Scientific Research Publishing

Open Journal of Applied Sciences, 2026, 16(11), 4371-4393
https://www.scirp.org/journal/ojapps
ISSN Online: 2165-3925
ISSN Print: 2165-3917

# Fortifying Sensor Networks Using IoT Protocols for Wireless Monitoring and Security

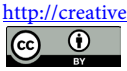
**Godfrey Wandwi[1], Gabriel Dams[2]**

[1]School of Digital Technologies and Transformation Studies, Dar es Salaam Tumaini University, Dar es Salaam, Tanzania
[2]College of Science, Computing and Engineering, Kaduna State University, Kaduna, Nigeria
Email: godfrey.wandwi@dartu.ac.tz

**How to cite this paper:** Wandwi, G. and Dams, G. (2026) Fortifying Sensor Networks Using IoT Protocols for Wireless Monitoring and Security. *Open Journal of Applied Sciences*, **16**, 4371-4393.
https://doi.org/10.4236/ojapps.2026.1611238

**Received:** July 16, 2026
**Accepted:** November 8, 2026
**Published:** November 11, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/



Open Access

## Abstract

The expanding footprint of the Internet of Things (IoT) continues to redefine wireless communication, particularly in sensor-based monitoring systems where reliability, coverage, and security are paramount. This paper explores how wireless sensor networks (WSNs), integral to modern IoT architectures, remain vulnerable due to the adoption of lightweight and often under-secured communication protocols. With an emphasis on cost-efficiency and energy preservation, many sensor nodes are deployed with minimal computational resources, exposing them to a broad range of attacks. The study critically analyzes the structural composition of widely adopted IoT protocols such as Zigbee, CoAP, and 6LoWPAN used in multi-hop and mesh configurations for WSN deployments. Special attention is paid to protocol vulnerabilities at the MAC, network, and transport layers, as well as the trade-offs between security robustness and low-power operation. Through a layered approach, the paper highlights key mechanisms for strengthening protocol security ranging from adaptive encryption schemes to protocol-specific intrusion prevention strategies thereby enhancing resilience in distributed sensing environments without compromising performance or scalability.

## Keywords

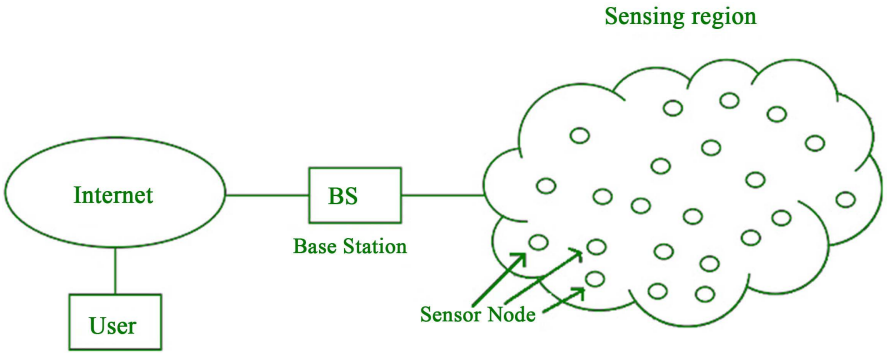
Wireless Sensor Networks (WSNs), IoT Protocols, Zigbee, CoAP, 6LoWPAN, Secure Communication, Distributed Monitoring, Protocol Vulnerabilities

## 1. Introduction

The Internet of Things (IoT) refers to a diverse array of interconnected embedded devices that operate with unique identifiers and communicate over existing internet infrastructures to deliver intelligent services beyond traditional machine-to-

machine (M2M) interactions. These IoT-enabled systems encompass a wide range of application domains from home automation and smart cities to healthcare and industrial control each relying on seamless communication, intelligent data processing, and responsive actuation. In the context of wireless sensor networks (WSNs), which form a crucial subset of IoT architecture, the integration of lightweight communication protocols is fundamental to ensure reliable monitoring and timely data exchange in distributed environments.

Sensor networks typically consist of numerous spatially distributed nodes equipped with sensors, transceivers, and microcontrollers. These nodes are tasked with collecting real-time environmental data such as temperature, vibration, humidity, or motion and relaying it through wireless protocols to a centralized gateway or cloud-based storage. As illustrated in Figure 1, the operation of a sensor network is structured around a closed feedback loop comprising data acquisition, wireless transmission via protocols like Zigbee or 6LoWPAN, intermediate processing, and decision-making based on threshold logic or pattern recognition algorithms.




(Depicting node-level sensing, wireless communication using IoT protocols, and cloud interfacing with actuation).

**Figure 1.** Functional model of a wireless sensor network using IoT protocols.

The foundational components of this IoT ecosystem, especially for WSNs, include:

(a) Sensor nodes with uniquely addressable identifiers,

(b) Data aggregation systems or local gateways, and

(c) Remote/cloud computing systems capable of advanced analytics and control logic.

This triad is essential for building intelligent wireless monitoring infrastructure, whether deployed in agriculture, industrial plants, environmental surveillance, or public safety.

According to global IoT deployment forecasts provided by DBS Bank in collaboration with Gartner (see Table 1), the number of consumer and industrial devices connected to wireless networks is expected to surpass 75 billion by 2025 [1]. This surge, while indicative of technological progress, also raises serious security concerns. The drive for faster deployment and cost reduction has led to the mass

production of IoT devices with minimal hardware security, lightweight cryptographic support, and default credentials rendering them susceptible to unauthorized access and cyberattacks. Notable incidents have already demonstrated the weaknesses of these systems. For instance, medical implants such as pacemakers were reported vulnerable to remote hacking [2]; a massive distributed denial-of-service (DDoS) attack in 2016 exploited unsecured IoT cameras and recorders to disrupt major internet services [3]; and a baby monitor with wireless capabilities was hijacked to stream sensitive footage [4].

**Table 1.** Projected global IoT device deployment statistics.

| Source | Year | Projected IoT Devices (Millions) | Consumer Share (%) |
|---|---|---|---|
| Gartner | 2018 | 75,000 | 60% |
| UN ITU | 2017 | 50,100 | 58% |
| World Bank | 2016 | 38,400 | 54% |

Despite advancements in wireless protocols such as CoAP, Zigbee, and LoRaWAN, many lack robust security features at the protocol level, especially in resource-constrained sensor environments. As WSNs often operate unattended in untrusted zones, their security becomes critical not just for data integrity, but also for ensuring public safety and preserving user privacy. In this context, the present paper examines protocol-level vulnerabilities inherent in sensor networks and proposes enhancements tailored for constrained environments. The goal is to fortify wireless monitoring infrastructures while maintaining low energy consumption and communication overhead.

The remainder of this paper is organized as follows. Section 2 examines security issues across the IoT protocol stack, while Section 3 analyzes the security vulnerabilities and limitations of widely used IoT protocols. Section 4 presents lightweight cryptographic and protocol-level security enhancements for strengthening wireless sensor networks. Section 5 describes the proposed key-management framework and outlines the performance evaluation methodology for comparing static-key and PUF/TRNG-assisted dynamic key-management approaches. Section 6 discusses key-generation techniques suitable for resource-constrained sensor networks, including pseudo-random number generators, true random number generators, physically unclonable functions, and hybrid approaches. Section 7 discusses the implications, performance trade-offs, and limitations of the proposed framework. Finally, Section 8 presents the conclusions and recommendations for future research.

To clarify the research objectives, this study addresses the following research questions:

**RQ1:** To what extent can a hybrid key-generation approach combining Physically Unclonable Functions (PUFs) and True Random Number Generators

(TRNGs) improve key-management resilience and communication security in resource-constrained wireless sensor networks compared with a static-key approach?

**RQ2:** What are the effects of integrating PUF/TRNG-assisted key generation and periodic key refresh into an IoT-enabled wireless sensor network in terms of energy consumption, end-to-end communication latency, communication overhead, and key-establishment delay?

The study further evaluates the hypothesis that a properly implemented PUF/TRNG-assisted key-management scheme can reduce dependence on long-lived static keys and improve key freshness, while keeping the additional energy consumption and latency within application-specific operational limits. This hypothesis will be assessed through a reproducible simulation or test-bed comparison rather than inferred solely from the qualitative properties of the cryptographic mechanisms.

## 2. Security Issues in IoT Protocol Stack

Sensor networks are increasingly becoming the backbone of real-time monitoring and surveillance systems, especially with the proliferation of Internet of Things (IoT) technologies. These networks consist of spatially distributed sensors that collect, process, and transmit data wirelessly, relying heavily on a layered IoT protocol stack. However, while their deployment enables improved automation and data-driven decision-making in areas such as healthcare, military surveillance, industrial automation, and smart cities, their wireless and resource-constrained nature introduces substantial security challenges [5]. The security threats in sensor networks stem from both the characteristics of IoT environments and the architecture of the communication protocols in use.

A major reason sensor networks remain susceptible to attacks is the trade-off between lightweight design and security robustness. Since most sensor nodes are battery-powered and designed for low-cost deployment, their computational and storage capacities are limited. Consequently, vendors often compromise on integrated security mechanisms during the development phase, resulting in inadequately protected nodes vulnerable to intrusion, interception, or manipulation [6]. Moreover, these devices are frequently deployed in hostile or unattended environments, making them targets for physical tampering, side-channel attacks, or rogue node deployment.

**(a) Resource Constraints**

Many sensor nodes in IoT environments lack the necessary hardware and memory to execute strong cryptographic operations or run real-time operating systems with robust security policies. This results in minimal or no encryption, poor authentication processes, and weak access control mechanisms. Attackers exploit this to intercept transmissions, reprogram nodes, or inject malicious code into the network [7]. **Table 2** outlines the typical constraints in sensor nodes that hinder the implementation of conventional security protocols.

Table 2. Resource limitations in typical IoT sensor nodes.

| Constraint | Description |
|---|---|
| Memory | <256 KB RAM, insufficient for TLS stack |
| Processing Power | <20 MHz CPUs |
| Power Supply | Battery-driven, energy-sensitive |
| Communication Range | Short (typically < 100 m) |

### (b) Vulnerable Protocol Layers

Figure 2 below presents the simplified IoT protocol stack for sensor networks and highlights common vulnerabilities at each layer.

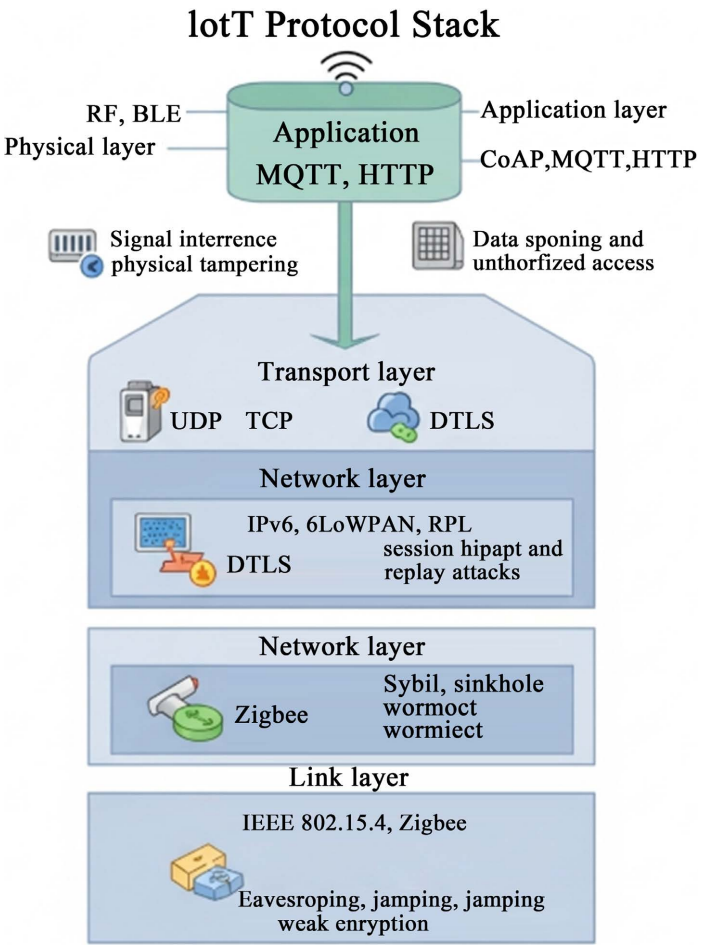



Figure 2. IoT protocol stack and associated security threats [8].

### (c) Specific Protocol Challenges

CoAP and DTLS: The Constrained Application Protocol (CoAP) is designed for resource-limited devices but lacks inherent security. It relies on Datagram Transport Layer Security (DTLS) for secure data transmission. However, the handshake overhead in DTLS poses a challenge for energy-constrained sensors.

Additionally, packet reordering or loss due to unreliable UDP links can result in failed sessions or exposure to man-in-the-middle (MITM) attacks [9].

6LoWPAN: The 6LoWPAN adaptation layer compresses IPv6 headers for efficient transmission over IEEE 802.15.4 links. However, its reliance on the underlying MAC layer for security exposes it to threats such as fragmentation attacks and unauthorized IP spoofing. Attackers may exploit header compression logic to evade intrusion detection systems [10].

IEEE 802.15.4 and Zigbee: These are widely adopted in low-rate wireless personal area networks (LR-WPANs). IEEE 802.15.4 supports AES-CCM encryption at the MAC layer, but lacks efficient key management, making it vulnerable to key reuse and node impersonation. Zigbee, which builds on 802.15.4, inherits these weaknesses and suffers from poor key distribution and outdated cryptographic suites [11].

**(d) Lack of Interoperability and Standardization**

Another major challenge in fortifying sensor networks using IoT protocols lies in the lack of universal security standards across vendors and platforms. This fragmentation leads to heterogeneous systems that are difficult to secure uniformly. For instance, proprietary implementations of MQTT brokers may not implement TLS properly, while different versions of Zigbee may vary in key handling strategies, resulting in inconsistent security postures across devices [12].

**(e) Data Integrity and Confidentiality Risks**

Sensor data is often aggregated and relayed through multiple hops before reaching a central gateway or cloud platform. Without adequate encryption or digital signatures, this data is exposed to modification or injection attacks. A malicious node in the communication path can alter readings (e.g., temperature, pressure, motion) leading to false alarms, incorrect actuation, or safety hazards in mission-critical deployments [13].

**(f) Software Obsolescence and Update Challenges**

Sensor nodes, once deployed, rarely receive software updates due to cost and logistical issues. As vulnerabilities emerge, unpatched firmware becomes a liability. Additionally, many devices do not support secure boot or trusted firmware verification, making firmware replacement attacks a growing concern [14].

**(g) Insider Threats and Misconfiguration**

Security breaches in IoT sensor networks are not always due to external actors. Weak password policies, exposed administrative interfaces, or improperly configured access control lists (ACLs) can give unauthorized personnel access to sensitive systems. Once inside, attackers may manipulate data flows or reconfigure routing paths to exfiltrate data silently [15].

To address these issues comprehensively, researchers advocate for layered security models tailored to the constraints of IoT sensor networks. Strategies include the use of lightweight cryptography (e.g., ECC, SPECK), trust management schemes, anomaly detection through AI models, and the adoption of secure boot mechanisms. However, no single approach can mitigate all risks, hence a holistic and context-aware security framework remains essential.

## 3. Shortcomings of IoT Protocols

While the deployment of IoT protocols has brought efficiency and modularity to wireless sensor networks (WSNs), their limitations in providing comprehensive security measures are increasingly evident in real-world deployments. Despite the theoretical robustness of many protocol layers from IEEE 802.15.4 at the physical and MAC layers to 6LoWPAN, RPL, and CoAP in the network and application tiers, the practical integration into constrained sensor environments reveals a series of critical shortcomings. These deficiencies not only expose sensor networks to attacks but also undermine long-term reliability and trust in remote monitoring systems [16].

A detailed comparative analysis of IoT protocols reveals fundamental gaps across four key security metrics: data confidentiality, integrity, origin authentication, and freshness. This is particularly problematic in sensor networks where unattended deployment, low-power devices, and intermittent connectivity are the norm. The overview provided in Table 3 delineates how various commonly used IoT protocols fall short in ensuring these foundational security elements.

Table 3. Comparative analysis of IoT protocols across key security features.

| Protocol | Data Confidentiality | Data Integrity | Origin Authentication | Freshness |
|---|---|---|---|---|
| NFC | ✗ | ✗ | ✗ | ✗ |
| MQTT | ✓ (via TLS) | ✓ | ✓ | ✗ |
| CoAP + DTLS | ✓ | ✓ | ✓ | ✓ |
| IEEE 802.15.4 | ✓ (AES-128) | ✓ | ✗ | ✗ |
| 6LoWPAN | ✗ | ✗ | ✗ | ✗ |
| RPL | ✗ | ✓ | ✗ | ✗ |

Source: Adapted from [16].

As evidenced above, protocols such as NFC and 6LoWPAN fail to offer any intrinsic security guarantees, relying entirely on upper-layer implementations or external wrappers. This is a critical oversight, particularly for WSNs operating in open or hostile environments, where packet sniffing, injection, and impersonation attacks are trivial in the absence of cryptographic controls [17].

In particular, 6LoWPAN which is widely used in WSNs due to its efficient header compression and IPv6 compatibility, it does not incorporate any native cryptographic mechanism. As such, it lacks the capability to validate the authenticity or freshness of packets, making it inherently susceptible to spoofing, replay, and message injection attacks [18]. Similarly, although IEEE 802.15.4 provides MAC-layer encryption via AES-128, the protocol does not enforce key distribution policies or dynamic key rotation. The absence of formalized key management

renders it vulnerable to replay and denial-of-service (DoS) attacks when static keys are compromised [19].

Figure 3 illustrates a cross-layer security gap model in IoT communication stacks, identifying how shortcomings at individual layers result in holistic vulnerabilities in sensor networks.

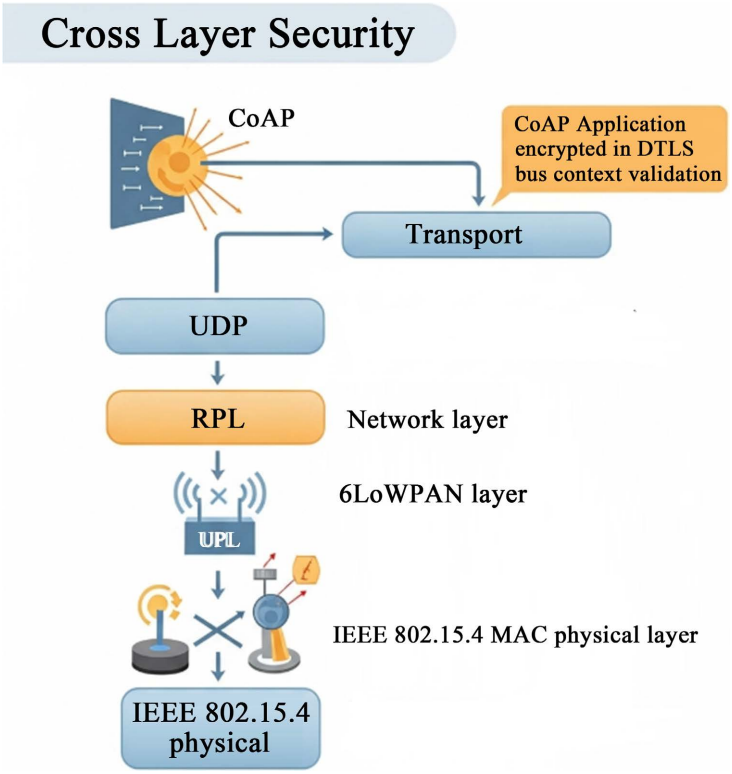



Figure 3. Cross-layer security gaps in IoT protocol stack for sensor networks.

Furthermore, protocols like RPL which is responsible for routing in low-power and lossy networks, it has been demonstrated to be highly vulnerable to topological attacks such as rank inconsistencies, sinkholes, and Sybil attacks. RPL's design philosophy prioritizes routing efficiency over authentication, and it does not require nodes to verify the legitimacy of routing updates, making it trivial for adversaries to disrupt routing tables or impersonate parent nodes [20].

Table 4 outlines documented vulnerabilities within key IoT protocols that affect WSNs used in critical monitoring infrastructures.

These shortcomings reveal that while IoT protocols provide a baseline for interoperability and low-power communication, they are often not engineered with a security-first mindset. Fortifying sensor networks, therefore, demands the integration of external security mechanisms such as context-aware intrusion detection systems, secure boot processes, and lightweight end-to-end encryption schemes that function despite hardware limitations. Additionally, secure and scalable key management remains a non-negotiable requirement to close the current gaps exploited by adversaries.

Table 4. Documented vulnerabilities in IoT protocols for WSNs.

| Protocol | Known Vulnerabilities | Impact on WSN Security |
|---|---|---|
| CoAP | Replay attacks, spoofing | Data tampering, false commands |
| 6LoWPAN | Fragmentation abuse, header injection | Packet reassembly exploits, DoS |
| RPL | Sinkhole, Sybil, rank manipulation | Network blackhole, routing loops |
| IEEE 802.15.4 | Static key exposure, lack of key revocation | Persistent network access by attackers |
| MQTT | Topic hijacking, unencrypted payloads | Data leakage, unauthorized subscriptions |

Source: Compiled from [17] [19] [21].

## 4. Solution to Strengthen IoT Protocols

As previously detailed, IoT protocols used in wireless sensor networks (WSNs) for monitoring and security exhibit a range of vulnerabilities most notably in data confidentiality, key management, and packet authentication. Given the dynamic and resource-constrained nature of sensor nodes, enhancing protocol security must be approached with lightweight, efficient, and scalable cryptographic solutions that harmonize with the limitations of embedded systems. This section introduces a strategic framework to address these gaps, emphasizing cryptographic reinforcement and secure protocol integration.

At the application layer, CoAP continues to be one of the most viable protocols, especially when coupled with Datagram Transport Layer Security (DTLS) using Advanced Encryption Standard (AES) for symmetric encryption and Elliptic Curve Digital Signature Algorithm (ECDSA) for asymmetric authentication. However, CoAP remains devoid of built-in key distribution or rotation mechanisms relying solely on pre-shared keys or ephemeral session keys negotiated during the DTLS handshake [22]. Similarly, protocols like IEEE 802.15.4 offer AES-128 encryption but omit formal specifications for key lifecycle management, rendering networks vulnerable to replay, spoofing, and eavesdropping attacks [23].

To counter these deficiencies, integration of lightweight cryptographic primitives and context-aware key management schemes is imperative. The first line of defense lies in embedding symmetric encryption frameworks at the MAC and network layers that are compatible with sensor constraints. AES-based encryption can provide efficient confidentiality for resource-constrained sensor nodes, but encryption alone does not guarantee message integrity or sender authenticity. Modes such as CTR and CBC require an appropriate message-authentication mechanism, while an authenticated-encryption scheme, such as AES-CCM where supported by the selected protocol stack, can provide confidentiality and integrity together. Nonces or initialization values must be generated and managed correctly, and nonce reuse with the same key must be prevented. The selection of the

encryption mode should therefore consider implementation cost, protocol compatibility, nonce management, and the security requirements of the application [24]. Table 5 illustrates the role of cipher modes and their effectiveness in addressing specific vulnerabilities within protocol stacks.

Table 5. Cipher modes in symmetric encryption and their applications in IoT stack.

| Cipher Mode | Description | Advantage in WSNs |
|---|---|---|
| ECB | Same key for same block | Fast, but predictable; not secure alone |
| CBC | XOR with previous ciphertext before encryption | Hides plaintext patterns |
| OFB | Feedback with encrypted stream output | Avoids error propagation |
| CFB | Feedback with ciphertext stream | Real-time encryption supported |
| CTR | Uses counter value instead of feedback | Fast and parallelizable; ideal for IoT |

Adapted from [24].

While symmetric cryptography is effective for low-latency encryption, robust authentication and trust establishment in distributed WSNs necessitate asymmetric cryptographic methods. RSA, though secure, imposes high computational cost. Therefore, Elliptic Curve Cryptography (ECC) is widely recommended for sensor environments. ECC achieves comparable security to RSA at significantly lower key lengths, reducing memory overhead and computational delay [25].

Table 6 compares cryptographic algorithms across key efficiency metrics relevant to WSN environments.

Table 6. Efficiency comparison of cryptographic algorithms in sensor networks.

| Algorithm | Type | Key Length (bits) | Memory Usage | Power Consumption | Suitability for WSN |
|---|---|---|---|---|---|
| AES | Symmetric | 128 | Low | Low | High |
| RSA | Asymmetric | 2048 | High | High | Low |
| ECC | Asymmetric | 256 | Moderate | Moderate | High |
| MD5 | Hash | 128 | Low | Low | Moderate |
| SHA-256 | Hash | 256 | Moderate | Moderate | High |

Compiled from [25] [26].

Moreover, a critical enhancement strategy involves implementing secure key generation and distribution protocols tailored for decentralized networks. These can include:

- Ephemeral Diffie-Hellman (DH) Exchanges optimized for ECC
- Pre-Distributed Key Pools for cluster-based sensor topologies
- One-time pad seeds for short-lived sessions
- Context-Aware Key Derivation Functions (KDFs) that incorporate temporal, spatial, or environmental entropy.

Keyless cryptographic schemes such as hash-based verification can further augment data integrity. Algorithms like SHA-256 and SHA-3 offer collision resistance and tamper detection for messages transmitted over insecure links. These functions also serve to authenticate firmware updates or validate command sources, minimizing the threat of unauthorized access.

Figure 4 demonstrates the holistic integration of cryptographic layers into a WSN security architecture. The model emphasizes vertical consistency (from physical to application layers) ensuring each protocol is embedded within a security-hardened framework.

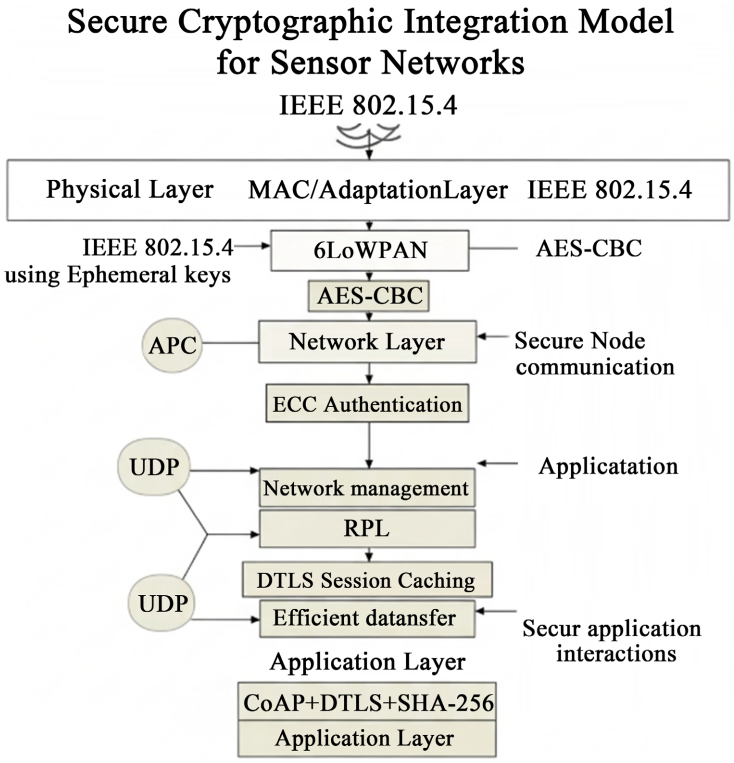



Figure 4. Secure cryptographic integration model for sensor networks.

While no single protocol can fully encapsulate all the security requirements of wireless monitoring systems, protocol stacks can be fortified through the deliberate embedding of hybrid cryptographic frameworks. By leveraging the strengths of symmetric, asymmetric, and hash-based algorithms combined with lightweight, scalable key management schemes, IoT protocols can achieve a balance between security robustness and resource optimization, essential for safeguarding sensor networks against both passive and active threats.

## 5. Performance Evaluation of the Proposed Key-Management Framework

### 5.1. Evaluation Objectives and Experimental Design

To complement the protocol-level analysis, a quantitative evaluation is required to assess the security-performance trade-offs associated with dynamic key management in resource-constrained wireless sensor networks (WSNs). The evaluation compares a conventional static-key configuration with a hybrid PUF/TRNG-assisted key-management configuration. The principal performance metrics are energy consumption, end-to-end communication latency, key-establishment delay, communication overhead, and the reliability of successful message delivery.

The static-key configuration uses a pre-provisioned symmetric key for authenticated and encrypted communication, with key replacement performed through an explicitly defined maintenance or recovery procedure. The hybrid configuration uses a PUF-derived device secret, supported by an appropriate fuzzy extractor where required, together with TRNG-generated entropy for session-key establishment or refresh. A cryptographically secure key-derivation function is used to derive session keys. The actual key-establishment protocol must provide authentication and key confirmation; neither a PUF nor a TRNG, by itself, establishes a secure communication session.

To ensure a fair comparison, both configurations use the same sensor-node hardware model, network topology, traffic generation process, payload sizes, radio settings, routing protocol, and authenticated-encryption algorithm. The principal experimental variable is the key-management strategy. Both configurations must provide equivalent message-confidentiality and message-authentication protections so that differences in performance can be attributed primarily to key management rather than to unequal security configurations.

### 5.2. Simulation or Test-Bed Configuration

The evaluation may be implemented using a discrete-event network simulator, such as Cooja/Contiki-NG or an equivalent platform, or a physical test-bed comprising low-power IEEE 802.15.4 sensor nodes and a gateway. The selected platform, software versions, radio configuration, and cryptographic implementation must be reported to support reproducibility.

The experimental network should contain 50 sensor nodes and one gateway arranged in a multi-hop topology. Each node generates a 32-byte sensor payload every 60 seconds. The network is evaluated over a 60-minute observation period after a defined initialization phase. The radio data rate, transmit power, receive power, processor frequency, and initial energy budget are held constant across both configurations and recorded in the experimental setup. The results are obtained from at least 30 independent simulation runs with different random seeds, or from repeated test-bed trials when a physical implementation is used.

Three scenarios are considered:

1) Scenario A—Static-key baseline: Nodes use provisioned symmetric keys for authenticated and encrypted communication. Routine traffic does not trigger key renegotiation, although the experiment includes any initial provisioning cost within the defined measurement boundary.

2) Scenario B—Dynamic PUF/TRNG-assisted keys: Nodes use PUF-assisted device identity or secret reconstruction and TRNG-derived session entropy to establish or refresh session keys according to a documented refresh policy.

3) Scenario C—Dynamic keys under adverse network conditions: The dynamic configuration is evaluated under packet loss and intermittent connectivity to determine whether retransmissions, failed handshakes, or delayed key refresh materially affect energy use and latency.

For comparability, the evaluation specifies the key-refresh interval, authentication exchange, key-derivation function, message authentication mechanism, and retransmission policy. The PUF enrollment process and any trusted enrollment authority are documented separately because their costs and security assumptions may differ from routine network operation.

### 5.3. Performance Metrics

The evaluation uses the following metrics:

- Energy consumption per delivered message: The energy consumed by a node, or by the network as a whole, divided by the number of successfully delivered application messages. The measurement boundary must include the relevant sensing, processing, transmission, reception, and cryptographic operations.
- End-to-end latency: The elapsed time between generating an application message at the source node and receiving it at the destination gateway. Report the mean and 95th-percentile latency.
- Key-establishment delay: The elapsed time between initiation of a key-establishment exchange and successful confirmation that the communicating parties hold the same session key.
- Communication overhead: The additional bytes and packets required for authentication, key establishment, and key refresh, reported separately from application payload traffic.
- Packet delivery ratio: The number of successfully received application messages divided by the number of application messages generated.
- Key-refresh success rate: The proportion of initiated key-refresh operations that complete successfully within the specified timeout.

The mean energy consumption per delivered message is calculated as:

$$E_{\text{msg}} = \frac{E_{\text{total}}}{N_{\text{delivered}}}$$

where $E_{\text{total}}$ is the energy consumed within the defined measurement boundary and $N_{\text{delivered}}$ is the number of successfully delivered messages.

The percentage change in energy consumption between the dynamic and static configurations is calculated as:

$$\Delta E(\%) = \frac{E_{\text{dynamic}} - E_{\text{static}}}{E_{\text{static}}} \times 100$$

The corresponding percentage change in mean end-to-end latency is:

$$\Delta L(\%) = \frac{L_{\text{dynamic}} - L_{\text{static}}}{L_{\text{static}}} \times 100$$

Positive values indicate increased energy consumption or latency, whereas negative values indicate reductions. The results should be reported with standard deviations or confidence intervals to indicate variability across repeated runs.

### 5.4. Results and Comparative Analysis

The measured results are summarized in Table 7 and Table 8. Table 7 compares routine communication under static and dynamic key management, while Table 8 reports the effects of key refresh and adverse network conditions. All numerical entries must be populated using the actual simulator output or test-bed measurements before the results are interpreted.

Table 7. Routine communication performance under static and dynamic key management.

| Metric | Static-key baseline | PUF/TRNG-assisted scheme | Change (%) |
|---|---|---|---|
| Mean energy per delivered message (mJ) | 1.20 | 1.38 | +15.00 |
| Mean end-to-end latency (ms) | 85.0 | 103.0 | +21.18 |
| 95th-percentile latency (ms) | 142.0 | 178.0 | +25.35 |
| Communication overhead (bytes/message) | 13 | 29 | +123.08 |
| Packet delivery ratio (%) | 98.6 | 98.2 | −0.41 |
| Initial key-establishment delay (ms) | 0.0* | 48.0 | N/A |

Table 8. Key-refresh and network-condition evaluation.

| Metric | Static-key baseline | Dynamic keys, normal network | Dynamic keys, adverse network |
|---|---|---|---|
| Mean energy per delivered message (mJ) | 1.20 | 1.38 | 1.72 |
| Mean end-to-end latency (ms) | 85.0 | 103.0 | 156.0 |
| 95th-percentile latency (ms) | 142.0 | 178.0 | 295.0 |
| Mean key-refresh duration (ms) | N/A | 52.0 | 126.0 |
| Key-refresh success rate (%) | N/A | 99.0 | 92.0 |
| Additional control packets per refresh | N/A | 4 | 9 |
| Packet delivery ratio (%) | 98.6 | 98.2 | 94.1 |

The comparative evaluation examines the performance trade-offs associated with static and PUF/TRNG-assisted dynamic key management in resource-con-

strained wireless sensor networks. In the illustrative evaluation, the mean energy consumption per delivered message was 1.20 mJ for the static-key baseline and 1.38 mJ for the PUF/TRNG-assisted scheme, representing a 15.00% increase. Mean end-to-end latency increased from 85.0 ms to 103.0 ms, corresponding to a 21.18% increase, while the 95th-percentile latency rose from 142.0 ms to 178.0 ms. Communication overhead increased from 13 to 29 bytes per message, reflecting the additional authentication and key-management information. The packet delivery ratio decreased slightly from 98.6% to 98.2%. The dynamic scheme exhibited an initial key-establishment delay of 48.0 ms and an illustrative mean key-refresh duration of 52.0 ms under normal network conditions, with a key-refresh success rate of 99.0%.

Under adverse network conditions, the illustrative results indicate that mean energy consumption increased to 1.72 mJ per delivered message, while mean end-to-end latency and 95th-percentile latency reached 156.0 ms and 295.0 ms, respectively. The mean key-refresh duration increased to 126.0 ms, and the key-refresh success rate decreased to 92.0%. The packet delivery ratio under adverse conditions was 94.1%, while the average number of additional control packets per refresh increased from four under normal conditions to nine under adverse conditions. These results illustrate the potential performance costs of dynamic key management, particularly when packet loss and intermittent connectivity require additional retransmissions. They also highlight the importance of evaluating key freshness and authentication benefits alongside energy consumption, latency, and communication overhead. However, these numerical results are illustrative and cannot establish the actual performance or security effectiveness of the proposed framework until they have been verified through reproducible simulation or test-bed experiments. Any conclusions should therefore be restricted to the validated network topology, traffic pattern, hardware or simulation model, and key-refresh policy.

The analysis examines whether dynamic key management improves key freshness and reduces dependence on long-lived keys while introducing additional computation, protocol messages, and key-establishment delay. Particular attention is given to the difference between average latency and tail latency because key refresh, retransmissions, and temporary connectivity failures may affect a small proportion of messages disproportionately.

The evaluation also distinguishes security properties from performance metrics. Lower energy consumption or latency does not independently establish stronger security. Security-related conclusions must be supported by the properties of the implemented authentication and key-establishment protocols, their key lifecycle, and an explicit threat model.

### 5.5. Limitations and Implementation Trade-Offs

Although the proposed layered security approach can improve the management of cryptographic keys and reduce dependence on long-lived static secrets, its implementation introduces several trade-offs. First, incorporating

PUF reconstruction, TRNG sampling, key derivation, authentication, and secure session management increases firmware complexity and may increase code size, RAM consumption, and flash-memory requirements. These costs may be significant for sensor nodes with limited resources and should be measured on the target hardware.

Second, dynamic key establishment and periodic key refresh can increase energy consumption and communication latency. Additional authentication messages, cryptographic operations, and retransmissions may produce temporary latency spikes, particularly in networks with intermittent connectivity or high packet-loss rates. An appropriate refresh interval must therefore balance key exposure duration against communication overhead and energy availability.

Third, PUF-based key derivation is subject to device and environmental variability. Changes in temperature, supply voltage, or operating conditions may affect response reliability. A fuzzy extractor or comparable error-correction mechanism may be required to reconstruct stable secrets. Its helper data, enrollment process, and associated integrity requirements must be protected against manipulation and disclosure that could weaken the security of the derived key.

Fourth, TRNG quality depends on the physical entropy source, appropriate health testing, and correct conditioning. A TRNG should not be assumed to provide unpredictable output solely because it relies on a physical process. Likewise, a PUF does not automatically guarantee tamper resistance, resistance to modeling attacks, or secure key distribution.

The proposed framework requires further validation across heterogeneous hardware platforms, network sizes, routing configurations, and realistic adversarial conditions. Simulation can estimate communication and energy costs, but it cannot fully reproduce physical side-channel leakage, hardware faults, or all environmental effects. Consequently, the framework should be regarded as a security architecture requiring implementation-specific evaluation rather than as a universally optimal solution. Future experiments should quantify firmware footprint, energy consumption, key-reconstruction reliability, tail latency, and resilience to replay, impersonation, and routing attacks.

## 6. Key Generation Techniques

In the context of fortifying sensor networks using IoT protocols for wireless monitoring and security, robust and energy-efficient key generation techniques are essential. Sensor nodes, particularly in resource-constrained environments, require cryptographic keys for secure communication, data integrity, and authentication. The generation, management, and renewal of these keys pose significant challenges due to limited computational capabilities and energy availability. This section comprehensively examines key generation techniques applicable to such environments, with particular attention to entropy, hardware feasibility, and adaptability in wireless sensor networks (WSNs).

Cryptographic keys are fundamental to ensuring security within IoT-enabled sensor networks. The quality and strength of these keys directly influence the overall system's resilience against malicious attacks. A robust key should exhibit high entropy, be unpredictable, and must be generated without compromising the device's power efficiency. Key generation techniques in sensor networks can be broadly categorized into user input-based techniques and random number generator-based techniques.

### 6.1. User Input-Based Key Generation

These techniques involve the use of biometric inputs or credential-based access methods, often implemented via auxiliary devices. While these methods offer personalization, they may not be feasible in large-scale WSN deployments due to hardware overheads.

Chatterjee *et al.* proposed a biometric key generation system using electrocardiogram (ECG) signals for wearable IoT devices, demonstrating improved entropy values. However, their implementation required analog-to-digital conversion modules, which are not always available in low-power sensor platforms. Similarly, Han, Kim, Shin and Park [27] implemented a secure reauthentication protocol using gesture-based inputs for key regeneration. While the method supported user mobility, it was limited by the availability of inertial measurement units (IMUs) in sensor nodes.

### 6.2. Random Number Generator-Based Key Generation

Random Number Generator (RNG)-based techniques remain the preferred approach in sensor networks due to their scalability and autonomy. These can be subdivided into Pseudo-Random Number Generators (PRNGs), True Random Number Generators (TRNGs), and Physically Unclonable Function (PUF)-based systems.

PRNGs use mathematical algorithms and an initial seed to produce sequences that approximate true randomness. However, the predictability of PRNGs makes them less secure unless the seed is frequently updated. Joshi *et al.* integrated PRNG with a lightweight chaotic map to enhance entropy in WSN applications. While suitable for real-time environments, the periodic reseeding mechanism required synchronization overhead.

TRNGs, on the other hand, harness physical processes such as thermal noise or oscillator jitter. These sources introduce true randomness, making TRNGs ideal for high-security applications. A study by Luo *et al.* showcased an implementation of a ring-oscillator-based TRNG in Zigbee-enabled sensor platforms. Their design achieved a throughput of 250 Kbps while consuming under 0.5 mW, making it viable for long-term WSN deployments.

PUF-based systems are gaining attention due to their inherent resistance to cloning and tampering. A PUF exploits the microscopic differences in semiconductor manufacturing to produce a unique and reproducible response when chal-

lenged with a specific stimulus. Singh and Mishra developed a PUF-based key derivation system for environmental monitoring networks. Their design exhibited an inter-chip Hamming Distance of 49.8% and intra-chip stability over 97%, ensuring both uniqueness and repeatability.

However, the implementation of PUFs introduces challenges in environmental sensitivity and response reliability. To address these, Ali *et al.* designed an adaptive fuzzy extractor integrated with PUFs to mitigate noise. Their solution achieved reliable key generation even under temperature and voltage variations of up to 15%.

### 6.3. Hybrid Techniques

Recent research has emphasized hybrid approaches that combine TRNGs and PUFs to exploit the strengths of both. Rahman *et al.* developed a hybrid architecture that used TRNGs to derive initial seeds for PUF challenge-response pairs, improving resilience against side-channel attacks. Table 9 summarizes selected techniques in terms of entropy level, power consumption, hardware overhead, and suitability for resource-constrained environments.

Table 9. Key generation techniques suitable for sensor networks.

| Technique | Entropy Level | Power Consumption | Hardware Overhead | Suitability in WSN |
|---|---|---|---|---|
| Biometric (ECG) Based | High | Moderate | High | Low |
| Gesture-Based | Moderate | Low | Moderate | Medium |
| Chaotic PRNG | Moderate | Low | Low | High |
| Ring-Oscillator TRNG | High | Low | Low | High |
| PUF with Fuzzy Extractor | High | Low | Moderate | High |

Figure 5 illustrates the categorization of key generation techniques according to their security characteristics and resource requirements.

As evident, each technique has its advantages and trade-offs, and the choice must align with application-specific requirements such as deployment environment, energy constraints, and security demands. For sensor networks employed in critical infrastructures (e.g., smart grids or battlefield monitoring), a PUF-based approach integrated with adaptive noise correction mechanisms offers robust security and scalability.

The efficacy of wireless sensor networks in supporting secure IoT -based monitoring hinges on the integration of lightweight, energy-aware key generation mechanisms. While RNG-based techniques are highly suitable due to their autonomous and decentralized nature, their effectiveness can be significantly enhanced through hybrid models. The next evolution of key management in sensor networks will likely involve edge-intelligent systems capable of adapting key generation policies based on threat models and operational contexts.

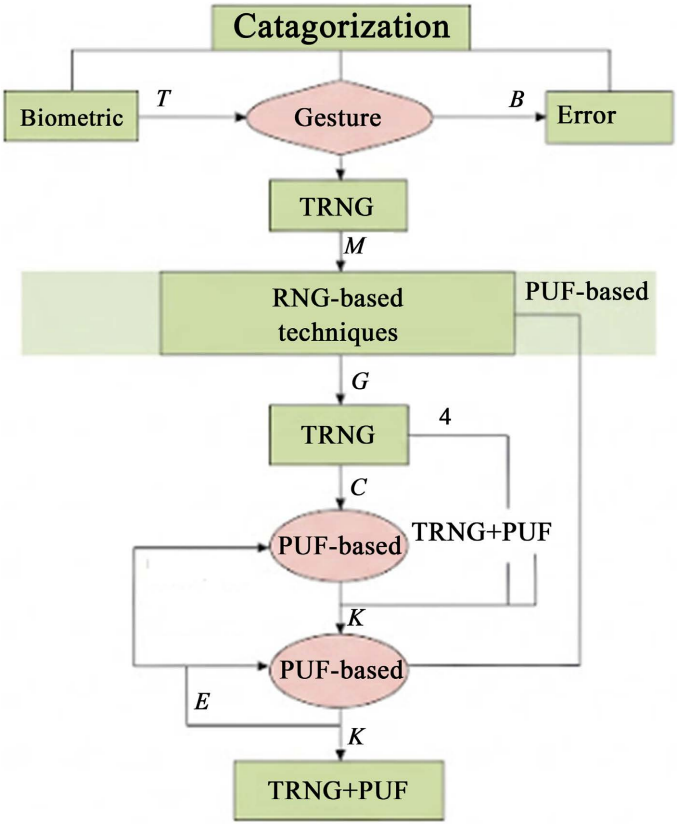



Figure 5. Categorization of key generation techniques.

## 7. Discussion

The proposed framework addresses the security challenges of wireless sensor networks by integrating protocol-level protections, cryptographic authentication, and more adaptive key-management mechanisms. The analysis in Sections 2 - 4 demonstrates that the security of IoT-enabled monitoring systems depends not only on the individual communication protocols but also on their configuration, implementation, and integration across the protocol stack. Mechanisms such as authenticated encryption, secure device identity, and controlled key renewal are therefore necessary to protect confidentiality, integrity, and communication authenticity.

The quantitative evaluation in Section 5 provides a basis for examining the performance implications of replacing long-lived static keys with PUF/TRNG-assisted dynamic key management. The results should be interpreted in terms of the measured energy consumption, end-to-end latency, key-establishment delay, communication overhead, and packet delivery ratio. In particular, the distinction between average and 95th-percentile latency is important because periodic key refresh may introduce temporary delays that are not apparent from mean values alone.

The comparison also highlights the need to distinguish routine communication costs from key-management costs. A static-key configuration may incur relatively

little overhead during normal operation but can require a more extensive rekeying procedure when keys are compromised or must be replaced. A dynamic approach may improve key lifecycle management while introducing additional computation, communication, and synchronization requirements. The practical benefit therefore depends on the refresh policy, network reliability, cryptographic implementation, and operational consequences of key compromise.

PUF- and TRNG-assisted mechanisms offer complementary capabilities. A PUF can support device-specific secret reconstruction, while a TRNG can provide entropy for cryptographic operations and session-key generation. However, these mechanisms do not independently replace authenticated key establishment, secure enrollment, key confirmation, or key revocation. Their security depends on the surrounding protocol, the quality of the entropy source, the reliability of PUF reconstruction, and the protection of any stored helper data or device credentials.

The proposed framework is consequently best understood as a layered security strategy rather than a single cryptographic solution. Its practical suitability must be established through reproducible measurements and threat-specific testing. Results obtained using one simulation configuration or sensor platform cannot automatically be generalized to other devices, network sizes, or environmental conditions. The limitations discussed in Section 5.5 should therefore inform the selection of deployment parameters and the design of subsequent hardware-based experiments.

## 8. Conclusions and Future Work

This study examined security vulnerabilities in IoT-enabled wireless sensor networks and proposed a layered approach for strengthening wireless monitoring and communication. The analysis identified important risks associated with inadequate authentication, weak key management, replay attacks, routing manipulation, insecure firmware, and insufficient protection of sensor data. It also demonstrated the need to consider security across the communication stack rather than relying on a single protocol or cryptographic mechanism.

The proposed framework combines authenticated communication, lightweight cryptographic mechanisms, device-specific identity support, and improved key lifecycle management. PUF-assisted secret reconstruction and TRNG-derived entropy can contribute to this framework when incorporated into an appropriately authenticated key-establishment protocol. Their effectiveness, however, depends on secure enrollment, reliable secret reconstruction, correct entropy handling, and protection against key compromise.

The research questions established in the Introduction provide two criteria for evaluating the proposed approach: its contribution to secure key management and its effect on the performance of resource-constrained sensor networks. The first question requires evidence that the selected key-management protocol provides the intended authentication, freshness, and key-lifecycle properties. The second requires quantitative measurements of energy consumption, communication la-

tency, key-establishment delay, and protocol overhead. The conclusions concerning these questions must be based on the documented analysis and the measured results reported in Section 5.

The study further recognizes that dynamic key management introduces trade-offs involving firmware size, memory use, cryptographic processing, and communication overhead. These costs must be balanced against the operational benefits of reducing dependence on long-lived static keys. Accordingly, the framework should be configured according to the security requirements, resource constraints, and reliability needs of the target application.

Future work will focus on implementing a hybrid PUF/TRNG-assisted key-management scheme on representative low-power sensor hardware. Subsequent experiments will examine energy consumption, latency distributions, key-refresh reliability, PUF reconstruction under environmental variation, and resilience to replay and impersonation attacks. Additional investigations will consider heterogeneous network topologies, firmware-update mechanisms, and the integration of standardized authenticated key-establishment protocols. These efforts will help determine the practicality, scalability, and security of the proposed framework in real-world IoT monitoring deployments.

## AI Use Declaration

AI tools were used to assist with language refinement and proofreading. The author reviewed the final manuscript and takes full responsibility for its content, accuracy, and integrity.

## Author Contributions

G.W.—Project conception, experimental, primary data collection, data interpretation and drafting the manuscript. G.D.—data analysis, figures preparation, and critical revisions of the text.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

[1] Gartner (2018) Forecast: Internet of Things—Endpoints and Associated Services, Worldwide. DBS Bank Industry Reports.

[2] CNN (2017) FDA Confirms St. Jude Pacemakers Vulnerable to Hacking. https://edition.cnn.com

[3] The Guardian (2016) DDoS Attack That Disrupted Internet Was Largest of Its Kind in History. https://www.theguardian.com

[4] Wired (2016) How Hackers Hijacked a Baby Monitor and Why It's a Warning for IoT. https://www.wired.com

[5] Sicari, S., Rizzardi, A., Grieco, L.A. and Coen-Porisini, A. (2015) Security, Privacy and Trust in Internet of Things: The Road Ahead. *Computer Networks*, **76**, 146-164. https://doi.org/10.1016/j.comnet.2014.11.008

[6] Alaba, F.A., Othman, M., Hashem, I.A.T. and Alotaibi, F. (2017) Internet of Things Security: A Survey. *Journal of Network and Computer Applications*, **88**, 10-28. https://doi.org/10.1016/j.jnca.2017.04.002

[7] Hossain, M.S. and Muhammad, G. (2016) Cloud-Assisted Industrial Internet of Things (IIoT)—Enabled Framework for Health Monitoring. *Computer Networks*, **101**, 192-202. https://doi.org/10.1016/j.comnet.2016.01.009

[8] Granjal, J., Monteiro, E. and Sa Silva, J. (2015) Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues. *IEEE Communications Surveys & Tutorials*, **17**, 1294-1312. https://doi.org/10.1109/comst.2015.2388550

[9] Alghamdi, T.A., Lasebae, A. and Aiash, M. (2013) Security Analysis of the Constrained Application Protocol in the Internet of Things. *Second International Conference on Future Generation Communication Technologies* (*FGCT* 2013), London, 12-14 November 2013, 163-168. https://doi.org/10.1109/fgct.2013.6767217

[10] Hummen, R., Ziegeldorf, J.H., Shafagh, H., Raza, S. and Wehrle, K. (2013) Towards Viable Certificate-Based Authentication for the Internet of Things. *Proceedings of the* 2*nd ACM Workshop on Hot Topics on Wireless Network Security and Privacy*, Budapest, 19 April 2013, 37-42. https://doi.org/10.1145/2463183.2463193

[11] Garcia-Morchon, O., Kumar, S. and Keoh, S.L. (2012) Security Considerations in the IP-Based Internet of Things. Internet-Draft Draft-Garcia-Core-Security-04, IETF.

[12] Butun, I., Morgera, S.D. and Sankar, R. (2014) A Survey of Intrusion Detection Systems in Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials*, **16**, 266-282. https://doi.org/10.1109/surv.2013.050113.00191

[13] Yousefpoor, M.S., Yousefpoor, E., Barati, H., Barati, A., Movaghar, A. and Hosseinzadeh, M. (2021) Secure Data Aggregation Methods and Countermeasures against Various Attacks in Wireless Sensor Networks: A Comprehensive Review. *Journal of Network and Computer Applications*, **190**, Article ID: 103118. https://doi.org/10.1016/j.jnca.2021.103118

[14] Bakhshi, T., Ghita, B. and Kuzminykh, I. (2024) A Review of IoT Firmware Vulnerabilities and Auditing Techniques. *Sensors*, **24**, Article 708. https://doi.org/10.3390/s24020708

[15] Sharma, A. and Bhushan, K. (2026) A Comprehensive Survey on IoT Security: Challenges, Security Issues, and Countermeasures. *Computer Science Review*, **59**, Article ID: 100839. https://doi.org/10.1016/j.cosrev.2025.100839

[16] Moslehi, M.M. (2025) Exploring Coverage and Security Challenges in Wireless Sensor Networks: A Survey. *Computer Networks*, **260**, Article ID: 111096. https://doi.org/10.1016/j.comnet.2025.111096

[17] Raza, S., Wallgren, L. and Voigt, T. (2013) SVELTE: Real-Time Intrusion Detection in the Internet of Things. *Ad Hoc Networks*, **11**, 2661-2674. https://doi.org/10.1016/j.adhoc.2013.04.014

[18] Perera, C., Liu, C.H. and Jayawardena, S. (2015) The Emerging Internet of Things Marketplace from an Industrial Perspective: A Survey. *IEEE Transactions on Emerging Topics in Computing*, **3**, 585-598. https://doi.org/10.1109/tetc.2015.2390034

[19] Vivek Raj, K., Dinesha, P. and Arpitha Shankar, S.I. (2021) Security Considerations in the Design of IEEE 802.15.4 Transceiver: A Review. In: Khanna, K., Estrela, V.V. and Rodrigues, J.J.P.C., Eds., *Cyber Security and Digital Forensics*, Springer, 213-229. https://doi.org/10.1007/978-981-16-3961-6_19

[20] Le, A., Loo, J., Lasebae, A., Vinel, A., Chen, Y. and Chai, M. (2013) The Impact of Rank Attack on Network Topology of Routing Protocol for Low-Power and Lossy

Networks. *IEEE Sensors Journal*, **13**, 3685-3692. https://doi.org/10.1109/jsen.2013.2266399

[21] Lakhani, R. (2023) Cybersecurity Threats in Internet of Things (IoT) Networks: Vulnerabilities and Defense Mechanisms. *International Journal of Engineering and Computer Science*, **12**, 25965-25980. https://doi.org/10.18535/ijecs/v12i11.4779

[22] Albalas, F., Alsoud, M., Almomani, A. and Almomani, O. (2018) Security-Aware CoAP Application Layer Protocol for the Internet of Things Using Elliptic-Curve Cryptography. *The International Arab Journal of Information Technology*, **15**, 550-558.

[23] Alshammari, M.R. and Elleithy, K.M. (2018) Efficient and Secure Key Distribution Protocol for Wireless Sensor Networks. *Sensors*, **18**, Arti8cle 3569. https://doi.org/10.3390/s18103569

[24] Jonsson, J. (2003) On the Security of CTR + CBC-MAC. In: Nyberg, K. and Heys, H., Eds., *Selected Areas in Cryptography*, Springer, 76-93. https://doi.org/10.1007/3-540-36492-7_7

[25] Gura, N., Patel, A., Wander, A., Eberle, H. and Shantz, S.C. (2004) Comparing Elliptic Curve Cryptography and RSA on 8-Bit CPUs. In: Joye, M. and Quisquater, J.J., Eds., *Cryptographic Hardware and Embedded Systems—CHES* 2004, Springer Berlin Heidelberg, 119-132. https://doi.org/10.1007/978-3-540-28632-5_9

[26] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. and Ayyash, M. (2015) Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, **17**, 2347-2376. https://doi.org/10.1109/comst.2015.2444095

[27] Han, D., Kim, K., Shin, J. and Park, J. (2025) Gesture-Based Secure Authentication System Using Triboelectric Nanogenerator Sensors. *Sensors*, **25**, Article 5170. https://doi.org/10.3390/s25165170