Key challenges in LoRaWAN-based Edge-Cloud infrastructures for security-sensitive smart cities applications

Maurizio Giacobbe[1,*,†], Ida Falco[2,†], Sarah Zanafi[1,†], Carmine Colarusso[2,†], Jiregna Abdissa Olana[1,†], Antonio Puliafito[1,†] and Eugenio Zimeo[2,†]

[1]University of Messina, Department of Engineering, Messina 98166, ITALY
[2]University of Sannio, Department of Engineering, Benevento 82100, ITALY

Abstract

The adoption of IoT technologies in smart cities has led to the development of hybrid edge-cloud architectures that balance the computational demands of real-time data processing with the scalability of cloud services. While these architectures enable innovative services and improved urban management, they also introduce significant cybersecurity challenges especially when recent technologies are used. This paper explores these challenges, when a low power communication technology, such as LoRaWAN is used as a smart city backbone infrastructure. To this end, we first discuss best practices derived from international standards such as ISO/IEC 27001 and LoRaWAN 1.1 specifications, providing a detailed analysis of these issues, then a specific case study is considered to show the application of the presented best practices in a smart city scenario, demonstrating their effectiveness in mitigating risks while preserving the operational efficiency of LoRaWAN networks. By analysing LoRaWAN security challenges, including potential threats such as data interception, unauthorized access, and denial-of-service attacks, this study sheds light on the risks facing smart city digital infrastructures with the aim of building resilient and trustworthy services for urban stakeholders.

Keywords

Cybersecurity, Hybrid edge-cloud, IoT, LoRaWAN, Smart Cities

1. Introduction

The adoption of Internet of Things (IoT) technologies has revolutionized numerous industries by enabling real-time data collection, processing, and analytics. Among IoT communication protocols, Long Range Wide Area Network (LoRaWAN) [1] has gained prominence due to its ability to facilitate low-power, long-range communication. This makes it ideal for use cases like smart agriculture, industrial automation, and environmental monitoring. However, the decentralized nature of IoT ecosystems [2], where edge devices communicate through gateways and interact with cloud platforms, introduces significant security challenges. Ensuring secure communication, data integrity, and privacy is crucial for ensuring trust and reliability in smart cities applications based on LoRaWAN networks [3].

LoRaWAN has been thought for ensuring secure communication. The LoRa Alliance [4] has developed a security framework emphasizing end-to-end encryption, mutual authentication, and message integrity to mitigate security risks. Confidentiality is achieved through efficient encryption mechanisms, such as AES-128 combined with separate network and application keys (NwkSKey and AppSKey) [5], but the implementation of security guidelines can vary, leading to inconsistencies and potential vulnerabilities, such as replay attacks, key compromise, and gateway spoofing. International standards play a pivotal role in addressing IoT security challenges. ISO/IEC 27001 [6] provides a comprehensive framework for

Joint National Conference on Cybersecurity (ITASEC & SERICS 2025), February 03-8, 2025, Bologna, IT
*Corresponding author.
†These authors contributed equally.
mgiacobbe@unime.it (M. Giacobbe); i.falco@studenti.unisannio.it (I. Falco); sarah.zanafi@unime.it (S. Zanafi); ccolarusso@unisannio.it (C. Colarusso); jiregnaabdissa.olana@unime.it (J. A. Olana); antonio.puliafito@unime.it (A. Puliafito); zimeo@unisannio.it (E. Zimeo)
0000-0001-6178-7132 (M. Giacobbe); 0009-0004-9507-1676 (I. Falco); 0000-0002-0126-7837 (S. Zanafi); 0000-0002-0914-1315 (C. Colarusso); 0009-0002-2220-1359 (J. A. Olana); 0000-0003-0385-2711 (A. Puliafito); 0000-0003-4683-5487 (E. Zimeo)

managing information security, while the NIST Cybersecurity Framework [7] offers guidelines for risk management in distributed systems. For IoT-specific scenarios, NISTIR 8259 [8] and ETSI EN 303 645 [9] define best practices, including secure boot, firmware updates, and key management. In industrial contexts, IEC 62443 [10] outlines measures to secure operational technology (OT) environments, including edge-cloud architectures. These standards collectively provide the foundation for building secure IoT systems, but they often require adaptation to address the unique features of LoRaWAN.

The integration of edge computing and cloud platforms enhances the scalability and efficiency of IoT systems. In LoRaWAN architectures, edge devices and gateways preprocess data locally before transmitting it to cloud services for storage and analysis. This hybrid approach (i.e., *hybrid edge-cloud architecture*) reduces latency and bandwidth usage but also creates additional attack surfaces. Zero Trust Architecture (ZTA) [11] and Security by Design [12] principles have emerged as key paradigms to ensure that every component in the system is verified and trusted.

With the growing complexity of hybrid IoT systems, continuous monitoring and rapid incident response are critical. Frameworks like NIST SP 800-61 [13] provide guidelines for managing cybersecurity incidents, including threat detection, containment, and recovery. Security Information and Event Management (SIEM) tools and automated Security Orchestration, Automation, and Response (SOAR) platforms are increasingly used to address threats in real-time.

As IoT data often crosses geographical boundaries, compliance with regulations such as the General Data Protection Regulation (GDPR) [14] and the EU Cybersecurity Act [15] is essential [16]. These regulations ensure data privacy and accountability, which are particularly relevant in LoRaWAN systems due to the diversity of devices and jurisdictions involved.

This work explores new security concerns related to the adoption of edge-cloud architectures, mediated by LoRaWAN networks, for implementing critical smart cities applications. By addressing security at multiple levels device, network, and cloud this study aims to provide a comprehensive approach to safeguarding LoRaWAN systems in real-world applications.

The structure of this paper is as follows: Section 2 reviews related work; Section 3 discusses a LoRaWAN-based Edge-Cloud architecture and presents Stack4Things as a framework to extend cloud capabilities to IoT; Security concerns related to the adoption of LoRaWAN are presented and discussed in Section 4; Conclusion ends the paper.

2. Background and Related Work

IoT networks exhibit various vulnerabilities across different layers of their architecture [17], each of which can be exploited to cause damage to the network or its users. Understanding and addressing these weaknesses is key to developing secure and reliable IoT systems.

2.1. From sensors to application-level vulnerabilities

The sensing layer, also known as the physical layer, includes a wide range of devices, such as sensors, actuators, and other smart devices. Many attacks at this layer exploit the inherent weaknesses of these devices, which often have limited power, low processing capabilities, and insufficient security mechanisms. As this layer is responsible for generating the data that flows through IoT systems, it is crucial that data generation, forwarding, and reception be secure.

IoT devices often work without tamper resistance policies, so malicious devices can replace legitimate ones or take control of existing nodes. This can lead to unauthorized data collection, manipulation, or disruption of services. Wurm et al. [18] show how physical access to a device may allow attackers to modify specific boot parameters and extract the root password. IoT devices, often battery-powered, are vulnerable to attacks that aim to drain battery life or cause physical damage due to environmental factors; manipulating ambient energy can be exploited to launch attacks on battery-less IoT devices, causing denial-of-service and starvation [19]. Another case is the node-capturing attack, in which the attacker can create a malicious IoT node to substitute one actual node [20]. The network layer is responsible for communication between IoT nodes, including data routing and congestion management.

Suppose network devices such as gateways or access points cannot manage high packet flows. In that case, they can be targeted in Denial of Service (DoS) attacks, making the network unavailable to legitimate devices [21].

The application layer provides services to end-users, and vulnerabilities here often affect the functionality and security of the entire system. Insufficient input validation can lead to attacks where malicious data is inserted into the system, such as SQL injection or other forms of data manipulation, as shown by Noman et al. for Wireless-Based IoT [22] . Using outdated or weak encryption protocols exposes the system to attacks that exploit known vulnerabilities. Message Queue Telemetry Transport (MQTT), a widely used IoT messaging protocol presents authentication, authorization, message delivery, and plaintext message exchange vulnerabilities. Instead, the CoAP protocol presents some vulnerabilities in bootstrapping, which could grant unauthorized nodes access to a CoAP environment [23]. Finally, for distributed IoT applications that need data synchronization and reconciliation [24], the Byzantine fault tolerance problem arises. Considering a network where IoT nodes try to reach a consensus, some nodes may intentionally provide incorrect votes to sabotage the decision-making process, leading to a Byzantine attack [25].

2.2. Wireless communication and security concerns

Wireless communication, especially in IoT systems, is a primary vulnerability. Attackers can intercept and manipulate data in transit without proper encryption, leading to data exposure or corruption. For example, Wood et al. [26] show how medical IoT devices may reveal sensitive data and metadata about users' behaviour and medical conditions. Incorrect routing or the manipulation of packet forwarding can lead to data loss or interception. Routing protocol for low power and lossy networks (RPL), a network layer protocol for IoT devices, is vulnerable to various attacks, including selective forwarding, blackhole, sybil, wormhole, and sinkhole attacks [27], [28]. Gateways play a critical role in handling communication between IoT devices and external networks, making them prime targets for attackers.

Since its initial specification in 2015, LoRaWAN has integrated security as a core component of its design. It has faced various security challenges, many of which have been identified and addressed. The most significant revision occurred with version 1.1, introducing the Join Server, implementing Network Server roaming, and enabling firmware updates, enhancing its architecture. Despite these advancements, LoRaWAN remains susceptible to general IoT vulnerabilities, with some posing heightened risks due to the protocol's specific characteristics. LoRaWAN enables long-distance connections: End Device can be placed in remote and infrequently visited locations, which might not be monitored. This vulnerability exposes the devices to tampering, physical damage, and data injection attacks.

Key management is a fundamental aspect of LoRaWAN security. As for the LoRaWAN specification, in the ABP (Activation by Personalization) mode, keys are pre-shared and saved as configuration files or hardcoded. If the root keys are compromised, the attacker can control or impersonate the device's communications for its entire lifecycle. However, with the OTAA (Over-The-Air Activation) mode, the activation happens on request, dynamically generating session keys during the join process, and the attacker can control the communications only until the device performs a new join. This method offers the highest reliability and security, however, the activation phase requires special attention because it involves steps without encryption. In LoRaWAN, a DevNonce is a value used during the OTAA process to ensure the uniqueness of each Join Request. It prevents replay attacks by ensuring that each Join Request is distinct. The Network Server (NS) tracks the DevNonce values to reject any repeated requests from the same device, preventing session key generation and maintaining secure communication.

LoRa's physical layer has a vulnerability that allows attackers to build covert channels by embedding information using an orthogonal modulation scheme to LoRa's chirp spread spectrum (CSS), as demonstrated by the CloakLoRa [29] implementation. This method, undetectable by current LoRaWAN security mechanisms, allows attackers to transmit hidden information over distances up to 250 meters without disrupting regular LoRa communications.

Beacon spoofing exploits the absence of authentication in LoRaWAN Class B beacons, enabling attackers to transmit fake beacons. By introducing spoofed beacons with timing offsets, attackers can

desynchronize devices, causing them to miss valid downlink messages or open their receive windows at incorrect intervals [30].

Moreover, an attack could exploit weaknesses in the Message Integrity Code (MIC) and the connection between the gateway and the Network Server forging valid packets. The MIC, a 4-byte integrity code calculated using the NwkSKey, ensures packets are authentic, but an attacker can brute force the MIC by trying all 4 billion combinations. The attack also targets the frame counter (FCnt). FCnt ensures packet validity in LoRaWAN by incrementing with each transmission. An attacker can exploit the 16-bit FCnt by brute-forcing the Message Integrity Code (MIC) for different FCnt combinations. Additionally, if over 65,535 packets are missed (e.g., due to jamming), the FCnt desynchronizes with the NS, causing a DoS as packets can no longer be validated [31].

3. A LoRaWAN-based Edge-Cloud Architecture

Hybrid edge-cloud architectures have emerged as a foundation for modern distributed systems, necessitating robust strategies for securing containerized microservices [32]. Edge-Cloud architecture based on LoRaWAN networks represents today a reference model for implementing infrastructures for smart city applications. In this section, we sketch such an architecture emphasizing the communication and the ingestion layers between the physical world and the cloud.

3.1. IoT Edge-Cloud system architecture

Fig. 1 represents the proposed IoT system architecture to enable edge-to-cloud continuum. It adopts a layered approach, combining edge, fog, and cloud computing to manage data collection, processing, and decision-making.

The Stack4Things (S4T) open-source framework [33] allows the scalability of IoT networks by enabling seamless interaction between edge and fog layers, allowing for efficient processing and synchronization of data across the system. The LoRaWAN transmission packet is relatively small, which helps optimize energy consumption and enables long-range communication. However, the limited amount of data sent in a single transmission requires appropriate preprocessing at the edge, ensuring that only essential information is sent to the fog or cloud layers. In smart city IoT applications, edge devices can filter, aggregate, or compress data locally, preserving power and network resources.

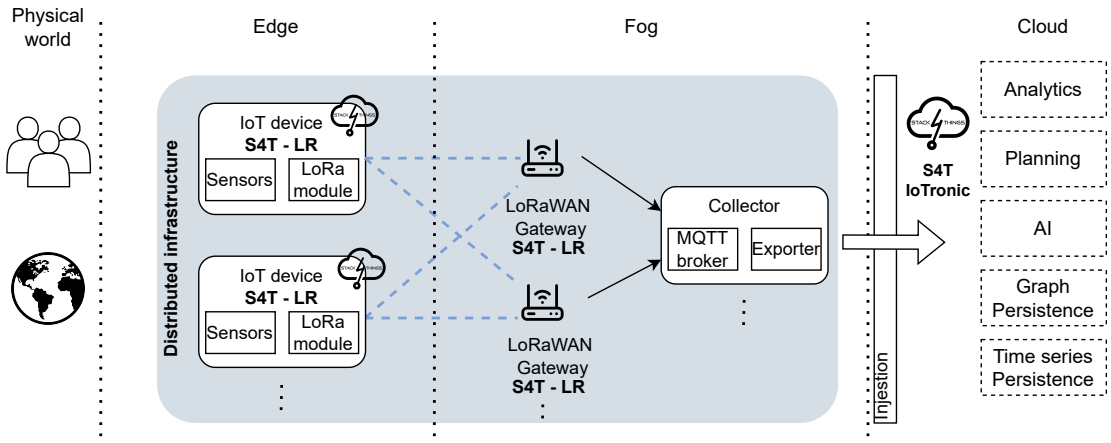

Figure 1: Reference architecture

3.2. Stack4Things

S4T is an evolution of the OpenStack environment [34] that, in addition to the typical functionalities for managing communication infrastructure, computing resources, and storage, also allows the management

of IoT objects (sensors and actuators). S4T, widely used and competitive for Infrastructure-as-a-Service like OpenStack, enables cloud-mediated interactions with fleets of nodes hosting sensors and actuators. Specifically, S4T provides a real-world scenario where the challenges of securing containerized environments—such as inter-service communication risks, runtime vulnerabilities, and orchestration misconfigurations—can be addressed in a controlled yet realistic setting.

By orchestrating IoT devices such as sensors and cameras as cloud resources, S4T bridges IoT ecosystems with traditional cloud infrastructures, addressing critical concerns related to secure communication, data integrity, and operational resilience.

S4T involves two main components: the IoTronic cloud-side service and the Lightning-rod (LR) device-side probe. IoTronic allows users to manage IoT resources. LR acts as the IoTronic counterpart, connecting the device to the cloud. Edge devices are equipped with advanced security measures, including secure boot, runtime anomaly detection, and encrypted data transmission, to mitigate vulnerabilities inherent in distributed environments. Together, these layers provide a unified and secure foundation for hybrid edge-cloud systems, automating security policy enforcement and ensuring compliance. S4T also enhances OpenStack's API capabilities, enabling the secure automation of IoT workflows and simplifying the onboarding of IoT devices through plug-and-play mechanisms, device authentication, and integrity verification to prevent unauthorized access or tampering.

The containerization of microservices is a cornerstone of S4T's approach to scalability, portability, and efficient resource management. Leveraging technologies like Docker or Kata Containers, microservices are encapsulated within containers that include all necessary code, libraries, and dependencies, ensuring consistent performance across diverse platforms. This modularity simplifies deployment, allowing individual components to be updated or replaced without disrupting the broader system. Therefore, S4T offers a robust solution for securing containerized microservices in hybrid edge-cloud systems, supporting critical applications in domains such as smart cities, industrial automation, and environmental monitoring. This comprehensive approach enables the reliable deployment of next-generation IoT-enabled cloud solutions while mitigating the risks associated with distributed infrastructures.

4. Deploying a LoRaWAN-based Edge-Cloud infrastructure: lesson learned

This section reports on the experience and lesson learned in deploying a LoRaWAN-based Edge-Cloud infrastructure for a smart city scenario, with a particular focus on security concerns revealed during the test of the network. The section outlines the experimental setup, methodology, and outcomes, demonstrating how such attack can compromise LoRaWAN network functionality.

4.1. Testbed

The testbed, used for the experiment, consists of two End Devices "Wi-Fi LoRa 32" boards designed by Heltec Automation, as shown in Fig. 2. Heltec provides libraries to send and receive data using the LoRaWAN protocol. One device operates as the trusted node (green in the figure), sending packets, while the other serves as the malicious device (red device), aiming to disrupt communication. The trusted node is configured as a Class A device using the Over-The-Air Activation (OTAA) procedure to join the network. The NS is set up with the corresponding DevEUI and AppKey to add the node device, enabling the test of various attack scenarios. The two devices are connected to two Raspberry Pi for programming and coordination.

The network utilized in the testbed is built around a Gateway device which implements both the LoRaWAN gateway and the Network Server. This device complies with the LoRaWAN 1.0 specifications. The Gateway connects to the NS using a wired network connection and communicates with the application server via the MQTT protocol.

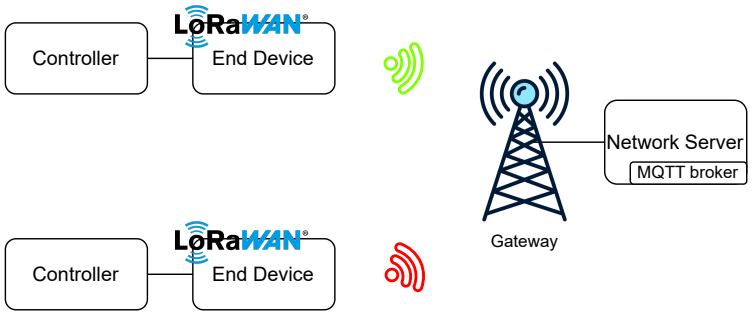

Figure 2: Testbed

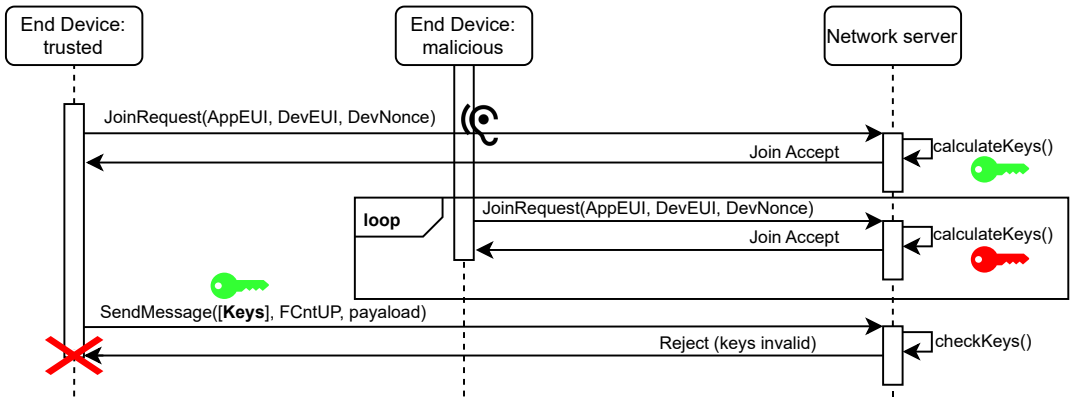

Figure 3: Sequence diagram of a Join Request replay attack

4.2. Join Request replay attack to a LoRaWAN gateway

The reuse of nonces during the OTAA join process poses a critical vulnerability in session key management and overall security. The nonce is a numerical value used to ensure the uniqueness of a communication or operation. If a nonce is reused or intercepted and retransmitted by an attacker, the NS may incorrectly accept it as a valid Join Request. As shown in the sequence diagram in Fig. 3, this results in the generation of new session keys, invalidating the old ones and causing desynchronization between the legitimate device and the server. The device, unaware of the key change, loses communication. Failure to detect and reject repeated nonces undermines network reliability and security, potentially enabling service disruptions or malicious infiltration. Effective monitoring of DevNonce and rejecting duplicates are essential to maintaining LoRaWAN security.

An experiment tested this vulnerability by replaying a previously logged Join Request from a malicious device. The NS incorrectly accepted the repeated request (as shown in Fig. 4), generating new session keys and disrupting the legitimate device's connection. Continued replay attacks effectively blocked the legitimate device from rejoining, creating a Denial of Service (DoS). This highlights a critical flaw in the server's handling of DevNonce reuse, indicating non-compliance with LoRaWAN protocol security requirements.

4.3. Addressing the issue in the described scenario

The Network Server must store and compare each received DevNonce, accepting only those that are new and greater than the previously recorded value.

Although LoRaWAN 1.0 already allows the use of random nonce management, it is important to

Device EUI/Group	Gateway ID	Frequency	Datarate	RSSI/SNR	Size	Fcnt	Type	Time
70B3D57ED00653C8	24E124FFFEF48414	868500000	SF9BW125	-/-	17	0	JnAcc	2024-10-06 20:26:15+02:00
70B3D57ED00653C8	24E124FFFEF48414	868500000	SF9BW125	-117/1.8	18	0	JnReq	2024-10-06 20:26:15+02:00
70B3D57ED00653C8	24E124FFFEF48414	868300000	SF9BW125	-/-	17	0	JnAcc	2024-10-06 20:25:42+02:00
70B3D57ED00653C8	24E124FFFEF48414	868300000	SF9BW125	-109/2.8	18	0	JnReq	2024-10-06 20:25:41+02:00
70B3D57ED00653C8	24E124FFFEF48414	868300000	SF9BW125	-/-	17	0	JnAcc	2024-10-06 20:25:25+02:00
70B3D57ED00653C8	24E124FFFEF48414	868300000	SF9BW125	-116/1.8	18	0	JnReq	2024-10-06 20:25:25+02:00
70B3D57ED00653C8	24E124FFFEF48414	868500000	SF9BW125	-/-	17	0	JnAcc	2024-10-06 20:24:51+02:00
70B3D57ED00653C8	24E124FFFEF48414	868500000	SF9BW125	-108/9.2	18	0	JnReq	2024-10-06 20:24:51+02:00

Figure 4: Repeated join observed by the gateway

check the correct implementation on the device. The server must reject requests from devices sending repeated nonces, thereby preventing the generation of new session keys for replayed packets. However, the protocol lacked stringent checks for ensuring the uniqueness of the DevNonce across multiple activations. Some servers might only track recent DevNonces or implement incomplete checks, leaving a gap for attackers to exploit by reusing older DevNonces. LoRaWAN 1.1 improved security by requiring the DevNonce to increment sequentially and introducing a 3-byte JoinNonce, generated by the NS, to prevent replay attacks. Moreover, the introduction of the Join Server decentralizes join operations, enhancing overall security.

From what has been illustrated above, it emerges the necessity of providing middleware solutions able to manage distributed IoT devices to monitor their activity and upgrade their software when behavior anomalies are detected. To this end, S4T can play an important role in conjunction with LoRaWAN to create a robust architecture for managing IoT devices and applications in low-power, long-range communication scenarios. S4T acts as a cloud-based framework for device provisioning, monitoring, and data aggregation, seamlessly connecting edge devices with cloud infrastructure. LoRaWAN connects decentralized IoT devices to gateways and the cloud. Together, these technologies enable scalable and application-focused IoT solutions, combining LoRaWAN efficient data transmission with S4T's centralized control and processing capabilities, creating an efficient and secure IoT ecosystem.

S4T provides robust security measures [35] to protect IoT systems, including advanced authentication, data encryption, access control, and real-time monitoring. These features ensure that only authorized entities access the system. It also handles aggregating logs, correlating events from multiple sources, and integrating dashboards like Kibana or Grafana for monitoring and alerting. The platform also supports periodic updates to security policies based on test results and provides the scalability needed to analyze and respond to threats across distributed environments, providing automated maintenance triggers. Additionally, S4T supports application-level encryption and anomaly detection, providing an extra layer of protection against attacks [36]. S4T strengthens authentication and enforces fine-grained access control by defining policies at the device, gateway, and application levels and supporting role-based access control (RBAC). These measures ensure that IoT deployments using LoRaWAN and S4T remain secure, scalable, and efficient.

5. Conclusion

The rapid integration of IoT technologies into smart cities has revolutionized urban management and service delivery, but it has also introduced a range of cybersecurity challenges. This paper has delved

into the vulnerabilities present in LoRaWAN-based IoT networks extensively used for implementing Edge-Cloud urban infrastructures. Issues such as data interception, unauthorized access, and denial-of-service attacks highlight the urgent need for robust and adaptive security measures. To address these challenges, we emphasize the necessity of leveraging frameworks like S4T. In large-scale, geographically distributed smart city environments, traditional security approaches often struggle with the complexity of monitoring and managing vast fleets of devices. S4T facilitates real-time analysis, enabling AI-driven detection agents to identify threats, trace vulnerabilities, and initiate automated patching and updates. By fostering secure and resilient infrastructures, we aim to support the growth of trustworthy IoT ecosystems, enabling smart cities to continue evolving as hubs of innovation and sustainability. Specifically, our investigation revealed how poor nonce management, a known security risk, has been improved in LoRaWAN 1.1 through better DevNonce handling, key separation, and the introduction of the Join Server. While LoRaWAN 1.0 already included replay attack prevention mechanisms, our analysis revealed that in practice, some gateways fail to correctly implement or enforce these protections, leaving networks vulnerable. The update and/or patch ensure that legitimate devices do not lose their connection due to DoS attacks and improve the overall security of LoRaWAN-based IoT networks. S4T's advanced logging and automated intervention capabilities enable detecting vulnerabilities, trigger targeted maintenance, and deploy necessary patches seamlessly across a vast network of devices, ensuring security to smart cities.

Future work will focus on developing a machine learning algorithm to predict failures by analysing parameters and detecting anomalies in attack patterns across distributed environments. This algorithm will be integrated into Stack4Things, enabling automated rejuvenation and enhancing resilience and security without manual intervention.

Acknowledgments

The European Union supported this work - Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUP D33C22001300002, partnership on "SEcurity and RIghts in the CyberSpace" (PE00000014 - program "SERICS")

Diclosure of Interests

The authors have no competing interests to declare that are relevant to the content of this article.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] M. Eldefrawy, I. Butun, N. Pereira, M. Gidlund, Formal security analysis of lorawan, Computer Networks 148 (2019) 328–339. URL: https://www.sciencedirect.com/science/article/pii/S1389128618306145. doi:https://doi.org/10.1016/j.comnet.2018.11.017.

[2] G. Tricomi, M. Giacobbe, I. Ficili, N. Peditto, A. Puliafito, Smart city as cooperating smart areas: On the way of symbiotic cyber–physical systems environment, Sensors 24 (2024). URL: https://www.mdpi.com/1424-8220/24/10/3108. doi:10.3390/s24103108.

[3] K. Ntshabele, B. Isong, N. Gasela, A. M. Abu-Mahfouz, A comprehensive analysis of lorawan key security models and possible attack solutions, Mathematics 10 (2022). URL: https://www.mdpi.com/2227-7390/10/19/3421. doi:10.3390/math10193421.

[4] Lora alliance, 2024. URL: https://lora-alliance.org/, last Accessed on December 11, 2024.

[5] N. Hayati, K. Ramli, M. Suryanegara, Y. Suryanto, Potential development of aes 128-bit key generation for lorawan security, in: 2019 2nd International Conference on Communication Engineering and Technology (ICCET), 2019, pp. 57–61. doi:10.1109/ICCET.2019.8726884.
[6] Iso/iec 27001 standard, 2024. URL: https://www.iso.org/standard/27001, last accessed on December 12, 2024.
[7] 2024.nist cybersecurity framework, 2024. URL: https://www.nist.gov/cyberframework, last accessed on December 11, 2024.
[8] Nistir 8259, 2024. URL: https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8259.pdf, last accessed on December 11, 2024.
[9] Etsi en 303 645 v3.1.3 (2024-09), 2024. URL: https://www.etsi.org/deliver/etsi_en/303600_303699/303645/03.01.03_60/en_303645v030103p.pdf, last accessed on December 11, 2024.
[10] Isa/iec 62443 series of standards, 2024. URL: https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards, last accessed on December 11, 2024.
[11] B. Huber, F. Kandah, Zero trust+: A trusted-based zero trust architecture for iot at scale, in: 2024 IEEE International Conference on Consumer Electronics (ICCE), IEEE, 2024, pp. 1–6.
[12] Y. Zheng, A. Pal, S. Abuadbba, S. R. Pokhrel, S. Nepal, H. Janicke, Towards iot security automation and orchestration, in: 2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), IEEE, 2020, pp. 55–63.
[13] Nist sp 800-61, 2024. URL: https://www.nist.gov/privacy-framework/nist-sp-800-61, last accessed on December 10, 2024.
[14] Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation) (text with eea relevance), 2024. URL: http://data.europa.eu/eli/reg/2016/679/oj, last accessed on December 10, 2024.
[15] Regulation (eu) 2019/881 of the european parliament and of the council of 17 april 2019 on enisa (the european union agency for cybersecurity) and on information and communications technology cybersecurity certification and repealing regulation (eu) no 526/2013 (cybersecurity act) (text with eea relevance), 2024. Last accessed on December 10, 2024.
[16] C. Colarusso, I. Falco, E. Zimeo, A greedy data-anchored placement of microservices in federated clouds, in: 2024 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), IEEE, 2024, pp. 103–110.
[17] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, 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 21 (2019) 2702–2733.
[18] J. Wurm, K. Hoang, O. Arias, A.-R. Sadeghi, Y. Jin, Security analysis on consumer and industrial iot devices, in: 2016 21st Asia and South Pacific design automation conference (ASP-DAC), IEEE, 2016, pp. 519–524.
[19] L. Mottola, A. Hameed, T. Voigt, Energy attacks in the battery-less internet of things: Directions for the future, in: Proceedings of the 17th European workshop on systems security, 2024, pp. 29–36.
[20] N. A. Khan, A. Awang, S. A. A. Karim, Security in internet of things: A review, IEEE access 10 (2022) 104649–104670.
[21] Y. Lee, W. Lee, G. Shin, K. Kim, Assessing the impact of dos attacks on iot gateway, in: Advanced Multimedia and Ubiquitous Engineering: MUE/FutureTech 2017 11, Springer, 2017, pp. 252–257.
[22] H. A. Noman, O. M. Abu-Sharkh, Code injection attacks in wireless-based internet of things (iot): A comprehensive review and practical implementations, Sensors 23 (2023) 6067.
[23] G. Nebbione, M. C. Calzarossa, Security of iot application layer protocols: Challenges and findings, Future Internet 12 (2020) 55.
[24] C. Colarusso, I. Falco, E. Zimeo, Towards business continuity with edge-cloud continuum, in: 2024 11th International Conference on Future Internet of Things and Cloud (FiCloud), IEEE, 2024, pp. 253–259.

[25] L. Lamport, R. Shostak, M. Pease, The byzantine generals problem, in: Concurrency: the works of leslie lamport, 2019, pp. 203–226.
[26] D. Wood, N. Apthorpe, N. Feamster, Cleartext data transmissions in consumer iot medical devices, in: Proceedings of the 2017 Workshop on Internet of Things Security and Privacy, 2017, pp. 7–12.
[27] A. Jahangeer, S. U. Bazai, S. Aslam, S. Marjan, M. Anas, S. H. Hashemi, A review on the security of iot networks: From network layer's perspective, IEEE Access 11 (2023) 71073–71087.
[28] R. Sahay, A. Nayyar, R. K. Shrivastava, M. Bilal, S. P. Singh, S. Pack, Routing attack induced anomaly detection in iot network using rbm-lstm, ICT Express 10 (2024) 459–464.
[29] N. Hou, X. Xia, Y. Zheng, Cloaklora: A covert channel over lora phy, IEEE/ACM Transactions on Networking 31 (2022) 1159–1172.
[30] F. Hessel, L. Almon, F. Álvarez, Chirpotle: A framework for practical lorawan security evaluation, in: Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks, 2020, pp. 306–316.
[31] F. L. Coman, K. M. Malarski, M. N. Petersen, S. Ruepp, Security issues in internet of things: Vulnerability analysis of lorawan, sigfox and nb-iot, in: 2019 Global IoT Summit (GIoTS), IEEE, 2019, pp. 1–6.
[32] K. P. Sah, N. Jain, P. Jha, J. Hawari, B. Beena, Advancing of microservices architecture with dockers, in: 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), IEEE, 2024, pp. 1–6.
[33] F. De Vita, D. Bruneo, Leveraging stack4things for federated learning in intelligent cyber physical systems, Journal of Sensor and Actuator Networks 9 (2020) 59.
[34] F. Longo, D. Bruneo, S. Distefano, G. Merlino, A. Puliafito, Stack4things: An openstack-based framework for iot, in: 2015 3rd International Conference on Future Internet of Things and Cloud, IEEE, 2015, pp. 204–211.
[35] G. Liu, B. Huang, Z. Liang, M. Qin, H. Zhou, Z. Li, Microservices: architecture, container, and challenges, in: 2020 IEEE 20th international conference on software quality, reliability and security companion (QRS-C), IEEE, 2020, pp. 629–635.
[36] Y. Li, H. Hu, S. Zhang, G. Cheng, W. Liu, An active security defense strategy for microservices based on deep reinforcement learning, in: 2023 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), IEEE, 2023, pp. 410–415.