INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2025, VOL. 71, NO. 3, PP. 1–8
Manuscript received June 2, 2025; revised June 2025. doi: 10.24425/ijet.2025.155451

Lightweight and scalable security for wireless IoT systems: challenges and research directions

Daniel Migwi, and Ryszard S.Romaniuk

***Abstract*—The rise of Wireless Sensor Networks (WSN) has redefined the modern digital infrastructure by enabling real-time sensing, decision making, and automation across diverse sectors. However, this rapid evolution has introduced unprecedented security challenges due to constrained computational resources, heterogeneous device environments, and wide-scale deployment of IoT nodes. This research provides a comprehensive review of lightweight and scalable security mechanisms tailored for wireless IoT systems, with a focus on practical deployment realities. It begins by outlining the security requirements and architectural constraints specific to IoT devices and then evaluates the security capabilities and vulnerabilities of commonly used wireless communication protocols. Emphasis is placed on the limitations of current implementations and protocol-level security inconsistencies. To address these gaps, the paper explores lightweight cryptographic techniques, particularly the NIST-approved Ascon algorithm suite, assessing its adaptability to resource-constrained environments. The discussion extends into scalable key management mechanisms and then investigates the challenges of large-scale deployment. It concludes by identifying future research areas that integrates security within broader system goals.**

***Keywords*—Wireless IoT Security, Scalable Key Management, Ascon-AEAD128, Ascon-Hash256, Ascon-XOF128, Ascon-CXOF128**

I. Introduction

THE Internet of Things (IoT) has emerged as a foundational pillar in the fourth industrial revolution that combines digital transformation and seamless inter-connectivity between the physical world and cyber systems [1]. This seamless inter-connectivity is reliant on wireless communication technologies to support low-power and scalable distributed deployments due to its properties of easy installation and provision of ubiquitous connections [2].

However, as IoT scales in both scope and complexity, security and privacy have become significant concerns. IoT devices are typically characterized by limited resources, low-power processors, and constrained memory that prevents the direct application of conventional cryptographic protocols used in traditional IT systems. Moreover, the wireless medium is inherently vulnerable to threats such as eavesdropping, jamming, spoofing, and man-in-the-middle (MITM) attacks. These risks are magnified in large-scale deployments where physical access to devices is often uncontrolled, and communication patterns are highly dynamic [3].

The challenge is further aggravated by the diversity and fragmentation of IoT systems. Devices differ widely in hardware capabilities, communication protocols (e.g., Zigbee, Bluetooth Low Energy (BLE), Wi-Fi, LoRaWAN and NB-IoT), and operating environments. This heterogeneity makes it difficult to deploy unified, scalable security frameworks because they are either protocol-specific, vendor-dependent, or fail to scale effectively. Key management, firmware updates, and identity verification become complex as systems grow in size, leaving them vulnerable to various forms of cyber attack and operational failures [4].

This research contributes to the development of lightweight and scalable security solutions suited for the constrained IoT environment by presenting an comprehensive evaluation of the following issues:

- *i* Core security requirements and architectural constraints of IoT devices.
- *ii* Security capabilities of common wireless communication protocols.
- *iii* Lightweight cryptographic techniques and scalable key management mechanisms.
- *iv* Challenges in large-scale deployments.
- *v* Future research directions essential for building a secure lightweight scalable WSN.

II. Security Requirements and Architectural Constraints

The security landscape of wireless IoT systems is uniquely shaped by the constraints imposed by the device's architecture, communication models, and deployment environments. Unlike traditional computing platforms, IoT networks operate with restricted resources which demand carefully optimized and lightweight security solutions. In this section, we shall discuss the constraints and requirements that shape wireless IoT security.

A. Core Security Requirements

Several fundamental security requirements in IoT devices that engage in sensing, processing, and communication across wireless channels often in open and/or hostile environments must be satisfied to protect the integrity and trustworthiness of such systems:

D.Migwi is an independent researcher, Poland (e-mail: migwisir@gmail.com - corresponding author; orcid: 0009-0002-8959-5989)
R.S.Romaniuk is with Institute of Electronic Systems, Warsaw University of Technology, Poland (e-mail: ryszard.romaniuk@pw.edu.pl)

i *Confidentiality*: Ensuring that sensitive data transmitted over wireless links remains accessible only to the authorized entities [4].
ii *Integrity*: System must ensure that data in transit is not altered, tampered with, or corrupted [4].
iii *Availability*: System should remain accessible and functional even under extreme conditions [4].
iv *Authentication*: Devices and users must reliably prove their identities before exchanging sensitive data [5].
v *Privacy*: Sensitive user data, including behavioral patterns or location, must be adequately protected against leakage [5].

B. Device-Level Architectural Constraints

IoT devices are typically embedded systems that operate on low-power processors, often with kilobytes of RAM and ROM. Popular platforms such as the ARM Cortex-M series or Atmel AVR microcontrollers are designed for minimal energy consumption and low cost, rather than intensive computations [6]. As a result, implementing complex cryptographic protocols such as RSA or full TLS is impractical for many of these IoT devices.

Nonetheless, many devices operate in sleep-wake cycles to conserve power, making real-time communication or handshake protocols difficult to sustain. Battery life, intermittent power harvesting, and lossy wireless connections further limit the viability of continuous cryptographic handshakes or large key sizes. These constraints require that any cryptographic or authentication scheme be lightweight in both computation and memory footprint while still preserving the necessary levels of security. Consequently, in 2023 NIST announced the selection of the Ascon suite of algorithms for standardization in order to provide efficient cryptography solutions for resource-constrained IoT devices [7].

C. Network Constraints and Scalability Challenges

IoT deployments often involve large-scale and highly dynamic topologies, particularly in applications like industrial sensor arrays or smart grid monitoring. Devices may be added or removed frequently, necessitating flexible and scalable security mechanisms. Traditional centralized key distribution models or certificate-based public key infrastructures (PKI) do not scale efficiently in such settings because:

i They require significant communication overhead to maintain at scale,
ii May introduce single points of failure and
iii Introduce complexities in maintaining synchronization. [8]

Additionally, low data rates, intermittent connectivity, and packet loss further constrain security design especially for protocols that depend on multi-round handshakes, acknowledgments, or long-lived connections.

D. Environmental and Physical Limitations

Unlike conventional computers housed in controlled settings, IoT devices are frequently deployed in unattended, untrusted, or physically hostile environments e.g., outdoor sensor networks, industrial sites, or consumer homes. These conditions increase risks of physical tampering, node capture, and side-channel attacks.

Attackers may exploit unencrypted firmware, unsecured debug interfaces, or extract keys from memory if physical access is gained. Therefore, secure boot, tamper detection, and hardware root-of-trust implementations are recommended, though not always feasible due to cost or complexity [1].

E. Software Heterogeneity and Fragmentation

IoT platforms vary widely in terms of operating systems, protocol stacks (e.g., CoAP, MQTT), and firmware architectures [9]. This software diversity presents challenges in:

i Applying uniform security patches or updates,
ii Ensuring consistent key handling and credential storage and
iii Supporting standards-based cryptographic libraries across the device fleet [4].

This lack of uniformity reinforces the need for flexible and modular security frameworks that can be adapted to different devices without requiring complete rewrites of firmware or network stacks.

F. Recognition of Constraints

Security solutions for wireless IoT must account for more than just cryptographic strength. They must be context-aware, resource-aware, and architecturally compatible with a broad range of device and network profiles. Recognizing and adapting to these constraints is essential to designing scalable, robust, and future-ready IoT security architectures as shown in Fig. 1.

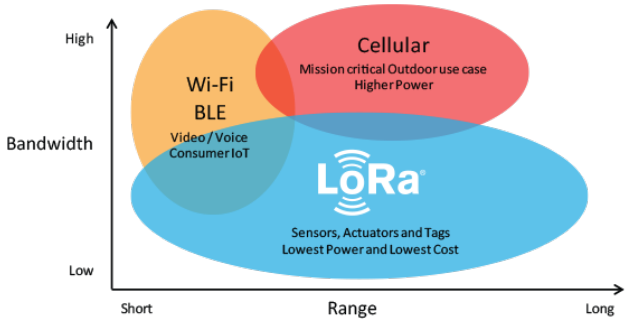

Fig. 1. Comparison between frequency and range supported by the IoT Protocols Source(https://www.semtech.com/lora).

III. Common Wireless Communication Protocols in IoT

These protocols are at the core of IoT devices' network connectivity, enabling data exchange between sensors, actuators, gateways, and cloud systems. This section reviews these commonly used IoT wireless communication protocols, assesses their native security features, and highlights their limitations in practical deployments. The Table III shows a summary of the common IoT wireless communication protocols to be reviewed in this section.

[10], [11]
[10], [11]

TABLE I
COMMON IOT WIRELESS PROTOCOLS OVERVIEW

Type / Range (Meters)	Protocol	Sub-Protocol(s)	Frequency Band	Common Use Case
Proximity *(Upto 0.1m)*	RFID	• **NFC**	13.56MHz	Contactless Payments, Secure Identification
	Bluetooth	• L2CAP • SDP • **Bluetooth Low Energy (BLE)**	2.400 -2.485GHz	Beacons, Monitors, Wearable Devices.
Short-Range *(0.1m - 10m)*	**Zigbee** (IEEE 802.15.4)		2.400-2.485GHz	Home Automation, Lights Control, Sensor Network
Mid-Range *(10m - 100m)*	Wi-Fi (IEEE 802.11 a/b/g/n)	• **Wi-Fi HaLow** (IEEE 802.11ah)	900MHz	Smart Homes, Offices, Industrial Automation
	5G	• **5G for IoT** • 5G NR	30GHz - 300GHz	
Long-Range *(Above 100m)*	LTE	• LTE-M • **NB-IoT**	700-960MHz & 1700-2700MHz	
	LPWAN	• Dash7 • Sigfox • **LoRaWAN**	433MHz (Asia) 868MHz (Europe) 915MHz (America)	Smart Agriculture, Smart Cities, Asset tracking

[10], [11]

A. Overview of Key Protocols

Wireless communication protocols are tailored to suit the unique requirements of the IoT application in terms of range, power consumption, and bandwidth. Most of these protocols are designed to work on the Industry, Scientific and Medical (ISM) band frequencies of 433MHz, 915MHz, 2.4GHz to 5GHz [10] and many more.

i *NFC*: This is a short-span communication technology that uses an unlicensed frequency of 13.56MHz in automatic identification of objects. It allows different data transmission rates with the maximum speed set to 848 kbps within a range of 10 cm [9].

ii *Bluetooth Low Energy (BLE)*: Bluetooth 4.0 specification proposed this protocol optimized as a proximity to short-range, energy-efficient ISM frequency band of communication used widely in wearable and personal devices [9], [10].

iii *Zigbee*: Named after the distinctive movement of bees, it is an energy-efficient protocol designed around the ISM frequency band that bears similarity to BLE but it is cheaper to operate and can have a longer range [10].

iv *Wi-Fi HaLow*: Mid-range protocol that provides higher transmission rates and longer range than BLE. With an ideal hardware configuration operating at the maximum allowed power transmission, the effective range can be extended to 1 km [9].

v *5G for IoT*: Mobile broadband protocol optimized for IoT use case. It utilizes a frequency band that ranges from sub-1GHz to mmWave characterized by very high data rates and a limited range of obstacles penetration [10].

vi *NB-IoT*: A low power consumption, 3GPP cellular-based protocol offering a licensed spectrum access and integration with legacy GSM, GPRS and LTE technologies. Since it operates in a licensed spectrum as LTE, it enables secure and reliable transmissions [9].

vii *LoRaWAN*: Built using the LoRa technology that defines the configuration of the physical layer so as to allow long-range, low-bandwidth wireless protocol designed for outdoor and wide-area IoT applications [9].

B. Security Features of Key Protocols

1) NFC: Most of the NFC tags rely on the limited proximity range supported by the protocol as a form of security. High-end NFC tags used in contactless payments and secure identification, e.g. NTAG 424 DNA tag supports encryption using AES-128 based encryption with 5 customer defined keys. The reader and the tag mutually authenticate each other at the same time, ensuring data remains encrypted over the contactless interface [12].

2) Bluetooth Low Energy (BLE): Offers three pairing modes namely; Just Works™, Passkey Entry (key 0-999,999 padded to 128 bits) and Numeric Comparison (128 bit value exchanged out-of-band), each with varying levels of protection. BLE 4.2 introduced LE Secure Connections with Elliptic Curve Diffie-Hellman (ECDH) key exchange to combat MITM attacks. However, legacy devices still using earlier BLE versions remain susceptible to sniffing and spoofing [13].

3) Zigbee: It uses AES-128 encryption at the MAC layer and includes features such as network key distribution and device authentication. However, default configurations often use shared keys across networks, making it vulnerable to key leakage and replay attacks. Security breaches in commercial Zigbee implementations have exploited hardcoded keys and poor key rotation strategies [14].

4) Wi-Fi HaLow: It uses TCP/IP architecture with IPv4/IPv6 protocols and WPA3 encryption similar to the conventional Wi-Fi standard. Experiences no interference issues with 2.4GHz and 5GHz frequency bands used on the conventional Wi-Fi. Nonetheless, hackers can still intercept data transmitted via the protocol using a packet-sniffer tools if encryption standards are not regularly updated [15].

5) 5G for IoT: Transport Layer Security (TLS) and other specifications for encrypting data in transit are being incorporated into the 5G standard; in contrast, previous cellular standards (2G/3G or 4G) did not specify transit encryption in the core network. On top of data protections and mutual authentication enforced, it supports network slicing where multiple networks can be managed separately. It can still be exploited by a device posing as a 5G tower luring others to join its network [16].

6) NB-IoT: It leverages existing LTE security infrastructure, including mutual authentication, encryption, and integrity protection via 3GPP mechanisms. Subscriber authentication uses SIM-based Authentication and Key Agreement (AKA), offering strong protections even though this introduces dependencies on mobile operators and adds provisioning complexity [17].

7) LoRaWAN: It uses the network and application layers for cryptography using AES-128. The network layer handles

mutual authentication and integrity protection while the application layer handles the end-to-end encryption. The protocol relies on an Over-The-Air Activation (OTAA) or Activation By Personalization (ABP) process. OTAA provides better security, but improper handling of join procedures, static keys, and reuse of session keys in ABP have been exploited in real-world deployments [18].

C. Implementation Limitations and Deployment Challenges

Despite protocol-level security mechanisms, real-world deployments often fall short of the requires level of security due to incomplete or poorly configured implementations:

i Use of default or static keys persists due to ease of provisioning.
ii Optional security features are often disabled to reduce overhead or improve compatibility.
iii In multi-vendor environments, interoperability issues arise from non-standard extensions or incomplete compliance with security specifications.

In addition, many protocols lack forward secrecy, making them vulnerable if long-term keys are compromised. Furthermore, key management mechanisms are rarely automated, increasing operational risks in large deployments [4].

D. Mitigation of Protocol-level Vulnerabilities

IoT communication protocols incorporates foundational security elements but its implementation is often optional, loosely enforced, or inconsistent across devices. To mitigate protocol-level vulnerabilities, system designers must enforce strong default configurations, adopt automated key life-cycle management, and supplement native protections with cross-layer security frameworks.

IV. Lightweight Cryptographic Techniques for IoT

Cryptographic mechanisms form the backbone of a secure communication in IoT networks. However, conventional cryptographic algorithms such as RSA, AES-256, or SHA-3 are computationally intensive making them unsuitable for constrained IoT devices. To address this gap, the field of lightweight cryptography has emerged to provide security primitives tailored for low-power and low-memory environments.

A. Requirements of Lightweight Cryptographic Algorithms.

They are built around the following several optimization principles:

i Minimal code size and RAM/ROM usage to fit within kilobyte-scale memory.
ii Low energy consumption and support for ultra-low-power devices.
iii Reasonable throughput to meet real-time processing needs.
iv Resilience to side-channel attacks, especially given the physical exposure of many IoT devices [19].

These designs must balance the trade-off between performance and cryptographic strength to maintain adequate protection while supporting constrained platforms.

B. Ascon Family of Lightweight Cryptographic Algorithms Overview.

In 2023, NIST released the finalists of its lightweight cryptographic algorithms standardization challenge and the Ascon family of algorithms selected include Ascon-AEAD128, Ascon-Hash256, Ascon-XOF128, and Ascon-CXOF128. They are characterized by lightweight permutation-based primitives and provide robust security, efficiency, and flexibility, making them ideal for resource-constrained environments, such as Internet of Things (IoT) devices, embedded systems, and low-power sensors [7]. In this section we shall review their individual technical and security specifications.

C. Ascon-AEAD128

This is a nonce-based Authenticated Encryption with Associated Data (AEAD) that provides a 128-bit security strength in a single-key setting.

- **Encryption Process**
  - **Inputs**: 128-bit Secret Key, 128-bit Nonce, Associated Data and Plaintext.
  - **Outputs**: Cyphertext whose size equals the plaintext and 128-bit Authentication tag.
  - **Steps**:
    i *Initialization of the State*: A container of size 320-bit is initialized.
    ii *Processing Associated Data*: Non-empty associated data is absorbed into the state and then domain bit separation is applied.
    iii *Processing plaintext*: Plaintext is partitioned into blocks of 128 bits where bitwise XOR and permutation operations are applied to generate a ciphertext block. Concatenating all the ciphertext blocks creates a complete ciphertext.
    iv *Finalization and tag generation*: After loading the secret key, a permutation operation is executed on the state. A bitwise XOR operation between the last 128 bits of the state and the secret key generates the authentication tag.
- **Decryption Process**
  - **Inputs**: 28-bit Secret Key, 128-bit Nonce, Associated Data, Cyphertext and Authentication Tag.
  - **Output**: Plaintext or fail status.
  - **Steps**:
    i *Initialization of the State*: A container of size 320-bit is initialized.
    ii *Processing Associated Data*: Non-empty associated data is partitioned into 128 bits blocks, absorbed into the state and then domain bit separation is applied.
    iii *Processing the ciphertext*: Ciphertext is partitioned into blocks of 128 bits where bitwise XOR and permutation operations are applied to generate a plaintext block. Concatenating all the ciphertext blocks creates a complete plaintext.
    iv *Finalization*: After loading the secret key, a permutation operation is executed on the state. A bitwise

XOR operation between the last 128 bits of the state and the secret key generates the authentication tag. If the computer tag matches the provided one, the computed plaintext is returned otherwise an error is returned [7].

D. Ascon-Hash256

It is a cryptographic hash function that produces a 256-bit hash of the input messages, offering a security strength of 128 bits.

- **Hashing Process**
  - **Input**: Variable-length Message.
  - **Output**: 256-bit digest
  - **Steps**:
    - *i Initialization*: A container of size 320-bit is initialized.
    - *ii Absorbing the message*: Non-empty message is partitioned into 64 bits blocks, absorbed into the state and then domain bit separation is applied.
    - *iii Squeezing the Hash*: Message is partitioned into blocks of 64 bits from which a hash block is extracted followed by a permutation operation until all blocks are processed. Concatenation of the 4 hash blocks generates the 256-digest [7].

E. Ascon-XOF128

It is an XOF, where the output size of the hash of the message can be selected by the user, and the supported security strength is up to 128 bits.

- **Hashing Process**
  - **Inputs**: Variable-length Message, Output length in bits (i.e. $length > 0$).
  - **Output**: <Output length>-bit digest
  - **Steps**:
    - *i Initialization*: A container of size 320-bit is initialized.
    - *ii Absorbing the message*: Non-empty message is partitioned into 64 bits blocks, absorbed into the state and then domain bit separation is applied.
    - *iii Squeezing the Hash*: Message is partitioned into blocks of [<Output length> / 64] bits from which a hash block is extracted followed by a permutation operation until all blocks are processed. Concatenation of the 4 hash blocks generates the <Output length>-digest [7].

F. Ascon-CXOF128

It is a customized eXtendable Output Function (XOF) that allows users to specify a customization string and choose the output size of the message hash. It supports a security strength of up to 128 bits.

- **Hashing Process**
  - **Inputs**: Variable-length Message, Output length in bits (i.e. $length > 0$), Customization string in bits (i.e. $0 < length <= 2048$)
  - **Output**: <Output length>-bit digest
  - **Steps**:
    - *i Initialization*: A container of size 320-bit is initialized.
    - *ii Customization*: Customization string is split into blocks of 64 bits where bitwise XOR and permutation operations are applied to update the state.
    - *iii Absorbing the message*: Non-empty message is partitioned into 64 bits blocks, absorbed into the state and then domain bit separation is applied.
    - *iv Squeezing the Hash*: Message is partitioned into blocks of [<Output length> / 64] bits from which a hash block is extracted followed by a permutation operation until all blocks are processed. Concatenation of the 4 hash blocks generates the <Output length>-digest [7].

This Ascon family of algorithms are designed with efficiency in hardware and simplicity of implementation offering a viable alternative where the Advanced Encryption Standard (AES) may not perform optimally [7].

G. Ascon Authenticated Encryption Algorithms Security Review

Ascon-AEAD128 provides a 128-bit security in plaintext encryption and decryption but cannot be able to hide its length. If the plaintext length is confidential, users must compensate by padding it. A nonce should never be repeated for two or more encryptions with the same secret key since this would make it easier to break the encryption using frequency analysis. The Ascon-AEAD128 has been extended to Ascon-AEAD128a and Ascon-AEAD80pq which offer increased resistance against a quantum adversary using Grover's algorithm for key search but this higher throughput comes at the cost of reduced robustness [7].

H. Ascon Hashing Algorithms Security Review

All Hash and XOF functions provide a 128-bit security against collision, preimage and second preimage attacks. The security of Ascon-XOF128, and Ascon-CXOF128 can be compromised if the Output-length-in-bits parameter is less than 256. Table II shows the collision and Preimage strength in bits.

TABLE II
SECURITY STRENGTHS OF ASCON'S HASHING ALGORITHMS [7]

Function	Output Size (bits)	Security Strength (bits)	
		Collision	Preimage 1 & 2
Ascon-Hash256	256	128	128
Ascon-XOF128	L	min(L/2, 128)	min(L,128)
Ascon-CXOF128	L	min(L/2,128)	min(L,128)

I. Critical Importance of Lightweight Cryptography

Lightweight cryptography is critical to securing the growing landscape of IoT devices. While trade-offs are unavoidable, advancements in hardware-aware cryptographic design and

algorithm standardization are making secure communication more feasible than ever in constrained environments. Ascon family of algorithms are perfect for lightweight cryptographic application in wireless IoT protocols.

V. Scalable Key Management Mechanisms

To establish a Secure Wireless Sensor Network (WSN) connecting to IoT devices, a secure communications channel is required to protect the information flow. Key management therefore becomes the cornerstone of this secure communication in remote IoT systems negotiating security credentials [20]. It encompasses the generation, distribution, storage, renewal, and revocation of cryptographic keys that protect device identities and data integrity. Unlike traditional IT environments, the scale, diversity, and constraints of IoT introduce unique challenges in implementing robust and lightweight key management protocols.

A. Key Management Challenges in IoT

The primary challenges in IoT key management originate from:

- *i* *Device heterogeneity*: Varying memory, power, and processing capabilities.
- *ii* *Scalability*: Supporting secure onboarding and communication among millions of nodes.
- *iii* *Mobility and dynamism*: Devices may join or leave networks unpredictably.
- *iv* *Limited connectivity*: Devices may operate intermittently or with low bandwidth.

Traditional key management techniques, such as X.509 certificate based PKI, are often too resource-heavy and administratively complex for constrained IoT environments [20]. In this section we shall discuss some key management mechanisms used to create secure channels in constrained devices namely; public key cryptography, pre-shared key strategies and group key management systems.

B. Public-Key Cryptography (PKC)

PKC, also known as asymmetric encryption, supports many public key primitives but Elliptic Curve Cryptography (ECC) primitive has become increasingly feasible for use in IoT due to its advances in fast and energy-efficient implementations. The time to execute scalar point multiplication reduced from 34 seconds in 2004 to less than 0.5 seconds in 2009 [20]. ECC also requires a smaller key size than RSA or DSA public key primitives for the same level of security, making it more suitable for use in resource-constrained devices [21]. Below shows the various application of PKC in IoT.

- *i* *PKC based on ECDH (Elliptic Curve Diffie-Hellman).* ECDH is a key exchange algorithm that allows secure sharing of public keys over an insecure communication channel. The standard PKC relies on this key exchange algorithm to securely these keys, [22] combating man-in-the-middle and impersonation attacks.
- *ii* *Identity-Based Encryption (IBE)* binds public keys to device identities, simplifying the trust management. The entity's public keys are derived from some certain aspects of its identity and shared using identity based key exchanges. A trusted third party known as Private Key Generator (PKG), generates the private keys [23], [8].
- *iii* *Certificateless Public Key Cryptography (CL-PKC)* eliminates the need for heavyweight certificate chains by combining the advantage of a standard PKC and IBE. Similar to IBE, a trusted third party known as a Key Generation Centre (KGC), computes partial private keys for users from their identities by using its global secret key and transmits it to the user via a secure channel but unlike PKG in IBE, KGC does not have access to user's private keys [23], [8].

Due to the robust security with asymmetric encryption, digital signatures, and certificate-based authentication various PKC mechanisms provide, they are highly suited for environments with strict security and standardized authentication requirements [22].

C. Pre-shared Key (PSK)

PSK models are simple and widely used in low-end IoT systems due to their minimal computational and communication overhead compared to PKC. IoT devices are provisioned with static symmetric keys during manufacturing or on-boarding when offline. These keys are used for encryption, authentication, or MAC generation [20]. Despite the benefit of negligible computational overhead there are some major drawbacks that should be considered selecting it for a wide scale deployment on IoT infrastructure. All secret information is preloaded before communication starts implying that heterogeneous devices configured differently will not be accepted into the network. A compromised IoT device can bring down the whole network when hacked. Revoking keys already in production is very difficult since each and every device will have to be individually re-keyed when offline to connect back to the network [20].

D. Group Key Management (GKM)

WSN clustered together, introduces a challenge in implementing a unicast communication to individual IoT devices with the limited ISM frequency bandwidths available. Multicast or broadcast communication groups become the most efficient means to send a shared message to a group of devices instead of individual unicast communication with each IoT device that floods the network with duplicate messages. This multicast communication is particularly useful in sending software patches or updates to similar devices [24]. A variety of GKM schemes have been proposed and grouped by the key establishment authority as either centralized, distributed or hybrid methods. Centralized methods are best suited for static WSN topologies unlike the distributed methods that are best suited for dynamic topologies where nodes can leave and join the network randomly [24]. We shall be reviewing the most lightweight centralized GKM schemes applicable to small and large networks.

- *i* *Logical Key Hierarchy (LKH)*: Arranges nodes in a tree structure and symmetric key is assigned to each leaf node

minimizing the rekeying overhead. It guarantees that only the existing members of the group receive the rekeying message but incur computation and storage cost since each member needs to maintain the keys from the leaf to the root node [24].

ii *Secure and Scalable Rekeying Protocol (S2RP)* is similar to LKH but has an added security to authenticate the rekeying message through a one-way hash function [24].

iii *Topological Key Hierarchy (TKH)* is another variant of LKH where the logical tree is mapped to the physical topology of the node reducing the communication cost of total rekeying messages [24].

GKM schemes reduce the complexity of rekeying in large groups but may require significant coordination during random device availability in the network. Another drawback is the majority of these schemes are not designed for resource-constraint devices with limited bandwidth. Worse off is that these schemes require interaction between group members to get access to the shared key which is hard to achieve since IoT topologies have minimal to none peer-to-peer communications [24].

E. Future Designs Considerations

As highlighted in the review above, current scalable and lightweight key management mechanisms for secure IoT systems cannot be applied universally, WSN constraints and invariants need to be considered before selecting a given approach. Future designs ought to support automated, decentralized, and context-aware approaches that minimize administrative burden and energy consumption while maintaining cryptographic robustness.

VI. Challenges in Large-Scale Deployment

As IoT systems scale to encompass millions of interconnected devices, the security and management challenges become significantly more complex. These challenges extend beyond the device constraints and begin to affect the network design, update logistics, anomaly detection, key management, and regulatory compliance. Addressing them is crucial to realizing a truly secure and scalable IoT ecosystem.

1) *Heterogeneity and Interoperability*: Large-scale IoT deployments typically consist of diverse components varying in hardware platforms, communication protocols, firmware, and security capabilities. This diversity makes the implementation of a unified security architecture difficult. Without standardization and enforcement of minimal security baseline models, secure system wide consistency is nearly impossible [5].
2) *Key Management at Scale*: As highlighted in the previous section, scaling cryptographic key management from a few devices to thousands or millions introduces numerous problems. A proper analysis of the WSN requirements, invariants and constraints must be conducted so as to guide on the best lightweight key management mechanism to apply. PKC introduces computation overhead, PSK approaches work best with small networks while GKM requires a cluster of similar sensor nodes [20], [24].
3) *Network Congestion, Latency and Secure Firmware Update Logistics*: At scale, pushing firmware updates securely and reliably becomes a critical yet resource intensive task. Goworko and Wytrebowicz, (2021) proposes a simplified solution to address these concerns. IoT devices are grouped into a subnet that communicates with an edge server via an IoT gateway device. The subnet prevents the IoT traffic from mixing with other traffic from non-IoT devices while the IoT gateway device acts as a communication proxy shielding the subnet from external network attacks [25].
4) *Anomaly Detection and Incident Response*: Real-time anomaly detection is inherently harder in large-scale deployment of IoT devices. Centralized monitoring systems can become overwhelmed while lightweight endpoints often lack adequate resources to conduct local analysis leading to delayed threat responses. Knowledge-base and behavioral-based Intrusion and Detection Systems (IDS) have been tested on general purpose computing systems but their practical implementation in the IoT ecosystem remains limited [4].
5) *Compliance and Policy Enforcement*: Large deployments often span multiple jurisdictions, each governed by its own data privacy and cybersecurity laws such as GDPR, HIPAA, and country-specific IoT mandates. This regulatory diversity requires adaptive security and data governance strategies that harmonize all the existing regulations into a unified security model that guarantees maximum security no matter IoT device constraints, requirements or invariants selected.

A. Non-Universality of Issue of Cryptographic Protection

The challenges of large-scale IoT deployment are more than just technical; they also involve operational scale, regulatory complexity, and architectural flexibility. Addressing all these issues will require a combination of modular security design, adaptive protocols, distributed intelligence, and global compliance strategies.

VII. Conclusion And Future Research Directions

The rapid expansion of the IoT has revolutionized how we collect, share, and act upon data across diverse sectors. From smart cities, healthcare, agriculture to industrial IoT; wireless communication among IoT devices has become a foundational component of modern infrastructure. However, the constrained nature of IoT devices paired with their often unattended, wireless, and large-scale deployments poses a set of challenges different from conventional cybersecurity strategies.

This research has presented a holistic review of lightweight and scalable security mechanisms tailored to the unique demands of IoT WSN. A review of the essential security requirements and architectural constraints that guide design considerations in resource-limited environments provided a basis from which the evaluation of security features in common wireless protocols became logical. Lightweight cryptographic techniques in the NIST-approved Ascon family of hash and encryption algorithms were reviewed after which we then

discussed lightweight key management mechanisms, outlining the evolution from basic pre-shared key models to PKI architectures and GKM schemes. In exploring challenges affecting large-scale deployment, we identified how heterogeneity, scalability, and regulatory fragmentation can erode even the most carefully designed secure systems.

The following areas have been identified as vital research directions for further exploration:

i Static security configurations are insufficient in dynamic IoT environments. AI and machine learning can enable predictive threat detection, anomaly classification, and adaptive access control.

ii IoT systems often handle very sensitive user and operational data. More efficient protocols and hardware support are required for real-world deployments

iii Scalability and interoperability are critical bottlenecks that must be addressed through flexible architectures and stronger industry standards.

iv IoT Security mechanisms often compete with functional requirements for limited device resources. Future research should explore co-optimization methods that balance security, power consumption, and performance under varying operational conditions.

Going forward, collaboration across disciplines including cryptography, embedded systems, networking, machine learning, and policy governance will be essential. Only through such concerted efforts can we build wireless IoT infrastructures that are scalable, efficient, resilient and trustworthy by design.

REFERENCES

[1] D. Migwi, "Trusted computing in the internet of things securing the edge through hardware-enforced trust," *ELEKTRONIKA - KONSTRUKCJE TECHNOLOGIE ZASTOSOWANIA*, vol. 1, pp. 35–42, 05 2025.

[2] J. Zhang, T. Duong, R. Woods, and A. Marshall, "Securing wireless communications of the internet of things from the physical layer, an overview," *Entropy*, vol. 19, p. 420, 08 2017.

[3] S. Pawar, "Iot attack surge: Threats and security solutions — ec-council," Cybersecurity Exchange, 07 2024. [Online]. Available: https://www.eccouncil.org/cybersecurity-exchange/ethical-hacking/the-rise-of-iot-attacks-endpoint-protection-via-trending-technologies/

[4] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying iot security: An exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations," *IEEE Communications Surveys & Tutorials*, vol. 21, pp. 2702–2733, 2019.

[5] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the internet of things: Security and privacy issues," *IEEE Internet Computing*, vol. 21, pp. 34–42, 03 2017.

[6] A. Bhavsar, "A guide for selecting the right microcontroller for your iot project," IIoT World, 02 2018. [Online]. Available: https://www.iiot-world.com/industrial-iot/connected-industry/a-guide-for-selecting-the-right-microcontroller-for-your-iot-project/

[7] S. Meltem, K. Turan, D. Mckay, J. Chang, J. Kang, and Kelsey, "Nist special publication 800 nist sp 800-232 ipd ascon-based lightweight cryptography standards for constrained devices authenticated encryption, hash, and extendable output functions initial public draft," *Ascon-Based Lightweight Cryptography Standards for Constrained Devices*, 11 2024. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-232.ipd.pdf

[8] G. Sharma, S. Bala, and A. Verma, "Extending certificateless authentication for wireless sensor networks: A novel insight," *IJCSI International Journal of Computer Science Issues*, vol. 10, pp. 167–172, 11 2013. [Online]. Available: https://ijcsi.org/papers/IJCSI-10-6-1-167-172.pdf

[9] A. Čolaković, A. Hasković Džubur, and B. Karahodža, "Wireless communication technologies for the internet of things," *Science, Engineering and Technology*, vol. 1, pp. 1–14, 04 2021.

[10] G. Hardesty, "Iot (internet of things) wireless protocols and their frequency bands," Data-alliance.net, 05 2024. [Online]. Available: https://www.data-alliance.net/blog/iot-internet-of-things-wireless-protocols-and-their-frequency-bands/

[11] T. Carpenter, "Ieee 802.15.4 for iot: Power-efficient networking for wireless deployments," Cwnp.com, 04 2025. [Online]. Available: https://www.cwnp.com/ieee-802.15.4-wireless-iot-powerhouse-network-managers/

[12] N. S. B.V., "A powerful mix of security, privacy & trust for nfc in today's iot these highly secure and remarkably powerful nfc tags protect data while enabling advanced functionality, so businesses can introduce smart, digitally connected products for trusted applications at scale. nxp ® ntag ® 424 dna — nxp ntag 424 dna tagtamper secure nfc tags ntag 424 dna — ntag 424 dna tagtamper," 10 2019. [Online]. Available: https://www.nxp.com/docs/en/brochure/NTAG424_BROCHURE.pdf

[13] *Bluetooth: with low energy comes low security*, vol. WOOT'13, Proceedings of the 7th USENIX Conference on Offensive Technologies. USENIX Association, 08 2013. [Online]. Available: https://www.usenix.org/system/files/conference/woot13/woot13-ryan.pdf

[14] T. ZIllner, "Zigbee exploited - the good, the bad and the ugly," 08 2015. [Online]. Available: https://www.blackhat.com/docs/us-15/materials/us-15-Zillner-ZigBee-Exploited-The-Good-The-Bad-And-The-Ugly-wp.pdf

[15] NEWRACOM, "Wi-fi halow is easy to adopt," Newracom.com, 12 2022. [Online]. Available: https://newracom.com/blog/wi-fi-halow-is-easy-to-adopt

[16] J. Shepard, "How does 5g help secure edge connectivity?" 5G Technology World, 12 2024. [Online]. Available: https://www.5gtechnologyworld.com/how-does-5g-help-secure-edge-connectivity/

[17] GSMA, "Security features of lte-m and nb-iot networks," Mobile IoT Security Report, 09 2019. [Online]. Available: https://www.gsma.com/solutions-and-impact/technologies/internet-of-things/wp-content/uploads/2019/09/Security-Features-of-LTE-M-and-NB-IoT-Networks.pdf

[18] A. Mohamed, F. Wang, I. Butun, J. Qadir, R. Lagerström, P. Gastaldo, and D. D. Caviglia, "Enhancing cyber security of lorawan gateways under adversarial attacks," *Sensors*, vol. 22, p. 3498, 01 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/9/3498/htm

[19] M. Katagi and S. Moriai, "Lightweight cryptography for the internet of things," 05 2012. [Online]. Available: https://www.researchgate.net/publication/267246530_Lightweight_Cryptography_for_the_Internet_of_Things

[20] R. Roman, C. Alcaraz, J. Lopez, and N. Sklavos, "Key management systems for sensor networks in the context of the internet of things," *Computers & Electrical Engineering*, vol. 37, p. 147–159, 03 2011. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0045790611000176?via%3Dihub

[21] S. Team, "Rsa, dsa and ecc encryption differences," Sectigo® Official, 01 2021. [Online]. Available: https://www.sectigo.com/resource-library/rsa-vs-dsa-vs-ecc-encryption

[22] M. El-Hajj and P. Beune, "Lightweight public key infrastructure for the internet of things: A systematic literature review," *Journal of Industrial Information Integration*, vol. 41, p. 100670, 08 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2452414X24001158

[23] S. Al-Riyami and K. Paterson, "Certificateless public key cryptography *," 2003. [Online]. Available: https://eprint.iacr.org/2003/126.pdf

[24] P. Nikbakht Bideh, "Lmgroup: A lightweight multicast group key management for iot networks," *Lecture Notes in Computer Science*, pp. 213–230, 2022.

[25] M. Goworko and J. Wytrebowicz, "A secure communication system for constrained iot devices—experiences and recommendations," *Sensors*, vol. 21, p. 6906, 10 2021.