Vuk Amižić [98]
University of Novi Sad, Technical faculty "Mihajlo Pupin", Zrenjanin/Serbia
Igor Vecštejn [99]
University of Novi Sad, Technical faculty "Mihajlo Pupin", Zrenjanin/Serbia
Tamara Milić [100]
Technical College of Applied Sciences in Zrenjanin, Zrenjanin/Serbia

IOT SECURITY TECHNIQUES – PRELIMINARY REVIEW

DOI: 10.5937/PIM26356A

Abstract:
The growth of the Internet of Things (IoT) devices has introduced significant security challenges due to energy efficiency and limited hardware resources. This research paper presents a preliminary review of security techniques applied in IoT systems, focusing on mechanisms for authentication, encryption, and intrusion detection as well as lightweight cryptographic approaches, and device-level security capabilities like Secure Boot (SB) and Trusted Platform Module (TPM). This review highlights current trends and common vulnerabilities, aiming to provide an improved understanding of IoT security strategies and challenges.

Keywords: *Internet of Things, Security Techniques, Secure Boot, Trusted Platform Module, Review*

Introduction

The Internet of Things represents a modern paradigm that enables direct human-free communication between various sensors and devices. These devices track and collect data on technological systems as well as human social behavior, providing access to a huge amount of smart endpoints that exchange data for services such as home automation systems, connected health care, security surveillance, agriculture, and critical infrastructure control [1].

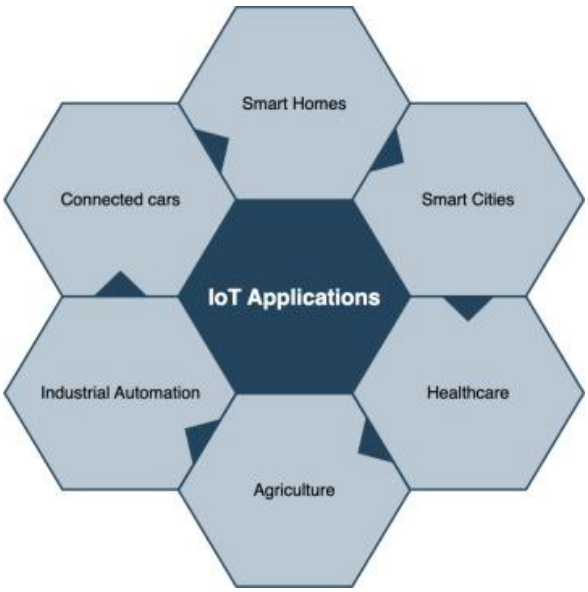
The Internet of Things connects the physical and digital worlds through sensors and actuators that collect data for processing. Because these devices are linked together, a threat to one device can spread to others and cause serious harm. This makes trustworthiness a key requirement for IoT devices to be widely adopted [2].

The growing number of connected devices has also led to more security threats, especially from botnets. To deal with these threats, machine

[98] vuk.amizic@uns.ac.rs, ORCiD: 0009-0003-7575-8473
[99] igor.vecstejn@uns.ac.rs, ORCiD: 0009-0009-9940-4411
[100] tamara.milic@vts-zr.edu.rs, ORCiD: 0009-0004-7273-4801

learning and deep learning methods have been used to detect IoT botnet attacks. These methods tend to perform better than older signature-based approaches. However, how well they work depends on the data set used, the features chosen, and the way results are measured. Overall, machine learning and deep learning show strong potential for spotting botnet activity in IoT networks, which makes them a useful tool for improving IoT security [3].

Many IoT devices still contain visible and hidden vulnerabilities, making them attractive targets for malicious actors. Such attacks may cause infrastructure damage, service disruption, and financial losses; therefore, detecting and isolating vulnerable devices before compromise is an important security measure [4].



Picture 1. *IoT Application Systems*

1. IoT Security Techniques

IoT devices are frequent targets of cyberattacks because they are interconnected, collect sensitive data, and often operate with limited resources. Secure data transmission is therefore necessary to prevent unauthorized access and data breaches. However, conventional cryptographic algorithms may be too resource-intensive for endpoint devices, which makes lightweight security techniques especially important [5].

1.1. Lightweight Cryptography

According to [5], the National Institute of Standards and Technology hosted a lightweight cryptography workshop in 2015 to address security and resource requirements needed for securing IoT ecosystems. A total of fifty-six algorithm proposals were received and published. Among the structures commonly used by lightweight cryptography algorithms are SPN networks, hash functions, and authentication techniques customized for target platforms.

1.1.1. SPN Networks

Substitution-permutation networks divide input into blocks and apply sequential substitution and permutation operations. Due to efficient memory and computational requirements, SPN structures are suitable for resource-constrained IoT devices and are often combined with Feistel architectures to reduce encryption time and improve energy efficiency [5, 6].

1.1.2. Sponge-based

SPONGENT is a lightweight hash function built on sponge construction with small state sizes. It uses S-boxes and linear diffusion that are optimized to keep gate count low. This makes it a good fit for ASIC and FPGA hardware. However, on small microcontrollers, SPONGENT is less effective in software due to reliance on lookup tables and bit-sliced permutations [7].

Sponge construction was introduced as a component of the Keccak hash algorithm, which was selected for the SHA-3 competition [5].

1.1.3. ARX Ciphers

ARX ciphers are symmetric-key algorithms based on modular addition, bitwise rotation, and XOR operations [5]. Their design avoids lookup tables and data-dependent memory access, which helps reduce timing-related vulnerabilities. Because they are compact, portable, and suitable for constant-time implementation, ARX constructions are a practical option for lightweight hashing and encryption on small microcontrollers [7].

1.1.4. AEAD Ciphers

Authenticated encryption with associated data (AEAD) provides both authentication and encryption in a single operation. This is achieved by encrypting plaintext and authenticating internally using an authentication tag. The advantage of this approach is that it ensures the

integrity of communication while keeping code size small and using less computational time [5].

2. Trusted Platform Module

The Trusted Platform Module, often called TPM2 in its current version, is a dedicated piece of hardware or software built around a cryptographic standard. Its job is to handle secure tasks like making random numbers, creating and storing keys, and performing encryption or decryption [8].

Trusted computing, more specifically the Trusted Platform Module, is seen as a promising way to deal with security vulnerabilities found in different layers of IoT design. TPM technology can improve node authentication without using too much energy. It offers benefits both in processing time and energy use. Performance tests have shown that TPM provides clear security improvements while keeping energy usage under control [9].

What makes a TPM useful is that all sensitive work stays inside the module. Private keys and secrets are never copied out to the main system memory. Everything happens behind a wall that the rest of the device cannot cross. Some versions of TPM are even designed to notice if someone tries to open the casing or physically interfere. When that happens, the module wipes its own contents to keep the secrets from falling into the wrong hands. TPMs can show up in two main ways. The most common is a separate chip soldered onto the device board. Another approach runs TPM functions as software within a safe zone of the processor, similar to how ARM TrustZone operates. Either way, the device talks to the TPM using a fixed set of rules laid out by the standard [8].

Beyond basic certificate checks, remote attestation adds another layer by verifying the software state of an IoT device and detecting any unwanted changes. In this process, one device sends proof of its software or memory condition to another device or server for validation. Remote attestation depends on Trusted Platform Modules to create those proofs. For IoT devices that include a TPM, this gives manufacturers and service providers more confidence in certificate-based authentication [10].

2.1. Trusted Platform Module and OPC UA Protocol

In setups that rely on the OPC UA protocol for machine-to-machine communication, a TPM can take over two specific duties. First, it holds

onto the asymmetric keys needed for proving identity. Second, it performs the actual signing and decryption steps required during a secure handshake. To make this work, the TPM must first be given an asymmetric key pair. The module creates the pair internally and allows the public half to be read out while the private half remains permanently locked away [8].

After that, the public part gets signed by a trusted certificate authority. That signed public key becomes the official proof of identity for whichever client or server is trying to connect [8, 10].

Once conPictured, the device can safely obtain group keys for publishing and subscribing, and no attacker can pull the private key out of the device because it never existed outside the TPM in the first place. A TPM keeps private keys out of reach as they do not sit in permanent storage. They do not go into active memory, and no one ever sees or handles them. This minimizes the areas that an attacker can go after. Without a hardware security module, private keys end up stored in memory or on disk at some point, and they become easier to grab. However, with a TPM in place, the risk is significantly lowered [8].

2.2. Secure Boot in IoT

Traditional Secure Boot uses one-way authentication and assumes the device starts in a safe state. However, with complex hardware supply chains, components can be tampered with before they even reach the user. In such cases, one-way authentication is not enough. Also, relying on Public Key Infrastructure adds overhead when managing certificates for many devices [11].

Secure boot is a process that protects the bootloader chain from tampering. It works in stages, where each stage checks the next one before handing over control. The starting point is called the root of trust, which is usually an unchangeable piece of code stored in on-chip ROM. Each stage carries a digest that gets verified before execution, forming a chain of trust. For small IoT devices with limited resources, complex solutions like UEFI cannot be applied. This makes custom lightweight approaches more suitable for secure boot in IoT devices [12].

Secure Boot on devices with low hardware resources cannot follow the same approach used on full computing platforms. IoT devices often lack large memory, dedicated cryptographic hardware, and secure key

storage. A lightweight Secure Boot design must keep the core properties of secure startup while working within these tight limits [13].

2.2.1. Hardware-based Secure Boot

Hardware secure boot relies on a physical root of trust built into the device. One approach uses ARM TrustZone to create an isolated execution environment, with hash values stored in eFuse acting as the starting point for a trusted chain. This chain uses hardware encryption to block tampering during startup. Another method replaces traditional hash verification with execution time benchmarks, checking boot sequence timing instead of cryptographic hashes. A fully hardware implementation using post-quantum signatures like XMSS has also been tested on FPGA platforms and shown to run faster than mixed hardware-software versions. Trusted Platform Modules provide secure key storage and integrity checks for devices like vehicle ECUs. Some designs use Physical Unclonable Functions to generate encryption keys from hardware delay paths, so any tampering with the boot image changes the key and stops the boot process. Hardware schemes are generally more secure and boot faster, but they require extra hardware resources, which raises cost and limits use to larger devices with more power [14].

2.2.2. Software-based Secure Boot

Software secure boot does not use a hardware root of trust. Instead, it places the bootloader in read-only memory and stores cryptographic hashes in protected storage to verify the operating system kernel. The bootloader checks kernel integrity before allowing the system to continue. This approach increases boot time slightly but uses fewer resources. Without a hardware anchor, software methods are more open to attacks and take longer to complete the secure boot process [14].

2.2.3. Hybrid Secure Boot

Hybrid methods combine elements of both hardware and software to balance security and resource use. One example uses a SIM card to run TPM 2.0 functions without adding separate hardware, binding the SIM-based module to the on-board trusted execution environment. Another approach uses a Memory Protection Unit that provides authenticated encryption for memory regions, creating dynamic trusted zones during boot. Hybrid schemes offer a middle ground. They use less hardware than pure hardware methods and boot faster than pure software methods [14].

3. Conclusion

This paper provided a preliminary review of security techniques used in IoT systems, with a focus on lightweight cryptography, Trusted Platform Module, and Secure Boot mechanisms. Lightweight cryptography offers practical solutions for resource-constrained devices through SPN networks, sponge-based hashing, ARX ciphers, and AEAD schemes. These approaches reduce computational overhead while still maintaining acceptable security levels. TPM technology adds hardware-backed protection by keeping private keys isolated from system memory and enabling secure node authentication. Secure Boot ensures that devices start from a trusted state by verifying each stage of the boot process, with hardware, software, and hybrid implementations available depending on device capabilities and cost constraints. Overall, combining lightweight cryptographic primitives with hardware security modules and verified boot processes provides a solid foundation for securing modern IoT environments. Future work may focus on developing unified frameworks that integrate these techniques while keeping energy use and boot latency low for ultra-constrained devices.

REFERENCES

[1] Thabit, F., Can, O., Aljahdali, A. O., Al-Gaphari, G. H., & Alkhzaimi, H. A. (2023). Cryptography algorithms for enhancing IoT security. Internet of Things, 22, Article 100759.

[2] Román, R., & Baturone, I. (2023). A quantum-resistant and fast secure boot for IoT devices using hash-based signatures and SRAM PUFs. In A. Nayyar, A. Paul, & S. Tanwar (Eds.), The Fifth International Conference on Safety and Security with IoT (pp. 121–136). Springer International Publishing. https://doi.org/10.1007/978-3-030-94285-4_8

[3] Nazir, A., et al. (2023). Advancing IoT security: A systematic review of machine learning approaches for the detection of IoT botnets. Journal of King Saud University – Computer and Information Sciences, 35(10), Article 101820.

[4] Doghramachi, D., & Ameen, S. (2023). Internet of things (IoT) security enhancement using XGBoost machine learning techniques. Computers, Materials & Continua, 77(1), 717.

[5] Radhakrishnan, I., Jadon, S., & Honnavalli, P. B. (2024). Efficiency and security evaluation of lightweight cryptographic algorithms for resource-constrained IoT devices. Sensors, 24(12), Article 4008.

[6] El-Shafai, W., Mesrega, A. K., Ahmed, H. E. H., Elbahnasawy, N. A., & El-Samie, F. E. A. (2025). Hybrid lightweight encryption for IoT: Integrating chaotic key generators with Feistel and substitution-permutation networks for secure 3DV transmission. Multimedia Tools and Applications, 84(32), 40147–40193. https://doi.org/10.1007/s11042-024-19203-x

[7] Amal, P. (2025). CHX-128: A lightweight ARX-based hash for resource-constrained IoT devices. Authorea Preprints.

[8] Gilles, O., Pérez, D. G., Brameret, P.-A., & Lacroix, V. (2023). Securing IIoT communications using OPC UA PubSub and trusted platform modules. Journal of Systems Architecture, 134, Article 102797.

[9] Papakotoulas, A., Mylonas, T., Panagidi, K., & Hadjiefthymiades, S. (2024). Optimizing IoT security via TPM integration: An energy efficiency case study for node authentication. ITU Journal on Future and Evolving Technologies, 5, 73–87.

[10] Sharma, G., Joshi, A. M., & Mohanty, S. P. (2023). Fortified-grid: Fortifying smart grids through the integration of the trusted platform module in internet of things devices. Information, 14(9), Article 491.

[11] Feng, Y., et al. (2023). IBCEB: A bi-authentication secure boot scheme for IoT device based on IBC. In 2023 3rd International Conference on Electronic Information Engineering and Computer Science (EIECS) (pp. 1314–1318). IEEE. Accessed April 20, 2026.

[12] Cano-Quiveu, G., Ruiz-de-Clavijo-Vazquez, P., Bellido, M. J., Juan-Chico, J., & Viejo-Cortes, J. (2023). IRIS: An embedded secure boot for IoT devices. Internet of Things, 23, Article 100874.

[13] Danchenko, O., Rozlomii, I., Yarmilko, A., & Naumenko, S. (2025). A lightweight secure boot mechanism for protecting the firmware of IoT devices.

[14] Wang, R., & Yan, Y. (2022). A survey of secure boot schemes for embedded devices. In 2022 24th International Conference on Advanced Communication Technology (ICACT) (pp. 224–227). IEEE. Accessed April 20, 2026