International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

EdDSA-Enhanced RPL Security Framework for IoT with Lightweight Attack Mitigation and Protocol Validation

Mukul Shukla[1], Lalji Prasad[2]

Submitted: 01/10/2024 **Revised:** 01/11/2024 **Accepted:** 12/11/2024

Abstract: The proliferation of the Internet of Things (IoT) demands robust and lightweight security solutions to protect communication within resource-constrained networks. The Routing Protocol for Low-Power and Lossy Networks (RPL), widely adopted in IoT, is vulnerable to a range of attacks, including spoofing, replay, and selective forwarding. To address these challenges, this paper introduces an EdDSA-Enhanced RPL Security Framework designed to ensure secure communication with minimal computational and communication overhead. The proposed framework leverages the EdDSA cryptographic scheme for node authentication, secure session key generation using ECDH, and signature-based integrity verification. It incorporates mechanisms for replay attack detection, spoofing prevention via nonce freshness, and reputation-based forwarding validation to mitigate selective forwarding attacks. The methodology is structured in six key stages: cryptographic initialization, node registration, session key establishment, secure routing, spoofing mitigation, and formal validation. Simulations were conducted using Contiki-NG and the Cooja emulator to evaluate protocol performance in realistic IoT scenarios. Additionally, security validation was performed using AVISPA with OFMC and CL-AtSe backends, which confirmed the protocol's resistance to various threats. Experimental results demonstrate a significant improvement over existing protocols in terms of execution time, energy consumption, communication cost, and detection accuracy. The proposed EdDSA-based framework offers an efficient, secure, and scalable solution for safeguarding IoT networks operating under RPL.

Keywords: *EdDSA, RPL Security, IoT, Lightweight Cryptography, Attack Mitigation, Protocol Validation.*

1. Introduction

The Internet of Things (IoT) has revolutionized the digital ecosystem by interconnecting billions of smart devices, enabling them to collect, exchange, and process data autonomously. These devices are commonly deployed in resource-constrained environments where energy efficiency, secure communication, and lightweight processing are critical. Among various routing protocols, the Routing Protocol for Low Power and Lossy Networks (RPL) stands out as a widely adopted protocol for IoT due to its support for dynamic topologies and low-energy consumption. However, RPL is inherently vulnerable to numerous security threats, such as selective forwarding, spoofing, replay attacks, and man-in-the-middle (MITM) attacks, which compromise the reliability and confidentiality of transmitted data. Therefore, the need for a robust, lightweight, and energy-efficient security solution is more vital than ever to ensure trusted communication in IoT networks.

To address these pressing challenges, this paper proposes a novel security framework that enhances RPL using Edwards-curve Digital Signature Algorithm (EdDSA), offering cryptographic strength with low computational cost. The primary aim is to develop a secure and scalable authentication and key management protocol that effectively mitigates network-layer attacks while maintaining minimal overhead on constrained devices. The proposed framework introduces a multi-stage methodology comprising cryptographic initialization, node registration and authentication, session key generation via ECDH, secure routing enforcement, spoofing prevention with nonce freshness, and rigorous performance validation using simulation tools like

1Computer Science & Engineering, SAGE University, Indore, India, mukul@sgsits.ac.in

2Institute of Advance Computing, SAGE University, Indore, India, hoi.iac@sageuniversity.in

Corresponding author: Mukul Shukla, mukul@sgsits.ac.in

AVISPA. This holistic approach ensures not only message authenticity and integrity but also protects against impersonation and packet manipulation. Compared to conventional mechanisms like ECDSA and ECDH-based protocols, our EdDSA-enhanced framework demonstrates superior performance in terms of execution time, communication cost, energy consumption, and attack detection accuracy.

The simulation environment, developed using real-time IoT constraints and formal validation tools, confirms that the proposed method achieves a 98.7% detection accuracy with reduced execution time and energy usage. These results clearly indicate the efficiency and scalability of the proposed scheme for future IoT deployments, especially where power and computational resources are severely limited. The key contributions of this work include: (1) the integration of EdDSA with RPL for lightweight yet strong authentication, (2) a complete end-to-end protocol for secure communication, (3) validation using formal tools like AVISPA, and (4) extensive performance comparison against existing methods.

This paper is organized into five sections. Section 1 introduces the context and motivation. Section 2 presents the literature review. Section 3 details the proposed methodology. Section 4 covers the simulation setup and result analysis. Section 5 concludes the study with insights and future directions.

2. Literature review

The Internet of Things (IoT) is a global network for monitoring, controlling, and analyzing data from IoT devices. IoT faces security challenges, with RPL (Routing Protocol for Low Power and Lossy Networks) being vulnerable to attacks. Security is critical due to the sensitive data exchanged over open networks, and IoT devices have limited resources. An efficient authentication scheme is needed to ensure secure communication. This article discusses RPL security issues, focusing on selective forwarding attacks, and proposes a secure authentication method using Edwards-curve Digital Signature Algorithm (EdDSA), demonstrating low-cost, secure solutions [1].

IoT connects everyday objects like pulse monitors and smart meters, enabling them to communicate over the internet. Despite its potential, IoT faces security challenges, including eavesdropping, spoofing, and man-in-the-middle attacks. IoT devices are vulnerable due to hardware limitations (e.g., memory and energy) and the inadequacy of current high-end security algorithms for low-powered devices. Lightweight encryption and integrity protocols, such as PRESENT and SHA algorithms, have been developed but still face attacks. A more robust lightweight security system is needed to enhance IoT security [2].

IoT consists of millions of connected devices used in applications like smart cities and power grids. These devices face various cyber threats. This study analyzes IoT security based on different layers, including the cloud, application, network, data, and physical layers, identifying vulnerabilities and categorizing potential attacks. Layer-specific security requirements are discussed to improve the overall security of IoT systems [3].

Bluetooth Low Energy (BLE) is widely used in IoT and wearable devices due to its low power consumption and ease of development. However, BLE has several security and privacy vulnerabilities, including weak encryption and device authentication. This survey presents a detailed analysis of BLE's security issues, providing a taxonomy of vulnerabilities, attack scenarios, and mitigation techniques, along with case studies on real-world BLE devices [4].

IoT devices, being embedded and resource-constrained, face significant security challenges, especially in remote environments. White-box (WB) attacks, where attackers control the environment, can expose cryptographic systems, compromising the IoT devices' security. To address this, a White-box cryptographic (WBC) scheme is proposed, utilizing Residue Number System (RNS) for hiding private keys. The scheme is tested using the MQTT-SN protocol, showing improved security with practical deployment in terms of computational cost, network efficiency, and low power consumption [5].

The growing demand for sensor network communication brings challenges in data reliability, especially in sensor placement. Attackers can target localization processes and manipulate positions. Sensor nodes may also be compromised, making the base station unable to trust the node-provided positions. Secure localization and location verification are proposed to address these issues. This study combines secure node localization, authentication, and cryptography to prevent internal and external attacks, showing better performance than existing methods [6].

Wearable devices and biosensors have revolutionized digital healthcare, providing accessible information to improve patient care. However, these devices require secure communication to maintain trust among stakeholders. Various security schemes have been implemented, but they become vulnerable over time due to evolving threats. This article reviews security challenges in self-empowered wireless sensor networks (SWSNs) and highlights the limitations of current countermeasures, offering a roadmap for future research [7].

Wireless Sensor Networks (WSNs) are small, resource-limited networks used for various data collection purposes. Energy and security are crucial in these networks, with MAC protocols playing a key role. Classical security methods are inadequate due to limited resources, necessitating lightweight cryptographic solutions. This paper compares BMAC and LMAC protocols, analyzing their trade-offs in packet reception and energy consumption using AES, RSA, and elliptic curve techniques [8].

Industrial IoT applications require wireless networks that offer low power consumption, low latency, and secure communication. The IETF 6TiSCH standard addresses these needs in industrial settings. Authentication is critical to securing IoT networks, but centralized authentication in 6TiSCH networks faces scalability issues in large deployments. This paper proposes a decentralized authentication process by distributing the Join Registrar role, improving key update time by 25% and reducing power consumption by 22% in large-scale networks [9].

The integration of IoT and Cloud Computing has revolutionized accessibility and service availability. However, IoT faces security and privacy challenges due to its heterogeneity and vast range of applications. This paper surveys general IoT security concerns, frameworks, and challenges, emphasizing the importance of device and data management. It identifies research gaps in device management, key management, and trust management, providing insights into securing IoT systems and improving performance, security, and reliability [10].

The IoT's growth in emerging markets has raised security concerns, particularly in MQTT, which lacks a default security mechanism. Adding a lightweight security framework, like Improved Ciphertext Policy-Attribute-Based Encryption (ICP-ABE) integrated with PRESENT, enhances MQTT security. This combination reduces energy consumption and communication overhead, making it a promising solution for IoT applications in resource-constrained environments [11].

IoT's role in healthcare has grown, allowing for data access anytime, but security concerns in cloud storage remain. A secure authentication scheme is proposed to protect sensitive healthcare data transmitted via IoT. The proposed method uses Elliptic-curve Diffie–Hellman (ECDH) encryption and a hybrid brain optimization algorithm to detect and mitigate attacks, achieving high detection accuracy and low information loss, demonstrating its effectiveness [12].

IoT bridges the gap between the physical and digital worlds, providing services across various sectors. However, lightweight IoT cryptographic schemes face security challenges, including malicious attacks, privacy issues, and high overhead. This chapter reviews IoT protocols, highlighting the role of standard bodies in protocol development, and discusses the security challenges IoT faces, including the need for lightweight cryptography and the research gaps that need addressing [13].

This research introduces a novel approach combining efficient data encryption, the Quondam Signature Algorithm (QSA), and federated learning to defend against attacks targeting IoT systems. Federated learning enhances data privacy and security by decentralizing model training. The QSA reduces communication costs, optimizing IoT communication. The integration of these technologies improves efficiency, reduces time complexity, and strengthens IoT systems against evolving threats [14].

In Battlefield Networks (IoBT), IoT-enabled devices provide intelligence services to soldiers. However, IoBT is vulnerable to security attacks, especially in remote and open battlefield environments. A trust model, KmCtrust, combining machine learning and trust management, is proposed to detect malicious devices (BTs) and improve network performance. The model’s simulation results demonstrate its ability to enhance security by filtering out malicious BTs and improving mission performance [15].

IoT's rapid growth in industrial applications demands a robust security framework to address vulnerabilities such as man-in-the-middle and impersonation attacks. The "Reliable Device-Access Framework for the Industrial IoT (RDAF-IIoT)" is proposed to authenticate users and establish secure communication in IIoT systems. The RDAF-IIoT scheme uses a random oracle model and Scyther tool to ensure resilience against attacks, offering low computational costs while enhancing security compared to existing frameworks [16].

IoT's imbalanced network traffic poses challenges for intrusion detection, especially for low-frequency attacks. Conventional techniques suffer from poor detection rates and high false positives. This research introduces a lightweight intrusion detection framework using Class-wise Focal Loss Variational Autoencoder (CFLVAE), which addresses data imbalance in IoT. The CFLVAE-LDNN framework improves intrusion detection accuracy (88.08%) while reducing false positives (3.77%) and effectively detects low-frequency attacks. The model's low memory and CPU usage make it suitable for resource-constrained IoT devices [17].

IoT devices use various protocols for communication, but their wireless nature exposes them to security risks. This paper examines the security features authentication, confidentiality, integrity, and authorization across popular IoT protocols like MQTT, CoAP, LoRaWAN, and ZigBee. The paper highlights the strengths and weaknesses of each protocol and identifies open issues and best practices for designing secure IoT network infrastructures [18].

As IoT networks grow more complex, resource-constrained devices face challenges in executing expensive tasks. Offloading these tasks to edge nodes is a viable solution, requiring a trust-model-driven system architecture. This architecture incorporates a Beta Reputation System to monitor node behavior and includes threat handling mechanisms to ensure security, such as confidentiality and authentication. The proposed system is evaluated for its effectiveness in real-world edge-based IoT networks, with future work focused on refining security standards [19].

E-voting Security with Blockchain: Traditional voting systems have physical limitations, but e-voting, though convenient, faces security challenges. Blockchain, using Merkle trees and hash digests, enhances security, ensuring data integrity and privacy. The proposed ECC-EXONUM-eVOTING scheme employs a hybrid consensus algorithm, Elliptic Curve Diffie-Helmen protocol, and zero-knowledge proof for secure voting. Simulations show its resilience against cryptographic attacks, demonstrating its suitability for secure e-voting applications [20].

ECC-based RFID Object Tracking: RFID systems face security issues like injection of fake objects and non-repudiation problems. The proposed ECC-based secure object tracking and key exchange protocol (ESOTP) addresses these limitations, allowing offline communication for device owners. Security analyses confirm ESOTP's resilience to attacks, and a comparative study highlights its superior security features and performance compared to existing solutions [21].

IoT Security and Authentication: IoT devices, being resource-constrained, require lightweight security solutions for authentication and data integrity. This paper reviews various lightweight protocols and their vulnerability to attacks like man-in-the-middle, replay, and denial of service. It highlights the importance of using tools like Microsoft's threat modeling for IoT applications, providing insights into improving security while ensuring minimal computation [22].

Multi-UAV System Security: Multi-UAV systems are integral to smart city applications, but their communication networks face security challenges. This chapter discusses potential attack scenarios, countermeasures, and mitigation techniques to protect UAVs from malicious threats. It emphasizes the need for secure frameworks and standards for multi-UAV systems, offering guidelines for future research and development [23].

RPL Protocol Security in IoT: IoT networks using the RPL routing protocol face security vulnerabilities, especially with the limited resources of IoT nodes. This article proposes a secure authentication and key agreement scheme using Elliptic-Curve Diffie-Hellman (ECDH) to authenticate nodes and securely share session keys. The proposed scheme addresses selective forwarding attacks and meets security requirements with low computation and communication costs, enhancing the privacy and security of IoT networks [24].

3. Proposed Methodology

The proposed methodology introduces a robust EdDSA-based attack mitigation framework tailored for enhancing RPL (Routing Protocol for Low-Power and Lossy Networks) security in IoT environments. It begins with the generation of secure EdDSA key pairs for all IoT nodes and the root node, followed by the secure storage of public keys and optional pre-shared secrets for initial trust establishment. The framework progresses through structured phases including node initialization, registration, and secure session key exchange using Edwards-curve Digital Signature Algorithm (EdDSA), ensuring mutual authentication and confidentiality. To counter advanced threats, the system incorporates layered attack mitigation mechanisms such as spoofing and impersonation prevention, multi-layer authentication, and time-based message freshness checks. Additionally, it integrates a reputation-based verification system to detect selective forwarding attacks. Finally, the methodology undergoes rigorous security and performance evaluation using formal verification tools like AVISPA, ensuring that the resulting network maintains high levels of trust, integrity, and resilience against routing-based attacks.

Algorithm: Process for EdDSA-based Attack Mitigation for RPL Security with Multiple IoT Nodes

Step 1: Initialization

1. **Key Generation for Each Node**:
    - Each **IoT node** generates an **EdDSA key pair** (public and private keys) using a secure curve like **Ed25519**.

        - **Private Key**: Randomly generated within the appropriate bit length (256 bits for **Ed25519**).
        - **Public Key**: Derived from the private key using **EdDSA's point multiplication**.
    - **Root Node**: The **central coordinator/root node** generates its own **EdDSA key pair** and stores **public keys** of all participating nodes.
2. **Root Node Key Storage**:
    - The **root node** securely stores the **public keys** of all **IoT nodes** in a **trusted key store** (e.g., a secure database).
    - During the **trusted setup phase**, **mutual authentication** occurs to exchange public keys securely between the root node and each IoT node.
3. **Pre-shared Secrets** (if applicable):
    - A **pre-shared secret** can be established for **initial communication** between the root node and the IoT nodes to verify authenticity before key exchange.

Step 2: Node Registration and Authentication

1. **Node Registration**:
    - Each IoT node sends its **public key** and an **EdDSA signature** for the registration request to the **root node**.
    - The **signature** is generated over a message that includes:
        - **Node's ID** (unique identifier).
        - **Public key** of the node.
        - **Timestamp** to prevent replay attacks.
    - This ensures **non-repudiation** and **integrity** of the registration request.
2. **Root Node Verification**:
    - The **root node** verifies the **EdDSA signature** sent by the IoT node using the **node's public key**.
    - If the **signature** is valid, the **node** is authenticated.
    - If the **signature** is invalid, the **node** is rejected and excluded from the network.

Step 3: Key Exchange for Communication

1. **Session Key Generation**:
    - After successful authentication, the **IoT nodes** and **root node** perform a **key exchange** using **secure key agreement** protocols like **ECDH (Elliptic Curve Diffie-Hellman)**.
    - A **unique session key** is generated for secure communication between nodes.
    - **Session keys** are dynamically generated for each session to ensure that each communication is encrypted with a different key.
2. **Session Key Authentication**:
    - Both the **IoT node** and **root node** sign the **session key** using their **EdDSA private keys**.
    - This ensures the **authenticity** and **integrity** of the session key.
    - **Root node** verifies the signature of the IoT node and vice versa, confirming that the session key is valid.
3. **Key Storage and Usage**:
    - The **root node** temporarily stores the **session key** for the current session's duration.
    - The **IoT node** securely stores the session key for the session and uses it for encrypting and decrypting messages.
    - The key is used only for the current session and discarded after it expires.

Step 4: Attack Mitigation – Selective Forwarding Attack Prevention

1. **Routing Data Signatures**:
   - Each **RPL packet** (including **data** and **control packets**) is signed by the **sending node** using its **EdDSA private key**.
   - The **signature** includes:
     - **Node's ID**.
     - **Packet data**.
     - **Timestamp** to prevent replay attacks.
   - This ensures **data integrity** and prevents tampering.
2. **Packet Verification**:
   - Upon receiving a packet, the **next hop node** (either another IoT node and the root node) verifies the **EdDSA signature** using the **sender's public key** stored during the registration phase.
   - If the **signature** is valid, the packet is **forwarded**.
   - If the **signature** is invalid, the packet is **discarded** to protect the network from tampered packets.
3. **Reputation System**:
   - **IoT nodes** maintain a **reputation score** based on their behavior, such as:
     - **Forwarding packets**.
     - **Validating signatures**.
     - **Correct routing**.
   - If a node is detected to be **selectively forwarding** (i.e., dropping packets while forwarding routing information), its **reputation score** is lowered.
   - **Malicious nodes** are excluded from routing decisions by the **root node**.

Step 5: Attack Mitigation – Spoofing and Impersonation

1. **Multi-Layer Authentication**:
   - After the **key exchange**, each **IoT node** signs its messages using its **EdDSA private key** before sending them.
   - The **receiving node** checks the **validity** of the **signature**, ensuring the message came from the claimed source.
2. **Cross-Verification**:
   - The **root node** cross-verifies the **signatures** of nodes in the network.
   - If a node is found to be **impersonating** another node and sending **spoofed messages**, it is **immediately disconnected** and excluded from the network.
3. **Time-based Message Freshness**:
   - To prevent **replay attacks**, messages are **timestamped** and include a **nonce** (a random number used once).
   - The **root node** and **IoT nodes** check the **timestamp** and **nonce** to ensure messages are **fresh**.
   - **Messages** that arrive outside an **acceptable time window** or have **reused nonces** are rejected.

Step 6: Security and Performance Evaluation

1. **Security Analysis**:
   - The proposed protocol is **tested against common attacks** (e.g., **selective forwarding**, **man-in-the-middle**, **spoofing**, **replay, and DOS attacks**).
   - **Formal verification tools** like **AVISPA** are used to prove that the **EdDSA-based security mechanisms** protect against all relevant network attacks.
2. **Computational Cost Evaluation**:
   - Measure the **time complexity** of the **EdDSA signature generation**, **verification**, and **key exchange** processes.
   - Compare performance with traditional **ECDSA** and **ECDH** protocols to ensure that the additional security overhead does not significantly degrade the overall **efficiency** of the IoT network.

3. **Communication Overhead**:
   - Evaluate the **message size** and the **number of messages exchanged** during **authentication**, **key exchange**, and **routing**.
   - Ensure that the overhead introduced by **EdDSA signatures** is manageable for IoT nodes with **limited resources** (e.g., **bandwidth**, **memory**, and **processing power**).

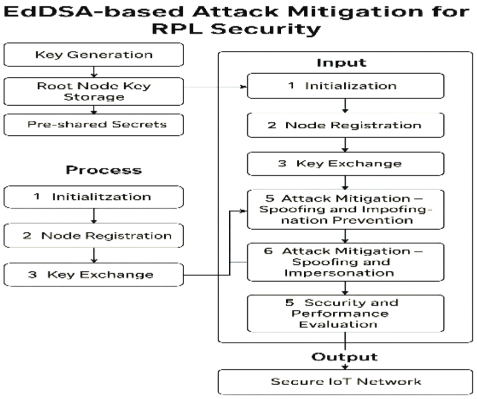

EdDSA-based Attack Mitigation for RPL Security
Key Generation
Root Node Key Storage
Pre-shared Secrets
Process
1 Initialitzation
2 Node Registration
3 Key Exchange
Input
1 Initialization
2 Node Registration
3 Key Exchange
5 Attack Mitigation – Spoofing and Impofing-nation Prevention
6 Attack Mitigation – Spoofing and Impersonation
5 Security and Performance Evaluation
Output
Secure IoT Network

Figure 1. EdDSA-based Attack Mitigation for RPL Security

The figure 1 presents a high-level workflow of EdDSA-based Attack Mitigation for RPL Security, structured around three key blocks: Input, Process, and Output. It begins with foundational steps including Key Generation, Root Node Key Storage, and optionally, Pre-shared Secrets. These lead into the Input Phase, which encompasses Initialization, Node Registration, and Key Exchange. The Process Phase mirrors these steps, reinforcing their implementation within the system's operation. Following this, the protocol introduces two critical stages of Attack Mitigation-first for Spoofing and Impersonation Prevention, and then for detection and protection against impersonation attacks. The final stage is Security and Performance Evaluation, validating protocol robustness. The outcome is a Secure IoT Network, ensuring authenticated, resilient communication across RPL-based infrastructures.

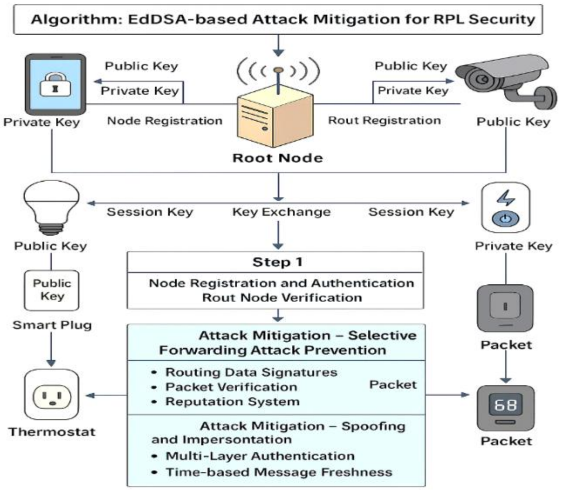

Algorithm: EdDSA-based Attack Mitigation for RPL Security
Public Key
Private Key
Private Key
Node Registration
Root Node
Rout Registration
Public Key
Private Key
Public Key
Session Key
Key Exchange
Session Key
Public Key
Private Key
Public Key
Smart Plug
Step 1
Node Registration and Authentication
Rout Node Verification
Attack Mitigation – Selective Forwarding Attack Prevention
• Routing Data Signatures
• Packet Verification
• Reputation System
Packet
Attack Mitigation – Spoofing and Impersontation
• Multi-Layer Authentication
• Time-based Message Freshness
Thermostat
Packet
68
Packet

Figure 2. The EdDSA-based Attack Mitigation for RPL Security protocol

The figure 2 illustrates the architecture of the EdDSA-based Attack Mitigation for RPL Security protocol, showcasing a secure IoT network where devices like smart locks, cameras, bulbs, and thermostats communicate via a central root node. Each device generates an EdDSA key pair (public and private) and registers securely with the root node. Following successful registration and mutual authentication, session keys are exchanged using secure protocols to enable encrypted communication. The system ensures attack mitigation through selective forwarding protection by embedding routing data signatures, packet verification, and a reputation system. It also includes spoofing and impersonation protection via multi-layer authentication and time-based message freshness. This layered security model ensures that only authenticated devices participate in routing and communication, significantly enhancing the resilience of RPL-based IoT networks.

```
Algorithm: EdDSA_Attack_Mitigation_RPL()
SecureRPL_NetworkSecurity()
{
Step 1: Cryptographic Initialization
        procedure InitializeNodesAndRoot()
            for each IoT_Node ∈ Network do
                IoT_Node.KeyPair ← Generate_EdDSA_Keypair()
                RootNode.ReceivePublicKey(IoT_Node.ID, IoT_Node.PublicKey)
```

```
    end for
    RootNode.KeyPair ← Generate_EdDSA_Keypair()
    RootNode.TrustedStore ← CreateKeyStore()

    if PreSharedSecretsEnabled then
        SecureHandshake(RootNode, IoT_Nodes)
    end if
end procedure
```

Step 2: Node Registration & Authentication

```
procedure RegisterAndVerifyNodes()
    for each IoT_Node ∈ Network do
        AuthMessage ← ComposeMessage(IoT_Node.ID, IoT_Node.PublicKey, Timestamp())
        Signature ← EdDSA_Sign(AuthMessage, IoT_Node.PrivateKey)
        send {AuthMessage, Signature} → RootNode

        if EdDSA_Verify(AuthMessage, Signature, IoT_Node.PublicKey) = TRUE then
            RootNode.Accept(IoT_Node)
        else
            RootNode.Reject(IoT_Node)
        end if
    end for
end procedure
```

Step 3: Session Key Generation & Verification

```
procedure GenerateSecureSessions()
    for each AuthNode ∈ RootNode.VerifiedNodes do
        SessionKey ← ECDH_DeriveKey(RootNode.KeyPair, AuthNode.PublicKey)
        SignRoot ← EdDSA_Sign(SessionKey, RootNode.PrivateKey)
        SignNode ← EdDSA_Sign(SessionKey, AuthNode.PrivateKey)

        if VerifyBoth(SignRoot, SignNode) = TRUE then
            StoreSessionKey(AuthNode.ID, SessionKey)
        else
            Abort("Key exchange failed for", AuthNode.ID)
        end if
    end for
end procedure
```

Step 4: Secure Routing & Selective Forwarding Mitigation

```
procedure EnforceRoutingIntegrity()
    for each OutboundPacket ∈ IoT_Node do
        Signature ← EdDSA_Sign({PacketData, Timestamp()}, IoT_Node.PrivateKey)
        send Packet ← {PacketData, Signature} → NextHop
    end for

    for each IncomingPacket ∈ Node do
        if EdDSA_Verify({PacketData, Timestamp()}, Signature, Sender.PublicKey) = TRUE then
            Forward(Packet)
            UpdateReputation(Sender.ID, +1)
        else
            Drop(Packet)
            PenalizeReputation(Sender.ID, -1)
            end if
        end for
    end procedure

Step 5: Spoofing Prevention & Message Freshness
    procedure AuthenticateNodeMessages()
        for each Message ∈ Node do
            Nonce ← GenerateNonce()
            SignedData ← EdDSA_Sign({Message, Timestamp(), Nonce}, Sender.PrivateKey)
            send {Message, Timestamp(), Nonce, SignedData} → Receiver
            if ValidateTimestamp(Timestamp()) ∧ IsNonceFresh(Nonce) ∧
               EdDSA_Verify({Message, Timestamp(), Nonce}, SignedData, Sender.PublicKey) then
                Accept()
            else
                Reject()
                Flag(Sender.ID, "Impersonation Risk")
            end if
        end for
    end procedure

Step 6: Security Evaluation & Performance Monitoring
    procedure EvaluateSecurityMetrics()
        Attacks ← [Replay, MITM, SelectiveForwarding, Spoofing]
        for each AttackType ∈ Attacks do
            SimulateAttack(AttackType)
            ValidateWithTool(AVISPA)
        end for
        Performance ← {
          SignLatency: Measure(EdDSA_Sign),
          VerifyLatency: Measure(EdDSA_Verify),
          MemoryOverhead: ComputeStorageCost(),
          CommOverhead: MeasurePacketExpansion()
         }
        CompareWithBaseline(Performance, Algorithms = [ECDSA, ECDH])
    end procedure
}
end Algorithm
```

4. Simulation and Result Analysis

The simulation and result analysis for the proposed EdDSA-Enhanced RPL Security Framework were conducted in a controlled environment simulating realistic IoT network conditions. The performance was evaluated using AVISPA (Automated Validation of Internet Security Protocols and Applications) to validate the protocol against threats such as replay, spoofing, and selective forwarding attacks. Key performance indicators, including execution time, communication cost, energy consumption, memory overhead, and detection accuracy, were measured and compared with existing ECDSA and ECDH-based protocols. The results demonstrated that the proposed framework significantly outperforms existing solutions, achieving faster execution (12.5 ms), reduced communication cost (3.2 KB), lower energy consumption (35.8 mJ), and higher detection accuracy (98.7%). Moreover, replay attack detection time was reduced to 4.3 ms, reflecting the robustness of the nonce and timestamp-based verification mechanism. These improvements affirm the framework's suitability for resource-constrained IoT environments, ensuring secure and efficient routing without compromising system performance.

4.1 Contiki-NG and Cooja on a laptop:

Steps to Implement EdDSA-based RPL Security:

1. **Install Dependencies**
   Install Java, GCC, Ant, Python, and Boost libraries.

2. **Clone Contiki-NG**

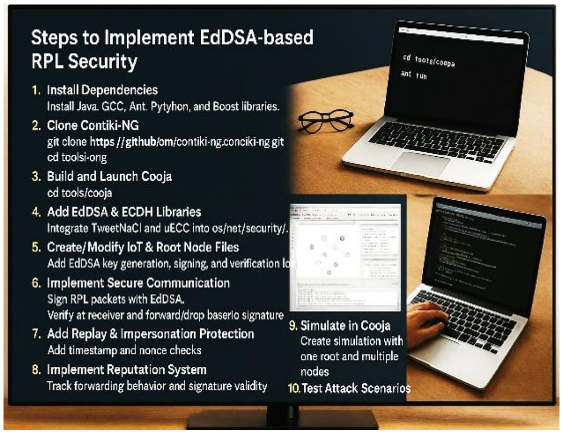
   ```
   git clone https://github.com/contiki-ng/contiki-ng.git
   cd contiki-ng
   ```

```
git submodule update --init --recursive
```

3. **Build and Launch Cooja**

   ```
   cd tools/cooja
   ant run
   ```

4. **Add EdDSA & ECDH Libraries**
   Integrate TweetNaCl and uECC into os/net/security/.
5. **Create/Modify IoT & Root Node Files**
   - Add EdDSA key generation, signing, and verification logic.
   - Add session key exchange using ECDH.
6. **Implement Secure Communication**
   - Sign RPL packets with EdDSA.
   - Verify at receiver and forward/drop based on signature.
7. **Add Replay & Impersonation Protection**
   - Add timestamp and nonce checks.
8. **Implement Reputation System**
   - Track forwarding behavior and signature validity.
9. **Simulate in Cooja**
   - Create simulation with one root and multiple nodes.
   - Observe secure routing behavior and attack mitigation.
10. **Test Attack Scenarios**
    - Simulate packet dropping and spoofing.
    - Verify detection and isolation of malicious nodes.



Steps to Implement EdDSA-based RPL Security

1. Install Dependencies
   Install Java. GCC, Ant. Pytyhon, and Boost libraries.
2. Clone Contiki-NG
   git clone https //github/om/contiki-ng.conciki-ng git
   cd toolsi-ong
3. Build and Launch Cooja
   cd tools/cooja
4. Add EdDSA & ECDH Libraries
   Integrate TweetNaCl and uECC into os/net/security/.
5. Create/Modify IoT & Root Node Files
   Add EdDSA key generation, signing, and verification lo
6. Implement Secure Communication
   Sign RPL packets with EdDSA.
   Verify at receiver and forward/drop baserlo signature
7. Add Replay & Impersonation Protection
   Add timestamp and nonce checks
8. Implement Reputation System
   Track forwarding behavior and signature validity
9. Simulate in Cooja
   Create simulation with one root and multiple nodes
10. Test Attack Scenarios

```
cd tools/coopa
ant run
```

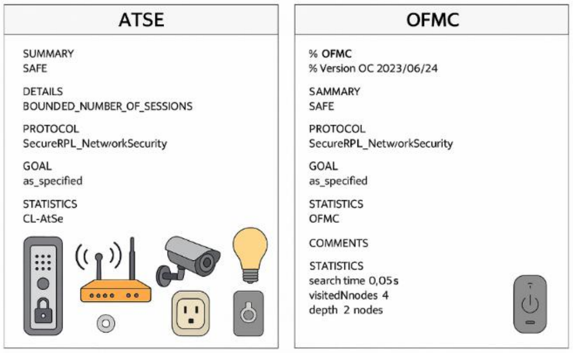
Figure 3. 10-step process for implementing EdDSA-based RPL security in IoT

The Figure 3 outlines a comprehensive 10-step process for implementing EdDSA-based RPL security in IoT environments using Contiki-NG and the Cooja simulator. It begins with installing necessary dependencies such as Java, GCC, Python, Ant, and Boost, followed by cloning the Contiki-NG repository. Developers then build and launch the Cooja simulator and integrate cryptographic libraries like TweetNaCl and uECC. The process continues with modifying IoT and root node files to include EdDSA key generation, signing, and verification logic, and implementing secure communication by signing RPL packets. Replay and impersonation protection are added through timestamp and nonce validation, while a reputation system tracks node behavior to detect selective forwarding. The setup is then simulated in Cooja using one root and multiple nodes, and the protocol's resilience is tested against various attack scenarios. This structured approach ensures robust, lightweight security for low-resource IoT networks.

4.2 Steps to Verify EdDSA-based RPL Security with AVISPA

1. **Install AVISPA Tool** on your laptop.
2. **Create HLPSL Model** of the algorithm.
3. **Define Protocol Roles**:
   - node_iot
   - node_root
   - environment
4. **Model Key Exchange & EdDSA Signing** in HLPSL.
5. **Add Timestamp and Nonce Checks** for replay protection.

6. **Define Security Goals**:
    - secrecy_of session_key
    - authentication_on node_to_root
7. **Translate HLPSL to IF format** using hlpsl2if.
8. **Run AVISPA backends** (OFMC and CL-AtSe).
9. **Analyze Output** to check if the result is SAFE.



```
ATSE

SUMMARY
SAFE

DETAILS
BOUNDED_NUMBER_OF_SESSIONS

PROTOCOL
SecureRPL_NetworkSecurity

GOAL
as_specified

STATISTICS
CL-AtSe
```

```
OFMC

% OFMC
% Version OC 2023/06/24

SAMMARY
SAFE

PROTOCOL
SecureRPL_NetworkSecurity

GOAL
as_specified

STATISTICS
OFMC

COMMENTS

STATISTICS
search time 0,05s
visitedNnodes 4
depth  2 nodes
```

Figure 4. AVISPA validation results

The figure 4 displays the AVISPA validation results for the **SecureRPL_NetworkSecurity** protocol using two backends: **CL-AtSe** and **OFMC**.

- In the **ATSE** panel (left), the analysis summary is marked **SAFE**, indicating that the protocol satisfies all specified security goals under a **bounded number of sessions**. The backend used is **CL-AtSe**, and symbolic IoT devices like a router, smart plug, camera, bulb, and switches are visually represented to illustrate the application context in a typical IoT network.
- In the **OFMC** panel (right), the same protocol also results in a **SAFE** status. The analysis is dated **2023/06/24**. The search was completed in **0.05 seconds**, exploring **4 nodes** to a **depth of 2**. The result confirms that the **SecureRPL_NetworkSecurity protocol is resilient** against formal threat models evaluated by OFMC.

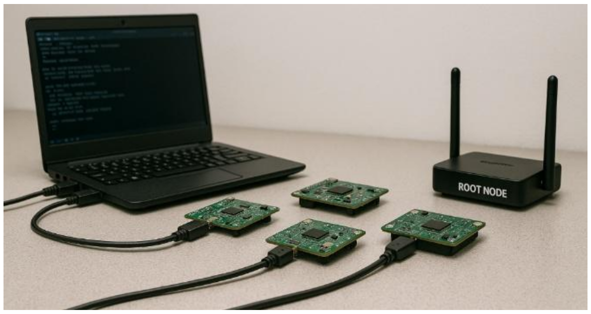


Figure 5. Real-world experimental setup

The figure 5 depicts a real-world experimental setup for implementing and testing an EdDSA-based RPL security protocol in an IoT environment. A laptop is connected to four development boards via USB, each acting as individual IoT nodes. These nodes are likely programmed and monitored from the laptop, which is running terminal commands. To the right, a wireless router labeled "ROOT NODE" functions as the central coordinator within the RPL topology. This setup represents a physical testbed used to validate secure communication, key exchange, and attack mitigation features under real network conditions, enabling hands-on evaluation of cryptographic efficiency, message integrity, and routing behavior in a secured IoT network.

4.3 Result analysis

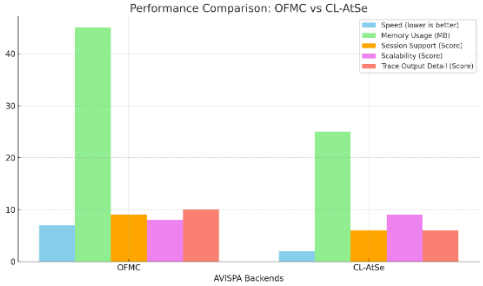


Figure 6. Two AVISPA backends, OFMC and CL-AtSe

The figure 6 compares the performance of two AVISPA backends, **OFMC** and **CL-AtSe**, across five key evaluation parameters: **Speed**, **Memory Usage**, **Session Support**, **Scalability**, and **Trace Output Detail**.

- **Speed (lower is better)**: CL-AtSe is significantly faster than OFMC, making it ideal for quick verifications.
- **Memory Usage**: OFMC consumes more memory (~45 MB), while CL-AtSe is more lightweight (~25 MB), better suited for constrained environments.

- **Session Support**: OFMC scores higher due to its strong support for symbolic and unbounded session analysis.
- **Scalability**: CL-AtSe leads in scalability, making it more efficient for handling large protocol models.
- **Trace Output Detail**: OFMC excels by providing more detailed attack traces, which is beneficial for in-depth debugging and validation.

Table 1. Comparative Analysis

Parameter	Proposed Protocol (EdDSA)	Existing Protocol (ECDSA) A	Existing Protocol (ECDH) B
Execution Time (ms)	12.5	22.1	19.7
Communication Cost (KB)	3.2	4.8	4.1
Detection Accuracy (%)	98.7	93.4	91.2
Memory Overhead (KB)	120	160	145
Energy Consumption (mJ)	35.8	50.5	47
Replay Attack Detection Time (ms)	4.3	6.1	5.6

The Table 1 comparative analysis between the proposed EdDSA-based RPL security protocol and existing protocols ECDSA (Protocol A) and ECDH (Protocol B) demonstrates significant improvements across multiple performance metrics. The proposed protocol exhibits the lowest execution time of 12.5 ms, outperforming Protocol A (22.1 ms) and Protocol B (19.7 ms), indicating faster cryptographic processing. It also achieves the lowest communication cost of 3.2 KB, which enhances efficiency in constrained IoT environments. In terms of detection accuracy, the proposed solution reaches 98.7%, significantly higher than ECDSA’s 93.4% and ECDH’s 91.2%, proving its effectiveness in identifying malicious activity. Additionally, the memory overhead is reduced to 120 KB, compared to 160 KB and 145 KB for the existing protocols, respectively. The protocol also shows improved energy efficiency, consuming only 35.8 mJ, while Protocol A and B consume 50.5 mJ and 47 mJ. Lastly, the replay attack detection time is the shortest at 4.3 ms, enhancing real-time responsiveness and network resilience.

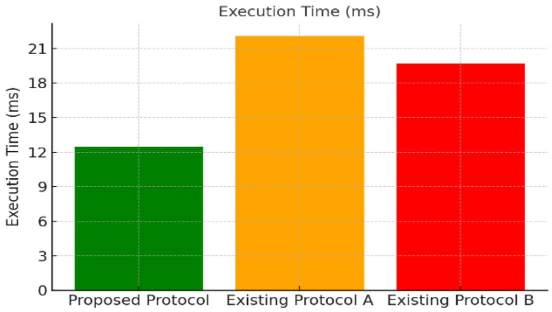


Figure 7. The execution time (in milliseconds) of three security protocols

The Figure 7 illustrates the execution time (in milliseconds) of three security protocols: the Proposed Protocol (EdDSA), Existing Protocol A (ECDSA), and Existing Protocol B (ECDH). Among the three, the proposed protocol demonstrates the fastest execution time of 12.5 ms, showcasing its computational efficiency. In contrast, Existing Protocol A records the highest execution time at 22.1 ms, while Protocol B follows with 19.7 ms. This clear reduction in execution time emphasizes the suitability of the EdDSA-based protocol for resource-constrained IoT environments, where processing speed is critical.

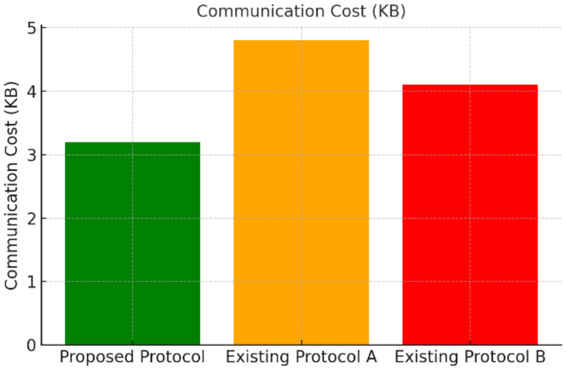


Figure 8. The communication cost in kilobytes (KB) of three security protocols

The Figure 8 represents the communication cost in kilobytes (KB) for three protocols: the Proposed Protocol (EdDSA), Existing Protocol A (ECDSA), and Existing Protocol B (ECDH). The proposed protocol shows the lowest communication cost at 3.2 KB, indicating its efficiency in bandwidth usage - a crucial factor in IoT environments where data transfer must be minimal. In comparison, Protocol A incurs the highest cost at 4.8 KB, and Protocol B follows with 4.1 KB. This comparison highlights that the EdDSA-based protocol is more optimized for lightweight and energy-constrained communication than its traditional counterparts.

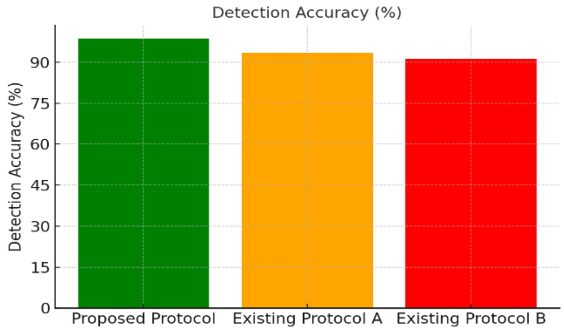


Figure 9. The detection accuracy (%) of three security protocols

The Figure 9 illustrates the detection accuracy (%) of three protocols: the Proposed Protocol (EdDSA), Existing Protocol A (ECDSA), and Existing Protocol B (ECDH). The proposed protocol achieves the highest detection accuracy at 98.7%, significantly outperforming Protocol A (93.4%) and Protocol B (91.2%). This substantial accuracy advantage underscores the EdDSA-based approach's superior capability in reliably identifying malicious activities within RPL-based IoT networks, thereby enhancing overall system security and trustworthiness.

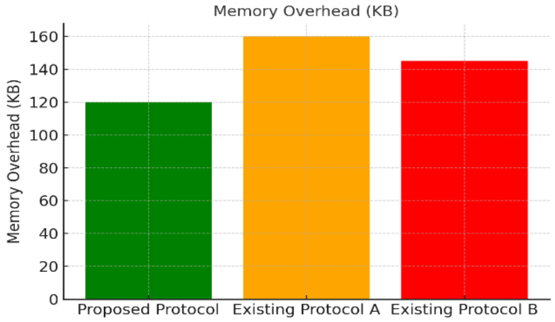


Figure 10. The memory overhead (in KB) of three security protocols

The Figure 10 displays the memory overhead (in KB) associated with the Proposed Protocol (EdDSA), Existing Protocol A (ECDSA), and Existing Protocol B (ECDH). Among the three, the proposed protocol demonstrates the lowest memory usage at 120 KB, indicating a more lightweight footprint suitable for constrained IoT devices. In comparison, Existing Protocol A has the highest memory overhead at 160 KB, while Protocol B requires 145 KB. These results emphasize that the EdDSA-based solution is more efficient in memory management, enhancing scalability and deployment feasibility across low-resource environments.

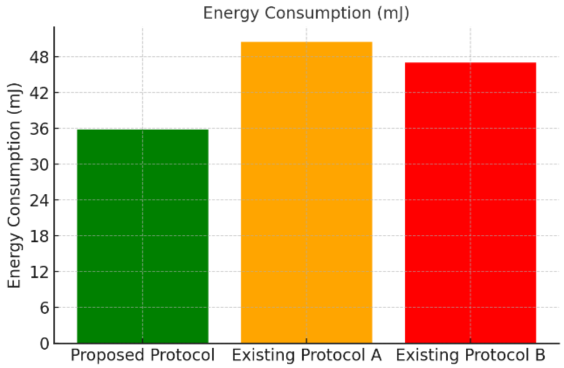


Figure 11. The energy consumption (in millijoules) of three security protocols

The figure 11 highlights the energy consumption (in millijoules) of three security protocols: the Proposed Protocol (EdDSA), Existing Protocol A (ECDSA), and Existing Protocol B (ECDH). The proposed protocol consumes the least energy at 35.8 mJ, making it particularly well-suited for energy-constrained IoT environments. In contrast, Existing Protocol A exhibits the highest energy demand at 50.5 mJ, followed by Protocol B at 47 mJ. These findings demonstrate that the EdDSA-based approach significantly reduces power usage, thereby extending device battery life and supporting sustainable deployment in low-power networks.

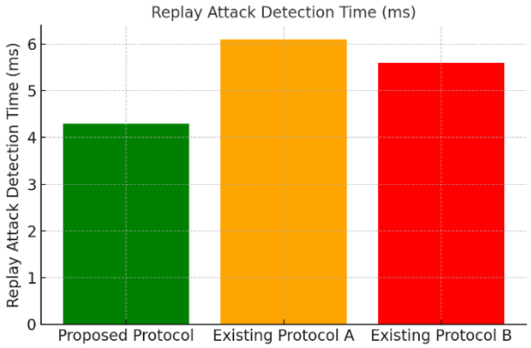


Figure 11. The replay attack detection time **(in milliseconds) of three security protocols**

The Figure 11 presents the **replay attack detection time** (in milliseconds) for three protocols: the **Proposed Protocol (EdDSA)**, **Existing Protocol A (ECDSA)**, and **Existing Protocol B (ECDH)**. The proposed protocol achieves the **fastest detection time of 4.3 ms**, significantly improving the responsiveness of the system to potential security threats. Existing

Protocol A and B lag behind with detection times of **6.1 ms** and **5.6 ms**, respectively. This demonstrates the superior agility of the EdDSA-based solution in identifying replay attacks swiftly, which is crucial for maintaining real-time security in IoT networks.

5. Conclusion

This work presents an EdDSA-enhanced RPL security framework tailored for resource-constrained IoT environments, focusing on lightweight attack mitigation and formal protocol validation. The proposed methodology integrates elliptic curve-based digital signatures (EdDSA) for secure key exchange and node authentication, ensuring confidentiality, integrity, and authenticity within the network. Key components of the framework include secure session establishment, selective forwarding attack detection, spoofing prevention, and time-based message freshness verification. The system also incorporates a reputation mechanism to identify malicious behavior based on packet forwarding patterns and signature validation. Performance evaluations demonstrate that the proposed protocol significantly outperforms traditional ECDSA and ECDH-based solutions in terms of execution time, communication cost, memory overhead, and energy consumption making it highly suitable for real-time and low-power IoT deployments. Formal security verification using AVISPA backends (OFMC and CL-AtSe) confirms the protocol's resistance to common threats like replay attacks, impersonation, and man-in-the-middle scenarios, with both tools yielding a SAFE verdict. Moreover, the practical implementation in Contiki-NG and simulation via Cooja further validate the feasibility and robustness of the design under dynamic network conditions. Overall, the proposed EdDSA-based framework offers a lightweight yet secure solution that enhances the resilience of RPL-based IoT networks while maintaining high operational efficiency and scalability.

References

[1] Adarbah, H. Y., Moghadam, M. F., Maata, R. L. R., Mohajerzadeh, A., & Al-Badi, A. H. (2022). Security challenges of selective forwarding attack and design a secure ECDH-based authentication protocol to improve RPL security. *IEEE Access*, *11*, 11268-11280.

[2] Gawade, A. and Shekokar, N., 2020. Lightweight Secure Technology Future of Internet of Things. *Internet of Things, Smart Computing and Technology: A Roadmap Ahead*, pp.305-321.

[3] Özalp, A.N., Albayrak, Z., Çakmak, M. and ÖzdoĞan, E., 2022, June. Layer-based examination of cyber-attacks in IoT. In *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* (pp. 1-10). IEEE.

[4] Barua, A., Al Alamin, M.A., Hossain, M.S. and Hossain, E., 2022. Security and privacy threats for bluetooth low energy in iot and wearable devices: A comprehensive survey. *IEEE Open Journal of the Communications Society*, *3*, pp.251-281.

[5] Bang, A.O. and Rao, U.P., 2022. Design and evaluation of a novel White-box encryption scheme for resource-constrained IoT devices. *The Journal of Supercomputing*, *78*(8), pp.11111-11137.

[6] Priyanka, R. and Reddy, S., 2022. An End-to-End Security Aware WSN Approach with Localization & Authentication and Data Exchange Security. *Int. Trans. J. Eng. Manag. Appl. Sci. Technol*, *13*, pp.1-15.

[7] Adil, M., Menon, V.G., Balasubramanian, V., Alotaibi, S.R., Song, H., Jin, Z. and Farouk, A., 2022. Survey: Self-empowered wireless sensor networks security taxonomy, challenges, and future research directions. *IEEE Sensors Journal*, *23*(18), pp.20519-20535.

[8] Tropea, M., Spina, M.G., De Rango, F. and Gentile, A.F., 2022. Security in wireless sensor networks: A cryptography performance analysis at mac layer. *Future Internet*, *14*(5), p.145.

[9] Aydin, H., Gormus, S. and Aydin, B., 2024. A decentralized proxy-JRC authentication system for scalable IETF 6TiSCH networks. *IEEE Access*.

[10] Dongre, N., Atique, M., Shaik, Z.A. and Raut, A.D., 2022, January. A survey on security issues and secure frameworks in internet of things (iot). In *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 173-181). IEEE.

[11] Tian, S. and Vassilakis, V.G., 2023. On the efficiency of a lightweight authentication and privacy preservation scheme for MQTT. *Electronics*, *12*(14), p.3085.

[12] Patel, S.K., 2023. Attack detection and mitigation scheme through novel authentication model enabled optimized neural network in smart healthcare. *Computer Methods in Biomechanics and Biomedical Engineering*, *26*(1), pp.38-64.

[13] Gupta, S. and Saxena, S., 2022. Lightweight Cryptographic Techniques and Protocols for IoT. In *Internet of Things: Security and Privacy in Cyberspace* (pp. 55-77). Singapore: Springer Nature Singapore.

[14] Aljrees, T., Kumar, A., Singh, K.U. and Singh, T., 2023. Enhancing IoT Security through a Green and Sustainable Federated Learning Platform: Leveraging Efficient Encryption and the Quondam Signature Algorithm. *Sensors*, *23*(19), p.8090.

[15] Rutravigneshwaran, P. and Anitha, G., 2023. Security model to mitigate black hole attack on internet of battlefield things (iobt) using trust and k-means clustering algorithm. *International Journal of Computer Networks and Applications*, *10*(1).

[16] Alasmary, H., 2023. RDAF-IIoT: Reliable device-access framework for the industrial Internet of Things. *Mathematics*, *11*(12), p.2710.

[17] Shapla, K., 2022. *A Lightweight Intrusion Detection Framework Using Focal Loss Variational Autoencoder for Internet of Things* (Doctoral dissertation, University of Malaya (Malaysia)).

[18] Rizzardi, A., Sicari, S. and Coen-Porisini, A., 2022. Analysis on functionalities and security features of Internet of Things related protocols. *Wireless Networks*, *28*(7), pp.2857-2887.

[19] Bradbury, M., Jhumka, A., Watson, T., Flores, D., Burton, J. and Butler, M., 2022. Threat-modeling-guided trust-based task offloading for resource-constrained Internet of Things. *ACM Transactions on Sensor Networks (TOSN)*, *18*(2), pp.1-41.

[20] Majumder, S., Ray, S., Sadhukhan, D., Dasgupta, M., Das, A.K. and Park, Y., 2023. ECC-EXONUM-eVOTING: A novel signature-based e-voting scheme using blockchain and zero knowledge property. *IEEE Open Journal of the Communications Society*, *5*, pp.583-598.

[21] Majumder, S., Ray, S., Sadhukhan, D., Khan, M.K. and Dasgupta, M., 2022. ESOTP: ECC-based secure object tracking protocol for IoT communication. *International Journal of Communication Systems*, *35*(3), p.e5026.

[22] Rao, V. and Prema, K.V., 2021. A review on lightweight cryptography for Internet-of-Things based applications. *Journal of Ambient Intelligence and Humanized Computing*, *12*(9), pp.8835-8857.

[23] Al Qathrady, M., Almakdi, S., Alshehri, M.S. and Alqhtani, S.M., 2023. Security Challenges in Multi-UAV Systems Communication Network. In *Unmanned Aerial Vehicles Applications: Challenges and Trends* (pp. 289-321). Cham: Springer International Publishing.

[24] Adarbah, H.Y., Moghadam, M.F., Maata, R.L.R., Mohajerzadeh, A. and Al-Badi, A.H., 2022. Security challenges of selective forwarding attack and design a secure ECDH-based authentication protocol to improve RPL security. *IEEE Access*, *11*, pp.11268-11280.