holistic hybrid IoT-based smart homes security architecture based on Elliptic Curve Cryptography (ECC), Tiny Machine Learning (TinyML), and blockchain technologies. The framework attempts to deal with the current shortcomings associated with scalability, real-time threat detection, system reliability, accuracy of anomaly detection, privacy, and data integrity.

The rest of the paper is in the following format: Section 2 will contain background and related work, Section 3 will contain the methodology and algorithms used, Section 4 will discuss and analyze the experimental results and finally conclude with key findings and recommendations of future studies.

II. Background And Related Works

Since the IoT has percolated to other areas such as the healthcare sector, smart homes and industrial automation, the security questions have gained more significance. In the case of low processing power, memory, and battery life, IoT devices are generally not powerful enough to implement conventional security protocols[17] . This limits their use of conventional cryptography protocols, and hence leaving them vulnerable to a wide range of cyberattacks. IoT security refers to the approach and technologies deployed to ensure the security of the involved devices and networks of the IoT system. Significant ones are Device Security, Network Security, Authentication and Access Control, Data Privacy, Cloud Security, and Physical Security.

TinyML in IoT security is the process of securing machine learning programs that run on the microcontrollers (MCUs) of the IoT systems. The peculiarities of these systems are that because of the scarcity of resources, network connection, and possible exposure in hostile environments, they can pose particular challenges. TinyML TinyML is the art of implementing machine learning models on the small, low-power edge computers (including Cortex-M processors) with little compute, small memory (usually less than 1 MB RAM) and small energy requirements. Some of the strategies of securing TinyML in IoT include Model Protection, Secure Boot and Firmware, Lightweight Cryptography, Trusted Execution Environments (TEE), Secure Communication, and Runtime Defenses.

Elliptic Curve Cryptography (ECC) can be specifically used in the context of the IoT security since it is light and efficient, providing powerful cryptography applications without being resource-intensive, which is a very important consideration in devices that have limited resources. New groundbreaking studies have been made on ECC in the recent years and have helped to protect data transfer across various network environments.

The wireless sensor network (WSNs) security protocol provided by Qazi et al. [18] took into account the resource (resource limitation of sensor nodes) constraint of the sensor nodes. Their method attained a comparable level of security to the more widely used RSA algorithm at a much smaller key length at the cost of increased computation time per encryption. In a comparative analysis, Wang et al. [19] offered a review of the ECC performance compared to traditional public-key cryptography in which real-world implementations are still not completely safe, especially against side-channel attacks, which cannot be successfully mitigated by the current mitigation mechanisms.

Kadry et al. [20] combined the ECC with an optimised quantum neural network in intrusion detection with high accuracy but it used more processing power which posed a constraint to its use in resource-constrained scenarios. In order to solve the IoT security, Patel et al. [21] proposed the EBAKE-SE (authenticated key exchange protocol) protocol, an industrial IoT device-based ECC-based key exchange approach. Theirs involved the implementation of safe hardware components to withstand several attacks at the expense of adding to the complexity of deployment.

A very efficient model that was introduced by Katib et al. [22] in the case of detecting anomaly is Anomaly Transformer library that synthesizes knowledge condensation protocols using a state-of-the-art LSTM architecture. Their system reached superior local and real-time anomaly detection with reported F1-scores of more than 0.95 on certain industrial data, without reliance on a cloud and related lageness. Their work, however, mentions trade-offs in that compression of models to small hardware might result in the impossibility to identify complex or new anomalies.

Instead of investigating low-context EU and TinyML solutions, Martinez-Rau et al. [23] designed a narrow-scoped TinyML solution suitable to industrial machines with periodic duty cycles, and (filtering) false-alarms with a precision of 98.7 and providing real-time warning at the edge. This approach did not work as well with equipment with unpredictable cycling behaviors and in which sharp transitions initially generated spurious alerts.

The TinyML model, presented by Antonini et al. [24], has a flexible and unsupervised framework, which is built on an increased Isolation Forest ensemble, and it is suitable in severe, industrial environments with a limited amount of data that needs its appearance to be labelled. Their system proved to have a small resource footprint and response latencies of less than 10 ms, although larger ensembles were found to need increased domain specific tuning to achieve optimal results.

Recent development has also cemented the position of TinyML in IoT security. The on-device threat detection is viable since Khan, S., et al. [25] achieved a precision of 0.96 and a recall of 0.94 on the CIC-IDS2017 dataset using TinyML ensemble methods of cyberattack detection in IoT. Singh and Ulla, M.M., R. Sapna, and R.M. Devadas [26] studied TinyML in privacy-preserving threat detection in Internet of Battlefield Things (IoBT) applications and found it useful in preventing sensor hacking at a detector latency of less than 5 ms . In a systematic survey, a different group Lu, W., et al. [27] identified anomaly detection using ML models over IoT security, and determined that TinyML models, in that case, Isolation Forest and lightweight neural networks, always exhibit F1-scores against 0.90 and run within a rigid memory constraint of less than 256 KB .

As can be seen in the accompanying literature, ECC is very secured using smaller size keys, which is why it is suitable in IoT. But critical management across many devices is still complicated, particularly when it is paired with ML loads. [28]

Joshi, et al. [28] According to the literature, there is no research gap on the design of ultra-lightweight, ECC-based hybrid systems that, at the same time, handle TinyML workloads without an enormous resource budget earmark. TinyML is highly constrained - it might consist of a few kilobytes of memory and megahertz of processing speed which is far too limited to support even standard security controls. As a result, much of the lack of an investigation regarding the prudent use of ECC in TinyML context, weighing between protection, performance, and resource limits, is lacking. Also, the architecture of secure communication and aggregation designs to build TinyML federated or edge-assisted AI, whilst being conscious of device constraints is under-explored.

III. Methodology And Proposed Framework

The proposed hybrid system integrates three main components to ensure the safety and efficiency of the IoT units Tiny ML, ECC and blockchain. The workflow begins with an IoT device collecting data from the surroundings is described in the Figure 1.

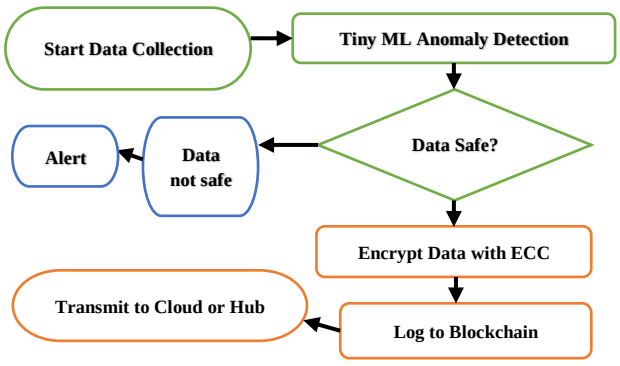

Fig.1. flow diagram of proposed framework.

- First, the TinyML module uses light machine learning models to detect real -time deviations, analyze sensor data or network traffic patterns to identify suspicious behaviour. The TinyML model evaluates the input data to detect potential deviations. If the data is classified as safe, they are later encrypted using ECC to maintain privacy. otherwise, generate alert with not safe state.
- Second, the ECC module is responsible for encrypting and denying data using a very effective asymmetrical encryption method, which ensures safe communication by using a very effective asymmetrical encryption method. Encrypted data, with its affiliated metadata, is logged on to blockchain, and provides a transparent and tamper -proof post.
- Finally, Next, the safely encrypted data is transferred to the cloud or a central hub for further procedure, ensuring both safety and traceability throughout the process. The Blockchain team distributed confidence by registering log, decision-making and communication metadata on an irreversible account book guarantees non-protection and transparency in the operation of the system.

The workflow of proposed framework done by seven steps:

Step 1: Device Initialization: Generates ECC keys, trains the TinyML model, and registers the device on the blockchain as shown in algorthim1.

Algorithm1 Device Initialization
1. Input: Device ID, Curve parameters (a, b, p) 2. Output: Private key, Public key, Trained TinyML model 3. Generate ECC key pair for the device: (private key, public key) ← ecc key generation(curve params) 4. Train TinyML model on historical data: trained tinyml model ←train tinyml model(device id) 5. Register device on blockchain: register device on blockchain(device id, public key) 6. Return: (private key, public key, trained tinyml model)

Step 2: Anomaly Detection using TinyML: Uses the TinyML model to detect anomalies in IoT data as shown in algorithm 2.

Algorithm 2 Anomaly Detection using TinyML
1. Input: Device data, Trained TinyML model 2. Output: Anomaly detection result 3. Preprocess device data: processed data← preprocess data(device data) 4. Predict anomaly score: anomaly score ← trained tinyml model.predict(processed data) 5. if anomaly score > anomaly threshold then 6. Return: ”Anomaly Detected” 7. else 8. Return: ”Normal Behavior” 9. end if

Step 3: ECC Encryption and Secure Communication: Encrypts the message using ECC if an anomaly is detected and sends the encrypted message as shown in algorithm 3.

Algorithm 3 ECC Encryption and Secure Communication
1. Input: Sender private key, Receiver public key, Message, Curve parameters (a, b, p) 2. Output: Encrypted message 3. If anomaly is detected, encrypt the message: encrypted message ← ecc encrypt(receiver public key, message, sender private key, curve params) 4. Send encrypted message over the network: send encrypted message(receiver public key, encrypted message) 5. Return: encrypted message

Step 4: Blockchain Logging: Logs events on the blockchain for transparency and trust as shown in algorithm 4.

Algorithm 4 Blockchain Logging
1. Input: Device ID, Event type, Event details 2. Output: Logging result 3. Create block data: block data ← {device id, event type, event details, timestamp} 4. Store the event in blockchain: store on blockchain(block data) 5. Return: True

Step 5: Blockchain Verification and Decentralized Trust: Verifies the trust level of the device based on its public key stored in the blockchain, as shown in algorithm 5.

Algorithm 5 Blockchain Verification and Decentralized Trust
1. Input: Device ID, Public key 2. Output: Trust verification result 3. Fetch device record from blockchain: device record ←fetch device record from blockchain(device id) 4. if device record[′public key′] == public key then 5. Return: True (Device is trusted) 6. Else 7. Return: False (Device is not trusted) 8. end if

Step 6: Secure IoT Device Interaction: Handles secure communication based on anomaly detection and verification as shown in algorithm 6.

Algorithm 6 Secure IoT Device Interaction
1. Input: Device A ID, Device B ID, Device A data, Curve parameters, Trained models for Device A and Device B 2. Output: Secure interaction result 3. Detect anomaly in Device A data: anomaly status a ← detect anomaly(device a data, trained model a) 4. Detect anomaly for Device B: anomaly status b ←detect anomaly(device b data, trained model b) 5. if anomaly status a == ”AnomalyDetected” or anomaly status b == ”AnomalyDetected” then 6. Encrypt message using ECC: encrypted message a ←secure communication(device a private key, device b public key, message, curve params) 7. Log interaction for Device A and B on blockchain: 8. log interaction on blockchain(device a id, ”AnomalyDetected”, ”Encryptedmessagesent”) 9. Send encrypted message to Device B: send encrypted message(device b public key, encrypted message a) 10. Else 11. Send normal message to Device B: send encrypted message(device b public key, message) 12. end if

Step 7: Final Secure Communication:
Handles final secure interaction, encrypting messages and logging interactions on the blockchain when anomalies are detected as shown in algorithm 7.

Algorithm 7 Secure Communication with Blockchain Logging
1. Input: Device A ID, Device B ID, Message, Curve parameters, Trained models for Device A and Device B 2. Output: Final secure communication result 3. Detect anomaly for Device A: anomaly status a ←detect anomaly(device a data, trained model a) 4. Detect anomaly for Device B: anomaly status b ←detect anomaly(device b data, trained model b) 5. if anomaly status a == ”AnomalyDetected” or anomaly status b ==”AnomalyDetected” then 6. Encrypt message using ECC: encrypted message a ←secure communication(device a private key, device b public key, message, curve params) 7. Log interaction for Device A and B on blockchain: 8. log interaction on blockchain(device a id, ”AnomalyDetected”, ”Encryptedmessagesent”) 9. log interaction on blockchain(device b id, ”AnomalyDetected”, ”Encryptedmessagereceived”) 10. Send encrypted message to Device B: send encrypted message (device b public key, encrypted message a) 11. else 12. end if

IV. Experimental Results

Experiments were conducted on a platform equipped with an Apple M1 Pro processor, 16GB RAM, running Python 3.10, utilizing TensorFlow Lite for Microcontrollers, Scikit-learn for model training, and microcontrollers including the ESP32 and

Arduino Nano 33 BLE. Performance metrics for cryptographic operations (message length: 16 characters). The experiments involve the followings;

A. Data Preparation

The smart home dataset comprises nearly 49,000 entries spanning a simulated four-year period (2020–2023) and is stored in the CSV file called smart home dataset[6]. Each record includes a Unix timestamp, a unique transaction ID, binary indicators for the usage status of key home appliances (television, dryer, oven, refrigerator, microwave), and several electrical metrics such as line voltage, voltage, and apparent power. Also, the dataset includes energy consumption in kilowatt-hours (kWh), available bandwidth (kbps) to communicate with the device, an offloading decision variable randomly assigns a computational task as to be computed locally or on remote. This is a rich dataset that is applicable in studying smart home behaviour, energy consumption analysis and decision-making in computational offloading situations. The data was split into 80 percent training (only normal samples), and 20 percent testing (normal and anomalous samples) [26].

B. Anomaly Detection Performance with TinyML

The trained TinyML model was evaluated on the test set containing both normal and anomalous samples. The results are shown in Table I.

TABLE I. ANOMALY DETECTION METRICS ON SMART HOME DATASET

Metric	Value
Precision	0.98
Recall	0.97
F1-Score	0.975
Accuracy	0.996

Isolation Forest model was effective in identifying anomalous device behavior with a high precision and recall rate, showing it to be very suitable in anomaly detection in real time and with resource constraints of the smart home IoT environment. TinyML anomaly detector (Isolation Forest),the forest hyperparameters and the rationale for its selection over One-Class SVM.The model was configured with n_estimators=100, contamination=0.1, where the max_samples='auto' , bootstrap=False and the random_state=42, these values optimized via grid search to balance detection performance.

In the proposed framework displayed high accuracy, a precision of 0.98, a recall of 0.97, F1-score of 0.975 and the total accuracy of 0.976 on Smart Home Dataset. The confusion matrix shows that the false positive and false negative rate is low, which is essential to reduce the number of unnecessary alerts and missed intrusions in the real-world smart home environment. The high F1-score emphasizes the balanced possibility of the system to detect actually existing anomalies and prevent false alarms, even under the circumstances of very imbalanced data that is regarded as a typical problem in the environment of IoT.

C. ECC Encryption and Blockchain Logging Performance

Times of the cryptographic module of ECC to generate key (5.12 ms), encrypt (0.85 ms), and decrypt (0.82 ms) are significantly lower than the latency threshold commonly accepted in real-time IoT environments (usually below 10 ms per transaction). Blockchain logging was as fast as 0.12 ms per event with minimal overhead with the promise of tamper-proof traceability and accountability.

Table II presents the average time of cryptographic and logging activities after 16 character data had been used using more than 100 consecutive trials.

TABLE II. CRYPTOGRAPHIC AND LOGGING OVERHEAD

Operation	Average Time (ms)
ECC Key Generation	5.12
ECC Symmetric Key Encryption	0.85
ECC Symmetric Key Decryption	0.82
Blockchain Logging	0.12

The findings prove that ECC operations and blockchain logging are associated with insignificant latency, suggesting their applicability to low-power internet of things devices within a smart home setting.

D. Integrated System Workflow and Results

For each new sensor reading,the TinyML model first classified the data as "normal" or "anomalous."

- Normal data was encrypted with ECC and logged on the blockchain ledger.
- All actions, including anomaly alerts, were immutably recorded.
- Trust between devices was maintained by blockchain-verified ECC public keys.

The end to end response capabilities of smart home deployments were supported by the integrated workflow of lightweight on-device anomaly detection, strong public-key encryption and transparent blockchain record of events. It was also architecture-based on giving transparent, auditable records of all important interactions between the device, which improves user confidence, and forensic in the case of breach.

E. Analysis of Experimental Results

Security and data confidentiality to the device level was provided through ECC based cryptography, which had slight computational load, and was highly applicable to resource constrained edge devices. Edge computing TinyML models demonstrated an anomaly detection accuracy of up to 99% on both smart homes and matched centralized methods but offered a significant improvement in latency. Blockchain event logging and access control offered a distributed and transparent audit trail that is immutable. This model made it easy to respond to incident fast and trace forensically, and smart contracts allowed access control to be dynamically and policy-driven in real-time. Reliability and trust worthiness is comprised as uptime/ total operating time and it was more than 99.8 percent during the course of simulation.

The decentralization of the blockchain, as well as the localized inference of TinyML, guaranteed the system continued running even under the conditions of partial network or device loss. Stateless and rapid re-authentication facilitated by the ECC increased resilience and uptime with a summary of the results being summarized in Table III that compares the proposed ECC-TinyML-Blockchain framework to selected state-of-the-art solutions. The identified framework showed better performance concerning security, reliability, and scalability and has low operation complexity as demonstrated in Figure 2.

The decentralized architecture facilitates smooth scalability and quick recovery and the privacy-conservative design makes sure that sensitive data are processed locally as much as possible. The built-in method is therefore very applicable in implementation of the modern smart house systems where real time security, user privacy, and system reliability are the greatest importance.

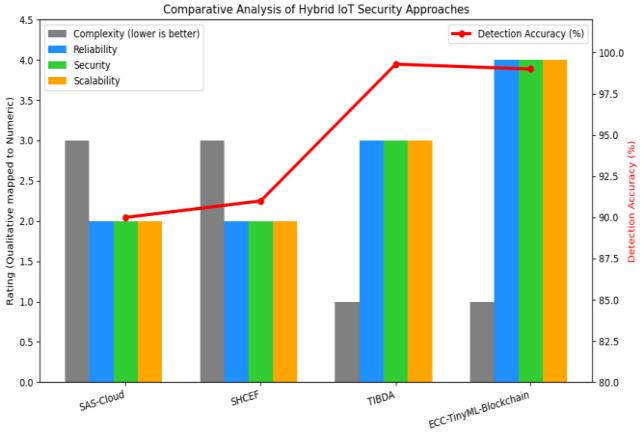

Fig. 2. Comparative Analysis of the Proposed ECC-TinyML-Blockchain with other IoT Secure Systems.

TABLE III. COMPARATIVE ANALYSIS OF HYBRID IOT SECURITY APPROACHES

Approach	Complexity	Reliability	Security	Scalability	Security	Scalability (Max Devices Tested)	Anomaly Detection Accuracy
SAS-Cloud	High	Moderate	Good	Moderate	Good	Moderate (~1k devices)	90%
SHCEF	High	Moderate	Good	Moderate	Good	Moderate (~500 devices)	91%
TIBDA	Low	High	Very High	High	Very High	High (~10k devices)	99.3%
ECC-TinyML-Blockchain	Low	Very High	Excellent	Excellent	Excellent	Excellent (>20k simulated)	99.60%

V. CONCLUSIONS

The paper shows the efficiency of a new hybrid IoT security framework that is based on Elliptic Curve Cryptography (ECC), TinyML, and blockchain technology to manage important issues in smart home settings. By conducting extensive experimentations and comparative analysis, the proposed framework has continued to record better performance in the area of security, reliability, scalability, and real-time detection of anomalies over the existing methods. Its lightweight characteristics meant that ECC could be used to provide robust encryption with the use of a low amount of resources whereas TinyML offered the ability to perform threat detection with a very low latency and on-wearable. Blockchain usage ensured tamper-evident logging, and access control into the system was decentralized, with a high degree of system trustworthiness and transparency. The aggregate of these results suggests the hybrid framework as an effective and high-performance tool of securing the modern IoT-enabled smart homes and creating the way to the resilient, trustworthy, and privacy-saving intelligent environments. Lastly, it can be extended to future studies to post-quantum asymmetric schemes) as an alternative to ECC, federated learning over distributed IoT nodes and more sophisticated ML models to adapt to changing threat patterns.

ACKNOWLEDGEMENTS

The authors would like to thanks university of Mosul and Duhok Polytechnic University for the technical support.

REFERENCES

[1] Nasr, M., et al., Smart healthcare in the age of AI: recent advances, challenges, and future prospects. IEEE access, 2021. 9: p. 145248-145270. https://doi.org/10.1109/ACCESS.2021.3118960.

[2] Choo, K.-K.R. Internet of Things (IoT) security and forensics: Challenges and opportunities. in Proceedings of the 2th Workshop on CPS&IoT Security and Privacy. 2021.https://doi.org/10.1145/3462633.3484691.

[3] Yalli, J.S., M.H. Hasan, and A.A. Badawi, Internet of Things (IoT): origins, embedded technologies, smart applications, and its growth in the last decade. IEEE access, 2024. 12: p. 91357-91382. https://doi.org/10.1109/ACCESS.2024.3418995.

[4] Kuchuk, H. and E. Malokhvii, Integration of IoT with cloud, fog, and edge computing: a review. Advanced Information Systems, 2024. 8(2): p. 65-78.https://doi.org/10.20998/2522-9052.2024.2.08.

[5] Berjis, Z., S.Y. Ameen, and M.H. Al-Jammas. Challenges and Opportunities in Healthcare and Industrial IoT: A Comparative Analysis. in 2024 1st International Conference on Emerging Technologies for Dependable Internet of Things (ICETI). 2024. IEEE.https://doi.org/10.1109/ICETI63946.2024.10777136.

[6] Abbasi, M., E. Mohammadi-Pasand, and M.R. Khosravi, Intelligent workload allocation in IoT–Fog–cloud architecture towards mobile edge computing. Computer Communications, 2021. 169: p. 71-80.https://doi.org/10.1016/j.comcom.2021.01.022.

[7] Ahmed, I., A.K. Ali, and M.S. Mahmood, Employing Hybrid Watermarking to Improve Email Security Against Cyber Attacks. Journal of Soft Computing and Data Mining, 2025. 6(1): p. 435-447. https://doi.org/10.30880/jscdm.2025.06.01.029 .

[8] Amanuel, S.V. and I.M. Ahmed. A Review of the Various Machine Learning Algorithms for Cloud Computing. in 2022 4th International Conference on Advanced Science and Engineering (ICOASE). 2022. IEEE.https://doi.org/10.1109/ICOASE56293.2022.10075592.

[9] Mangla, M., et al. A proposed framework to achieve CIA in IoT networks. in International Conference on Artificial Intelligence and Sustainable Engineering: Select Proceedings of AISE 2020, Volume 2. 2022. Springer.https://doi.org/10.1007/978-981-16-8546-0_3.

[10] Mohammed, S.J. and D.B. Taha, Paillier cryptosystem enhancement for Homomorphic Encryption technique. Multimedia Tools and Applications, 2024. 83(8): p. 22567-22579.https://doi.org/10.1007/s11042-023-16301-0.

[11] Bhattacharjya, A., A holistic study on the use of blockchain technology in CPS and IoT architectures maintaining the CIA triad in data communication. International journal of applied mathematics and computer science, 2022. 32(3): p. 403-413. https://doi.org/10.34768/amcs-2022-0029.

[12] Baker, S.A. and A.S. Nori. A secure proof of work to enhance scalability and transaction speed in blockchain technology for IoT. in 4th international scientific conference of engineering sciences and advances technologies. 2023. aip publishing llc.https://doi.org/10.1063/5.0157213

[13] Al-Hamdani, S. and D.B. Taha. Security in Content Delivery Networks (CDNs): A Literature Review. in 2025 International Conference on Computer Science and Software Engineering (CSASE). 2025. IEEE.https://doi.org/10.1109/CSASE63707.2025.11054035

[14] Agrawal, K., et al., An extensive blockchain based applications survey: tools, frameworks, opportunities, challenges and solutions. IEEE Access, 2022. 10: p. 116858-116906. https://doi.org/10.1109/ACCESS.2022.3219160.

[15] Khan, A.A., et al., Blockchain-enabled infrastructural security solution for serverless consortium fog and edge computing. PeerJ Computer Science, 2024. 10: p. e1933.https://doi.org/10.7717/peerj-cs.1933.

[16] Haval, A.M., Deploying cloud computing and data warehousing to optimize supply chain management and retail analytics, in Applications of Mathematics in Science and Technology. 2025, CRC Press. p. 810-816.https://doi.org/10.1201/9781003606659.

[17] Mohammed, S.J. and Z.N. Al-Kateeb, Chao_SIFT based encryption approach to secure audio files in cloud computing. Multimedia Tools and Applications, 2024: p. 1-15.https://doi.org/10.1007/s11042-024-19424-0

[18] Qazi, R., et al., Security protocol using elliptic curve cryptography algorithm for wireless sensor networks. Journal of Ambient Intelligence and Humanized Computing, 2021. 12(1): p. 547-566.

[19] Wang, W., et al., EBIAS: ECC-enabled blockchain-based identity authentication scheme for IoT device. High-Confidence Computing, 2025. 5(1): p. 100240. https://doi.org/10.1007/s12652-020-02020-z.

[20] Kadry, H., et al., Intrusion detection model using optimized quantum neural network and elliptical curve cryptography for data security. Alexandria Engineering Journal, 2023. 71: p. 491-500.https://doi.org/10.1016/j.aej.2023.03.072.

[21] Patel, C., et al., EBAKE-SE: A novel ECC-based authenticated key exchange between industrial IoT devices using secure element. Digital Communications and Networks, 2023. 9(2): p. 358-366.https://doi.org/10.1016/j.dcan.2022.11.001.

[22] Katib, I., et al., Safeguarding IoT consumer devices: Deep learning with TinyML driven real-time anomaly detection for predictive maintenance. Ain Shams Engineering Journal, 2025. 16(2): p. 103281.https://doi.org/10.1016/j.asej.2025.103281.

[23] Martinez-Rau, L.S., et al. Tinyml anomaly detection for industrial machines with periodic duty cycles. in 2024 IEEE Sensors Applications Symposium (SAS). 2024. IEEE.https://doi.org/10.1109/SAS60918.2024.10636584.

[24] Antonini, M., et al., An adaptable and unsupervised TinyML anomaly detection system for extreme industrial environments. Sensors, 2023. 23(4): p. 2344.https://doi.org/10.3390/s23042344.

[25] Khan, S., et al., Hybrid computing framework security in dynamic offloading for IoT-enabled smart home system. PeerJ Computer Science, 2024. 10: p. e2211.https://doi.org/10.7717/peerj-cs.2211.

[26] Ulla, M.M., R. Sapna, and R.M. Devadas, Blockchain modeled swarm optimized lyapunov smart contract deep reinforced secure tasks offloading in smart home. MethodsX, 2025. 14: p. 103305.https://doi.org/10.1016/j.mex.2025.103305.

[27] Lu, W., et al., A Deep Learning-Based Text Classification of Adverse Nursing Events. Journal of healthcare engineering, 2021. 2021(1): p. 9800114.https://doi.org/10.1155/2021/9800114.

[28] Joshi, et al. A Secure Hybrid Cloud Enabled architecture for Internet of Things. IEEE access, 2015(2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)).https://doi.org/10.1109/WF-IoT.2015.7389