Securing communication protocols for the Internet of Things

Ali Haj-Hassan

École Doctorale Polytechnique Hauts-de-France
Université Polytechnique Hauts-de-France
LAMIH UMR CNRS 8201
INSA Hauts-de-France
Université de Mons

Sécurisation de protocoles de communication pour l'Internet des objets

Thèse présentée et soutenue par **Ali HAJ-HASSAN**
Le **9 Janvier 2024** à **Mons**
En vue de l'obtention du grade de doctorat en **Informatique**

Composition du Jury:

Ramin Sadre	**Rapporteur**
Professeur, Université Catholique de Louvain	
Mawloud Omar	**Rapporteur**
Professeur, Université Bretagne Sud	
Tom Mens	**Président**
Professeur, Université de Mons	
An Braeken	**Examinateur**
Professeur, Vrije Universiteit Brussel	
Valeria Loscri	**Examinateur**
Chercheuse, Inria	
Antoine Gallais	**Directeur de thèse**
Professeur, INSA Hauts-de-France	
Bruno Quoitin	**Co-Directeur de thèse**
Professeur, Université de Mons	
Youcef Imine	**Encadrant**
Maître de Conférences, INSA Hauts-de-France	

I

Wings are a constraint that makes it possible to fly.

— Robert Bringhurst

To my parents...

Remerciements

Tout d'abord, je voudrais exprimer ma gratitude pour mon directeur de thèse Antoine Gallais, pour sa sagesse et sa bienveillance tout au long de ce parcours. Sa positivité et son soutien permanents ont été d'une aide précieuse.
Je remercie également mon co-directeur, Bruno Quoitin, pour sa précision, son insistance sur l'intégrité dans notre travail et son soutien constant. Sa gentillesse et son sourire ont apporté de la lumière à chaque étape de cette expérience.
Mon encadrant, Youcef Imine, mérite une reconnaissance spéciale pour sa direction très professionnelle, sa motivation constante et ses précieux conseils. Sa contribution a grandement enrichi mon expérience.
Je souhaite également remercier les membres du jury, An Braeken, Omar Mawloud, Ramin Sadre, Tom Mens et Valeria Loscri pour leurs précieuses remarques et contributions.
Je tiens à remercier sincèrement la Région Hauts-de-France et l'Université de Mons qui ont cofinancé cette thèse, et sans lesquels je n'aurais pas pu mener ces recherches.
Je tiens à remercier tout particulièrement mes collègues à Valenciennes, ainsi que ceux à Mons, pour avoir créé un environnement de travail exceptionnel et je les remercie pour leur soutien, leurs repas partagés et nos activités ensemble.
À mes amis proches, je tiens à exprimer toute ma gratitude pour leur soutien continu. Leur présence constante dans ma vie, leurs encouragements et leurs moments de convivialité ont été des piliers sur lesquels je me suis appuyé tout au long de ce voyage académique.
Enfin, le plus grand remerciement va à mes parents et à mes deux sœurs pour l'ensemble de leur soutien et de leurs sacrifices en ma faveur.

Lille, 6 Novembre 2023

Abstract

The fusion of IP-enabled networks with low-power wireless technology has given birth to the Industrial Internet of Things (IIoT). Due the large scale and dynamic nature of IIoT, securing such network is of paramount importance. One of the most critical attacks are those conducted during the joining phase of new nodes to an IIoT network. In this thesis, we focus our study on securing the joining phase of such networks.

Joining phases in IoT rely on mutual authentication methods based on a pre-shared key (PSK) shared between the network coordinator and the joining node. Standardization often lacks clear PSK sharing guidelines, which in large-scale and dynamic networks like IIoT makes pre-configuring each device with a unique key impractical. To address these concerns, this thesis introduces an autonomous mutual authentication and key establishment protocol for IIoT networks. In this solution, the network coordinator first authenticates the joining node via a certificate, and reciprocally, the joining node authenticates the network coordinator using a novel and lightweight consensus mechanism based on Shamir Secret Sharing. Once this mutual authentication is accomplished, a key is established between the network coordinator and the joining node over a public channel. Our solution was integrated into the 6TiSCH framework, ensuring robust security with high authentication success, even when dealing with malicious nodes. Additionally, it proved efficient in terms of communication, latency, and energy usage across various network scenarios, even on resource-constrained devices.

Moreover, during the IoT network joining process, proxy nodes play a pivotal role in forwarding Join Requests and Join Responses between the joining node and the network coordinator. Securing this phase is vital, as malicious proxy nodes can disturb new node joins or redirect them to another entity impersonating the coordinator. Therefore, we present a robust system

focused on identifying malicious proxy nodes during the joining phase. Centered around the coordinator, this system maintains a log table tracking each node's participation as a proxy node. After each joining phase, the coordinator receives an end-to-end encrypted packet from the joining node, detailing any encounters with malicious proxy nodes. This information is utilized to calculate the number of legitimate proxy node involvements for each node. The detection system utilizes these metrics, in conjunction with adjustable parameters, to categorize nodes as either malicious or trustworthy. Additionally, our solution accounts for potential attacks on the detection process, originating from both proxy nodes and joining nodes.

Résumé

La fusion des réseaux IP avec la technologie sans fil à faible consommation d'énergie a donné naissance à l'Internet Industriel des Objets (IIoT). En raison du large échelle et de la nature dynamique de l'IIoT, la sécurité de ce réseau est d'une importance capitale. L'une des attaques les plus critiques concerne celles menées lors de la phase d'intégration de nouveaux nœuds dans un réseau IIoT. Dans cette thèse, nous concentrons notre étude sur la sécurisation de la phase d'intégration de ces réseaux.

Les phases d'intégration dans l'IoT reposent sur des méthodes d'authentification mutuelle basées sur une clé prépartagée (PSK) partagée entre le coordinateur du réseau et le nœud d'intégration. La standarization manque souvent de clarifications sur le partage de PSK, ce qui rend impraticable la préconfiguration de chaque appareil avec une clé unique dans les réseaux à grande échelle et dynamiques tels que l'IIoT. Pour répondre à ces problématiques, cette thèse présente un protocole d'authentification mutuelle autonome et d'établissement de clés pour les réseaux IIoT. Dans cette solution, le coordinateur du réseau authentifie d'abord le nœud d'intégration via un certificat, et réciproquement, le nœud d'intégration authentifie le coordinateur du réseau en utilisant un mécanisme de consensus léger basé sur le partage de secret de Shamir. Une fois cette authentification mutuelle accomplie, une clé est établie entre le coordinateur du réseau et le nouveau noeud sur un canal public. Notre solution a été intégrée dans le cadre du protocole 6TiSCH, garantissant une sécurité robuste avec un taux d'authentification élevé, même en présence de nœuds malveillants. De plus, elle s'est prouvé efficace en termes de communication, de latence et de consommation d'énergie dans divers scénarios réseau, y compris sur des appareils aux ressources limitées.

De plus, lors du processus d'intégration du réseau IoT, les nœuds proxy jouent un rôle essentiel

en transférant les demandes d'intégration et les réponses entre le nœud d'intégration et le coordinateur du réseau. Sécuriser cette phase est essentielle, car les nœuds proxy malveillants peuvent perturber l'intégration de nouveaux nœuds ou les rediriger vers une autre entité se faisant passer pour le coordinateur. Par conséquent, nous présentons un système robuste axé sur l'identification de nœuds proxy malveillants lors de la phase d'intégration. Ce système, centré autour du coordinateur, tient un registre des participations de chaque nœud en tant que nœud proxy. Après chaque phase d'intégration, le coordinateur reçoit un paquet chiffré de bout en bout du nœud d'intégration, détaillant les rencontres avec des nœuds proxy malveillants. Ces informations sont utilisées pour calculer le nombre de participations légitimes de nœuds proxy pour chaque nœud. Le système de détection utilise ces métriques, en conjonction avec des paramètres ajustables, pour catégoriser les nœuds comme malveillants ou dignes de confiance. De plus, notre solution prend en compte les attaques potentielles sur le processus de détection, émanant à la fois des nœuds proxy et des nœuds d'intégration.

Contents

List of Figures

List of Tables

Publications

Journal articles:

- **Detecting Malicious Proxy Nodes during IoT Network Joining Phase**
  Ali Haj-Hassan, Youcef Imine, Antoine Gallais, Bruno Quoitin
  Computer Networks 2024
  https://doi.org/10.1016/j.comnet.2024.110308

- **Consensus-Based Mutual Authentication Scheme for Industrial IoT**
  Ali Haj-Hassan, Youcef Imine, Antoine Gallais, Bruno Quoitin
  Ad Hoc Networks 2023
  https://doi.org/10.1016/j.adhoc.2023.103162

Conference papers:

- **Zero-Touch Mutual Authentication Scheme for 6TiSCH Industrial IoT Networks**
  Ali Haj-Hassan, Youcef Imine, Antoine Gallais, Bruno Quoitin
  2022 International Wireless Communications and Mobile Computing Conference (IWCMC), Dubrovnik, Croitia
  https://doi.org/10.1109/IWCMC55113.2022.9824568

1 Introduction

1.1 Context

The Internet of Things (IoT) and its industrial counterpart, the Industrial Internet of Things (IIoT), are playing an innovative role in today's world. IoT is a transformative concept that connects a wide array of everyday objects and devices to the Internet, enabling them to collect, share, and process data autonomously [Cen+21]. IoT systems are networks that include sensors, actuators, and other devices, which communicate with each other and centralized servers through the internet. This interconnected system is at the heart of many modern innovations and applications. IoT comprises various subdomains, each customized for specific applications such as: the Industrial Internet of Things (IIoT) [Boy+18], Internet of Vehicles (IoV) [SS17], Smart Cities [Soo+18], Wearable Technology [Ome+21], Agriculture Technology (AgriTech) [Yan+21], Healthcare and Remote Monitoring [Zuh+17], Smart Homes [Ala+17a], Environmental Monitoring [US20], Retail and Supply Chain [Son+20], Energy Management [KRR15], etc. These areas demonstrate how versatile and impactful IoT technology is in different fields, making it a key factor in the current digital revolution. [Sal+18].

In contrast, the Industrial Internet of Things (IIoT) is a transformative integration of traditional industries with advanced digital technology, notably exemplified by the Industry 4.0 movement [Xu+18] [Sis+18]. This novel approach is centered on the interconnection of physical devices, such as sensors, machinery, and SCADA (Supervisory Control and Data Acquisition) systems, with the digital world through the internet. In this interconnected framework, these

devices no longer operate in isolation; instead, they function as essential elements within wide networks. This network environment enables the devices to actively share data, communicate, and even autonomously make decisions based on data analysis. The implementation of this concept heavily depends on key technologies, including wireless communication technologies, cloud and edge computing, which collectively empower this industrial context with real-time insights into their operations. This technology aims to improve operational efficiency, optimize production and reduce costs [Lu17].
The application of IIoT expands its influence to a wide range of industries, including manufacturing, energy, agriculture, and healthcare. In manufacturing, it brings about optimization of production lines and elevates the quality control processes, which results in cost-saving measures. The energy sector benefits from efficient monitoring and management of energy grids through IIoT. In agriculture, precision farming techniques are empowered, and in healthcare, the system aids in remote patient monitoring. These versatile applications eventually result in increased efficiency, lowered operational costs, and an overall improvement in quality across different industrial sectors.

What makes IIoT particularly challenging and exciting is the convergence of Information Technology (IT) and Operational Technology (OT) [EC20]. IT traditionally deals with digital data and enterprise-level systems, while OT is concerned with the physical processes in industrial environments. Thanks to bridging the IP-enabled networks and low power wireless networks, IoT devices are not only able to communicate between themselves but also with remote IoT devices through IP networks. This fusion presents remarkable possibilities for monitoring and control, but it also opens doors to new vulnerabilities. Enabling IP connectivity put low power wireless networks on a larger surface of risk. As we dive into this digital revolution, security becomes a major concern. The extensive scale and complex interconnections of IIoT and Cyber-Physical Systems (CPS) make them vulnerable to various threats, including cyber-attacks. These threats can lead to data breaches, espionage, and, in extreme cases, physical damage, operational disruptions, and even risks to human safety [Hum+17]. In the world of IIoT, we often encounter resource-constrained devices and environments where lightweight and efficient protocols are essential [Imt+21]. To make the most of the potential of these technologies while ensuring their safety, we need strong security measures. This in-

cludes robust authentication methods, encryption, and intrusion detection systems [Ala+17b]. Additionally, clear standards and regulations are crucial, especially when IIoT extends into critical areas like healthcare, energy, and transportation [Wan+21].

One of the most critical attacks are those conducted during the joining phase of new nodes to an IoT network. They have a high impact on the later security level of the network [Lag+21; Naz+21]. A joining protocol defines how new nodes join a network. In this phase, a new joining node exchanges join messages with a network coordinator, responsible of managing the network, and giving access to new nodes. The initial contact between the joining node and the network is established through an intermediate node, known as a proxy node, responsible for broadcasting the network beacons and receiving Join Requests [Bou+20]. If the proxy node were to engage in malicious behaviors, such as a selective forwarding attack by discarding requests, it could disrupt the joining process and compromise network integrity. Moreover, since a new joining node has no previous knowledge on the network, it has no way to verify the correctness of information during this joining phase. Therefore, malicious proxy nodes may forward the node's join requests to another malicious node pretending to be the network coordinator, instead of forwarding it to the true one. Additionally, in the absence of security measures during the joining phase, malicious nodes can join the network and disturb forthcoming joining events. In that case the network's coordinator would not have any knowledge about the Join Requests handled by that join proxy. In the realm of IoT security, addressing this threat is of paramount importance. The large scale and dynamic characteristic of IIoT make it an intriguing area of study. Existing solutions designed for traditional IoT might not be well-suited for these complex systems.

In essence, IoT and IIoT are changing the way we interact with the world and how industries operate. To ensure these changes bring benefits and not risks, we must make security a top priority. The advantages are enormous, but so are the challenges. By addressing these challenges, we can create a secure and connected future.

1.2 Contribution of the Thesis

1.2.1 Consensus-based mutual authentication

To address the security concerns during IIoT's joining phases, it is crucial to implement a mutual authentication. This mutual authentication accomplishes two key objectives allowing: (1) both the network coordinator and the new joining node to verify each other's identities and (2) to establish a common key to secure forthcoming exchanges between the node and the coordinator.
As mentioned earlier, in the context of large scale and dynamic IIoT network, ensuring the safety of the joining phase is not simple. The authentication mechanism must be autonomous enough in order to manage network access authorization, without needing to configure each device individually. Furthermore, considering the presence of resource-constrained devices, the mechanism must strive to be as lightweight as possible.

It is in this context that we propose in this thesis a novel zero-touch mutual authentication protocol for IoT networks where the concept of pre-configuring each node before the joining phase is a real challenge. On one hand, our solution is based on certificates to allow the network coordinator to authenticate new joining nodes. On the other hand, we propose a new consensus approach among network members, based on Shamir secret sharing, allowing new joining nodes to authenticate and establish keys with the coordinator. In order to make our solution more concrete, we adopted the 6TiSCH protocol [Vil+19] as an application scenario to integrate our solution with. Introduced by the IETF, it is a promising IIoT solution that enables high reliability IPv6 wireless sensor networks. It is based on the IEEE 802.15.4 Time Slotted Channel Hopping (TSCH) medium access control which combines time division multiplexing, to make access deterministic, and frequency hopping, for increased robustness against interferences. By integrating our solution with the industrial protocol 6TiSCH, and adapting it to large scale and dynamic IoT networks, it became more oriented towards IIoT. However, it can be integrated with different IoT applications, providing the following features:

- Mutual authentication without a pre-configured key.
- Low memory requirement demonstrated by its implementation on constrained IoT

devices.

- Easy integration with communication protocols adopted in IoT networks, demonstrated by our application to the joining phase of 6TiSCH.

1.2.2 Malicious Proxy nodes detection

As described in the previous sections, a joining phase is based on a contact between a joining node and multiple proxy nodes of the network. Without security measures during the joining phase, malicious proxy nodes may disrupt the process, forwarding requests to other malicious nodes impersonating the coordinator, potentially allowing malicious nodes to join and compromise network integrity. To tackle these challenges, it is crucial to implement a detection system for identifying malicious proxy nodes. This system is essential since it allows: (1) the network coordinator to be aware of the behaviour of proxy nodes handling Join Requests, (2) detect malicious ones and prevent them from disturbing the forthcoming joining phases. However, the joining phase relies usually on simple and lightweight mechanisms, thus leading to a lack of elements that a detection system can use to determine whether a proxy node is behaving maliciously or not.

In this context, we propose in this thesis a new approach for detecting malicious Proxy nodes participating in a zero-touch authentication during the network joining phase. The network coordinator in our system keeps record of how often each node acts as a Proxy node. After every joining phase, it receives an encrypted packet from the joining node, containing details about the Proxy nodes that may have behaved maliciously during the join process. Using this information, the system assesses the nodes' behaviour and classify them as either malicious or trustworthy. In order to prove the applicability of our proposal, we adopt the 6TiSCH protocol as an application scenario. We also consider our previously proposed consensus-based mutual authentication scheme [Haj+23] as the operating authentication protocol securing the joining phase. Additionally, an extensive performance evaluation is conducted to prove the efficiency of our protocol even in the case where various attacks target its detection process. We take into consideration the following attacks, that may be originated from both Proxy nodes and new joining nodes:

- ON-OFF conducted by a Proxy node in order to falsify the detection mechanism.
- False positive conducted by a joining node in order to exclude an honest Proxy node.
- False negative attack conducted by a joining node in order to avoid the detection of a malicious Proxy node.

1.3 Structure of the Thesis

The rest of this thesis is organized as follows. In chapter 2 we present the communication protocols in IoT and shed some light on 6TiSCH protocol. After that, we address the security aspects in IoT and we give an overview of the main contributions around it.
In chapter 3, we present our consensus-based mutual authentication solution for IIoT. First, we explain the solution in detail. Then, we provide an outline of the possible attack models and an assessment of their impact on security. Finally, we discuss the limitations of this solution.
An application of this solution on 6TiSCH and a detailed evaluation are presented in chapter 4.
In chapter 5, we present our solution to detect malicious proxy nodes in the joining phase of an IoT network. Then, we give alongside a deep security analysis taking into consideration several type of attacks.
An in-depth evaluation of this solution is presented in chapter 6. The results are discussed and a discussion about the solution is provided.
Finally, we conclude the thesis in chapter 7, and we give perspectives for our future work.

2 State of the Art

In this chapter, we start by representing the most known communication protocols and their features. We describe the industrial protocol 6TiSCH and we represent CoJP, its joining phase. Then, we delve into the security issues and challenges encountered within IoT and provide a review of authentication solutions and malicious node detection systems.

2.1 Prerequisites

2.1.1 Communication protocols

Communication protocols play a vital role in the functioning of the Internet of Things. They govern how devices share information, aiming for uninterrupted connectivity in this vast network of smart objects. These protocols are classified based on attributes such as network architecture, topology, power consumption, data rate, and range. Each protocol is suited to different IoT applications, from smart homes to industrial systems. In this section, we'll explore some of the leading communication protocols used in IoT, examining their individual strengths and applications [Ger+23].

- **IEEE 802.15.4**: IEEE 802.15.4 is a standard that defines the physical and media access control (MAC) layers for wireless communication over short range, emphasizing low data rates and power-efficient operations. It serves as the foundation for several network protocols, including Zigbee, Thread, WirelessHART, 6LoWPAN, and 6TiSCH. These

protocols are designed for various applications, such as home automation, industrial control [TAA+23].

- **Zigbee**: Zigbee is a wireless communication system created for short-range, low-power applications. It finds common use in areas such as home automation, industrial control systems, and various Internet of Things (IoT) devices. Zigbee is built on top of IEEE 802.15.4 standard, which sets the rules for the physical and data link layers of low-rate wireless personal area networks (LR-WPANs) [Zoh+23].

- **LoRaWAN**: LoRaWAN, short for "Long Range Wide Area Network," is a prominent Low-Power Wide Area Network (LPWAN) protocol. It is designed for long-distance, low-power communication in the Internet of Things (IoT) and machine-to-machine (M2M) applications. LoRaWAN leverages LoRa technology, known for its extended wireless range and energy efficiency, allowing IoT devices to transmit data across vast distances while conserving battery life. This technology enables the creation of large-scale IoT networks, offering a secure and efficient means for numerous devices to communicate with central gateways or servers. This adaptability makes LoRaWAN an excellent choice for a diverse array of IoT applications, including those in smart cities, asset tracking, and industrial monitoring, all within the framework of LPWANs that prioritize low power consumption and long-range communication [Jou+23].

- **Sigfox**: Sigfox is an LPWAN (Low-Power Wide Area Network) technology tailored for cost-effective, energy-efficient, and long-distance communication in the Internet of Things (IoT). It provides a dependable and efficient way for various IoT devices to transmit small data over extensive distances. Sigfox connects these devices to a global LPWAN network, making it an ideal choice for applications like environmental monitoring. It excels in establishing extensive IoT networks while minimizing power consumption and costs [Nae+23].

- **NB-IoT**: NB-IoT (Narrowband Internet of Things) stands as a Low-Power Wide Area Network (LPWAN) technology exclusively designed for the Internet of Things (IoT). It offers a dependable, energy-efficient, and cost-conscious method for IoT devices to transmit small data packets across substantial distances. NB-IoT integrates seamlessly

within cellular networks, rendering it suitable for a variety of applications such as smart metering. It provides a robust solution for connecting an array of IoT devices while preserving power resources and reducing operational expenses [Pra+23].

2.1.2 6TiSCH

In this thesis, we adopted 6TiSCH as our application scenario and we integrated our solutions into it. Therefore, in the following sections, we offer a detailed description of this protocol and its joining phase, CoJP.

Framework

6TiSCH [Vil+19], which stands for *IPv6 over the TSCH mode of IEEE 802.15.4e*, is a standard network stack developed by the IETF (The Internet Engineering Task Force, a standards organization for the Internet). It brings the low power industrial IEEE 802.15.4 physical layer to the IPv6 internet, providing a solid infrastructure for industrial applications by allowing constrained devices to be connected to remote IoT networks. 6TiSCH is composed of multiple layers, as illustrated in Figure 2.1.

The physical and data-link layers of the 6TiSCH stack are those defined by IEEE 802.15.4, operating in the *Time Slotted Channel Hopping* (TSCH) mode [Vog+18]. It combines time division multiplexing with frequency agility to offer deterministic, low-latency and robust medium access. The 6TiSCH Operation (6top) sub-layer defines the protocol and operations required for distributed scheduling at the MAC layer. At the network layer, IPv6 is used with 6LoWPAN for forwarding packets over IEEE 802.15.4 frames. The RPL protocol is used for routing [AAJ20]. It makes use of special ICMPv6 control messages such as the *DODAG Information Object* (DIO) to build up in a distributed manner a tree-like routing topology named a DODAG that spans the entire network. The DODAG can be rooted in one or more nodes. The non-storing mode of RPL is adopted by default in 6TiSCH. In this mode, only the root is aware of the full routing table while the other nodes only maintain their parents list in the tree. Hence, the root node is responsible for forwarding the packets between two nodes. Finally, the *Constrained Application Protocol* (CoAP) [RHM19] offers to constrained nodes a

data transfer service similar to HTTP. CoAP ensures transfer reliability on its own and relies on UDP instead of TCP.

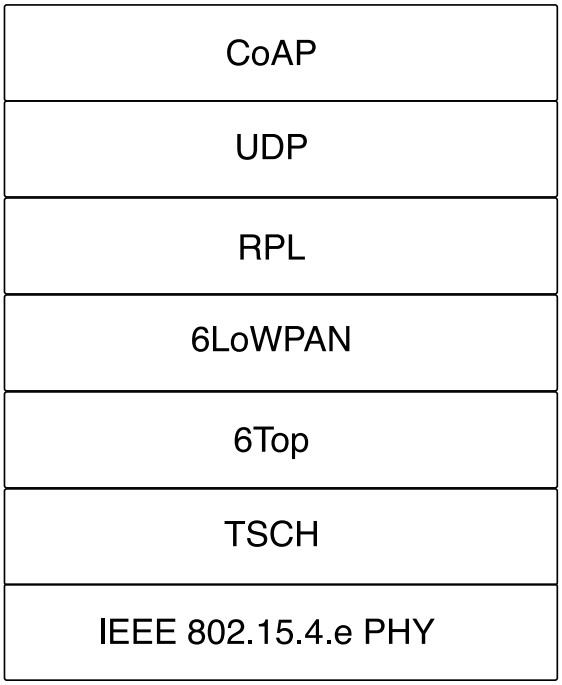


Figure 2.1: 6TiSCH protocol stack.

6TiSCH Joining Phase (CoJP)

In a 6TiSCH network, new nodes need to go through an initial joining phase to get admitted by the network coordinator and obtain link-level security credentials. The default authentication scheme proposed to join a 6TiSCH network is the Constrained Join Protocol (CoJP) [Vuč+21a]. It handles the parameter distribution needed for new nodes to join a 6TiSCH network. It operates at the application layer on top of CoAP, using a security protocol called OSCORE [Sel+19a; Sel+19b] which offers end-to-end protection by relying on symmetric encryption using per-device pre-shared keys.

In CoJP, three entities play a main role during the joining phase. (1) The Join Registrar/Coordinator (JRC) is the entity responsible for managing the network and giving access to new nodes. (2) The pledge is the node seeking to join the network. (3) The Join Proxy (JP) is a node already in the network that plays an intermediary in the exchanges between a pledge and the JRC.

During this phase different types of messages are exchanged:

- Enhanced Beacon (EB) [SPK+21]: message sent on a regular basis by JPs allowing pledges to discover the network and inviting them to join;
- Join Request: message sent from the pledge to the JRC, through the JP. It includes the role it requests to play in the network, as well as the identifier of the network it requests to join.
- Join Response: message sent from the JRC to the pledge through the JP. It contains different parameters needed by the pledge to become a fully operational network node.

2.2 IoT Security

2.2.1 Requirements

In the world of IoT, where devices constantly exchange data, three essential principles form the foundation of security: confidentiality, availability, and integrity. Confidentiality ensures that personal data, such as health records from wearable health devices, remains private and immune to unauthorized access. Availability guarantees that critical services, much like smart traffic management systems that keep city traffic flowing, are consistently accessible. Meanwhile, integrity ensures the reliability of data, like maintaining the accuracy of environmental measurements gathered by IoT sensors for informed decision-making. In the IoT context, these principles are not abstract concepts; they are the safeguarding elements that protect our privacy, keep essential services running smoothly, and maintain the trustworthiness of data used in real-world applications. Embracing these principles is the key to unlocking the potential of IoT while securing a world of trust, reliability, and functionality in our interconnected landscape [Tsi+21] [Nes+19] [Has+19b].

2.2.2 Issues

While confidentiality, availability, and integrity serve as the foundation for securing IoT, the digital landscape of interconnected devices is not without its challenges. In this section, we delve into the core security issues that affect IoT systems.

- **Authentication issues:** attackers can exploit ineffective authentication approaches, potentially leading to unauthorized access. This unauthorized access can take the form of impersonation attacks, where attackers may impersonate legitimate users or devices to gain access to IoT systems. The consequences of non-authorized access include appending malicious nodes, compromising data integrity, and ultimately intruding on IoT devices and network communications. Furthermore, the authentication keys exchanged and employed within the IoT network are at risk of being lost, destroyed, or corrupted, particularly when not securely stored or transmitted. In such cases, the effectiveness of authentication algorithms becomes insufficient, leaving the IoT netwok exposed to the risk of non-authorized access and impersonation attack. An example of this is when attackers steal or guess authentication credentials, gaining unauthorized entry into an IoT system, such as a smart home, which could result in unauthorized control and privacy invasion [HQS19].

- **Privacy issues:** The lack of privacy in IoT exposes individuals to various risks and potential attacks. In scenarios such as smart homes, attackers can intercept communication between devices, leading to privacy breaches, data leaks, and unauthorized access. For instance, eavesdropping on device communication may reveal users' routines and activities, compromising their privacy. In digital healthcare, the exposure of medical records and sensitive health data can lead to identity theft or unauthorized access, posing significant privacy threats. In the case of critical infrastructure, like power plants or water treatment facilities, breaches in privacy and security mechanisms can provide attackers with valuable knowledge about the infrastructure's vulnerabilities, enabling them to plan and execute sophisticated attacks that compromise safety and reliability. Thus, inadequate privacy safeguards may result in various privacy-related attacks, including eavesdropping, impersonation, and data breaches, undermining the confidentiality and security of IoT data [Ala+17b] .

- **Access Control issues:** Weak access controls create an open door for a variety of threats. Unauthorized access can result in data breaches, jeopardizing sensitive information and individual privacy. Additionally, inadequate access controls can lead to identity theft, enabling malicious actors to impersonate legitimate users or devices. These issues

can also facilitate the spread of malware within the network, introducing destructive elements that can disrupt operations, damage critical infrastructure, and lead to substantial financial losses. Furthermore, unauthorized access can provide a gateway for attackers to manipulate connected IoT devices, compromising system integrity. In industrial scenarios, the risks manifest in unauthorized control over manufacturing machinery or interference with essential systems like power grids [Con+18] .

- **Integrity issues:** Data protection in IoT and critical Cyber-Physical Systems (CPS) is of utmost significance, and encryption stands as a pivotal tool to safeguard data storage and transmission, ensuring that only authorized users can access and utilize the information. However, the resource constraints in IoT environments, affecting algorithm robustness and efficiency, can lead to vulnerabilities in encryption mechanisms. Attackers may exploit these limitations to compromise sensitive data or manipulate operations with relative ease. For example, as IoT devices multiply, the vast amount of generated data often necessitates cloud-based processing, which can introduce privacy concerns due to data traveling through multiple network hops. In such scenarios, a robust encryption mechanism becomes vital to maintain data confidentiality. Nonetheless, the vulnerability of IoT devices to attacks poses a risk to data integrity, with potential attackers resorting to techniques like Man-in-the-Middle (MitM) attacks. MitM attacks, which enable attackers to intercept and modify communications between IoT devices, can lead to the compromise of critical information, posing significant security threats [KS18].

- **Availability issues:** Ensuring the availability of IoT systems is paramount for their reliability, but it comes with several risks. These include Distributed Denial of Service (DDoS) attacks, network congestion, system failures, and jamming attacks, all of which can disrupt IoT operations. DDoS attacks flood IoT networks with excessive traffic, rendering services inaccessible. Network congestion occurs when IoT devices generate an overload of data, leading to slowdowns or outages, especially during peak times. System failures, whether from hardware issues or software glitches, can compromise IoT services, with severe consequences in critical applications. Furthermore, single points of failure, where one compromised device can disrupt the entire system, pose

a significant risk. Jamming attacks target IoT communication, causing wireless signal disruptions and rendering services unavailable. Robust infrastructure, redundancy, proactive monitoring, and strong security measures are essential for ensuring IoT availability. A single malfunctioning IIoT device in a factory can stop an entire production line, causing significant downtime and financial losses. Additionally, a jamming attack targeting wireless sensors can disrupt communication between machines, impacting manufacturing efficiency [KJ23].

2.2.3 Challenges

To address the multifaceted security issues in Industrial IoT (IIoT), the implementation of comprehensive security services is imperative. These services encompass authentication, access control, data privacy, integrity, availability, and trust management. However, several challenges need to be overcome to effectively deploy these services in IIoT environments. These challenges include:

- **Heterogeneity:** Heterogeneity within Industrial IoT (IIoT) involves diverse devices, communication protocols, and data formats, leading to interoperability challenges and security vulnerabilities. IIoT systems often comprise devices from various manufacturers and generations, posing difficulties in implementing uniform security measures. As a result, the security approach for IIoT needs to be adaptable and comprehensive to effectively address this diversity. Additionally, striking a balance between energy efficiency and safety remains a significant concern in this heterogeneous IIoT environment.

- **Resource-Constrained devices:** In the domain of industrial IoT (IIoT), energy efficiency is a crucial consideration, given the constraints on power resources, particularly in remote and inaccessible locations. Ensuring strong security while conserving energy is a significant challenge. To overcome this challenge, it's essential to create and implement energy-efficient security protocols and lightweight solutions that consider communication, computation, cryptography cost, and more. These measures enable IIoT devices to function reliably for extended periods without frequent battery replacements or recharging, contributing to the sustainability of interconnected systems.

- **Large scale deployments:** Securing extensive IIoT networks with dynamic, interconnected devices is complex. IoT networks require resource-intensive security management, involving cryptographic keys, forward secrecy, backward secrecy, authentication, and access control. Managing unique keys for multiple devices is impractical and resource-demanding. It includes secure key distribution and quick revocation of compromised or outdated keys, vital for system integrity and ensuring both forward and backward secrecy. Efficient key management across large industrial networks is crucial, demanding robust mechanisms that also account for scalability and operational efficiency, which make their authentication mechanisms even more challenging.

- **Real-Time requirements:** In industrial settings, real-time security is vital for timely decisions and efficient operations. However, strong security shouldn't cause significant delays that disrupt critical processes. This challenge calls for security solutions that can seamlessly work within tight time constraints, especially in manufacturing and autonomous systems where split-second decisions are crucial. Security protocols must reduce delays in data transmission and authentication while effectively protecting IIoT systems. Balancing real-time data processing with robust security demands innovative approaches in security protocol design and optimization.

- **Physical world integration:** In the context of Industrial IoT (IIoT), a significant security concern arises from the presence of unattended and autonomous devices. Many IIoT devices work independently in environments without continuous human supervision, making them vulnerable to unauthorized physical access. This vulnerability creates an appealing opportunity for potential adversaries aiming to take control of these devices, leading to various security risks. One particular aspect of this challenge is the compromise of nodes. In this scenario, malicious actors exploit their physical proximity to IIoT devices to compromise their integrity. This can lead to severe consequences, including physical damage to devices or the theft of critical data, such as cryptographic schemes and firmware. Attackers may also employ malicious nodes to replicate firmware or undermine the integrity of control and cyber data.

Addressing these challenges comprehensively is vital to ensure the security and safety of IoT

systems [Sha+18]. As previously mentioned, this thesis centers on enhancing the security of IoT networks within large-scale dynamic environments. Our primary focus encompasses critical aspects such as authentication, authorization, and network security in IoT. Furthermore, our solution includes a detection system that addresses malicious nodes detection in the IoT framework within the same context.

2.3 Authentication in IoT

Authentication methods in the field of IoT have garnered significant attention, leading to numerous proposed solutions [Kum+22] [El-+19] [AZ20] [Ash+23]. Various factors can be considered for mutually authenticating two entities, including pre-shared keys, certificates, physical unclonable functions, and more [El-+19; Yan+17; Hus+22; AZ20], [Mam+21; Kha+22; CAS21]. Additionally, innovative authentication schemes in heterogeneous networks may leverage multi-factor authentication. In what follows, we discuss each of these methods separately, highlighting their primary advantages and limitations.

2.3.1 Pre-Shared Key (PSK) Authentication

In most IoT wireless networks, authentication schemes are based on pre-configuring a new joining node with a pre-shared key (PSK), before the first phase of authentication. PSK operate by requiring both the device and the network coordinator to share a secret key before initiating communication. For instance, each device has a unique identifier and needs to pre-share a symmetric key with the network coordinator, which can further authenticate known devices upon new communication [Jan+14]. PSK methods offer a straightforward approach to authentication, as they rely on a shared secret for validation. This simplicity can be advantageous, particularly for resource-constrained IoT devices. However, a key challenge lies in securely distributing and managing these shared secrets, as provisioning every device with a unique key can be impractical in large-scale IoT networks. The shared PSK approach simplifies authentication but raises concerns about key management and security at scale, making it essential to explore more scalable and robust methods. In 6TiSCH protocol, the authors of the minimal security draft of the IETF [Vuč+21a] assume that the exchanges during

the joining phase of a new node is secured using a PSK. However, they do not describe how this key was shared. The same assumption is made in [SV18], as the authors proposed a 3-way mutual authentication mechanism based on a multi-key called secure vaults. Here, the PSK consists of the secure vault shared with the IoT devices during the deployment phase. This secure vault will be used by the coordinator to challenge a device and authenticate it. Other PSK-based authentication schemes were proposed in order to authenticate a joining node, for wireless network protocols like 6LoWPAN and LoRaWAN [San+18; Hus+13; Esf+17; ATW19; Cui+23; Min+22]. All these solutions require pre-configuring the IoT devices before initiating this phase, which is not efficient for the case of a large scale dynamic industrial network.

2.3.2 Certificate-based Authentication

Certificate-based authentication in the context of IoT involves using digital certificates issued by trusted Certificate Authorities (CAs) [Hus+22]. Each device or entity within the network has a digital certificate that contains identity details and a public key. When a device wants to communicate, it shares its digital certificate with the recipient, accompanied by a digital signature for ensuring data integrity. The recipient validates the certificate by verifying the digital signature and checking its authenticity through a trusted CA.
In [Por+14], the authors proposed a two-phase mutual authentication solution. First, it runs a registration phase where each edge device acquires its security credentials from a Certificate Authority (CA) and stores its chain of trust. Then, an authentication phase is executed where devices establish a secure communication channel. The authors in [KPB19] utilized Public Key Infrastructure (PKI) along with X.509 digital certificates to enhance device authentication, particularly within the context of embedded systems in the IoT. These certificates served a dual purpose, enabling both device identification and ensuring the integrity of the embedded systems involved in the IoT ecosystem. The X.509 digital certificates played a crucial role in securing IoT by providing a robust foundation for device authentication and integrity verification within embedded systems. However, these solution do not address scenarios with multiple CAs, which would require constrained devices to store a considerable number of chains of trust. Moreover, even though some solutions allow to assess the status of a given certificate (e.g. OCSP), the consistency of revocation lists may be endangered once large scale

and lossy networks are considered, such as the heterogeneous IoT networks targeted here.

2.3.3 Physical Unclonable Function (PUF) Authentication

Physical Unclonable Functions (PUFs) are like the unique fingerprints of electronic devices, created due to tiny variations in their hardware components [Bra18]. In the context of IoT authentication, PUF-based methods take advantage of these distinctive hardware-based fingerprints. These fingerprints are typically embedded in a device's hardware during its manufacturing process, and they serve as a means to verify the authenticity of the device. When authentication is required, a PUF challenge is issued to the device, which generates a response based on its unique hardware characteristics. This response is used to confirm the device's identity and grant access if it matches the expected response. Each IoT device's response to PUF challenges is inherently different, making it both a one-of-a-kind identifier and a cryptographic key. This makes PUF-based authentication a robust choice for securing IoT devices, as it relies on the inherent uniqueness of each device's hardware characteristics. In [MNC20], the authors propose a multi-factor mutual authentication scheme between IoT devices and servers in a star topology. This solution is based on configurable physical unclonable functions (PUF) and dynamic physical channel parameters. New user-credentials are used for each new session in order to prevent the risk of the reproduction of the session-key. In [Tia+22], the authors proposed a solution that negotiates a session key by achieving a mutual authentication between Unmanned Aerial Vehicles (UAVs) and the Ground stations (GS). This solution considers a central controller (CS) in which each UAVs need to perform a registration phase. In this phase, the CS sends a series of PUF challenges to the UAV and stores the produced responses. After that, any GS will then be able to authenticate the UAVs by getting access to the list of responses stored in the CS. In [Zhe+22], the authors address the challenge of authentication IoT P2P context. The authentication is based on a list of PUF challenges that needs to locally be stored in each device and used as basis to generate common keys between the devices. There have been other solutions that achieve authentication in IoT context based on PUFs [Bar+19] [Sha+20b] [Mal+22].

However, all these PUF-based solutions adopt a PUF challenge-response mechanism. Therefore, they require, at a certain point of time, a physical intervention on the IoT device to collect

and store the challenge-response pairs to be used in any further.

2.3.4 Radio Frequency Fingerprint Identification

Physical layer fingerprint authentication in IoT is an advanced technique that relies on the inherent imperfections within wireless signals. These imperfections, evident in radio frequency (RF) characteristics and signal propagation, contribute to crafting unique fingerprints for individual devices. Originating from components like transmitters, receivers, antennas, and the surrounding environment, these imperfections establish a distinctive wireless identity for each device [GZC19][JJK22]. Throughout the authentication process, incoming signals with imperfections undergo analysis, leading to the extraction of physical layer fingerprints. This involves comparing the specific wireless irregularities to pre-stored references, a crucial step in validating the authenticity of the device [Zha+23]. In [Che+19], the author present an innovative and lightweight Radio Frequency Fingerprinting Identification (RFFID) scheme employing a two-layer model tailored for authenticating resource-constrained terminals within Mobile Edge Computing (MEC) environments, eliminating the need for encryption-based methods. In the initial layer, MEC devices assume responsibilities such as signal collection, extraction of RF fingerprint features, dynamic feature database storage, and making access authentication decisions. The subsequent layer, overseen by the remote cloud, focuses on learning features, generating decision models, and implementing machine learning algorithms dedicated to recognition. This two-layer framework capitalizes on machine-learning training methods and harnesses the computational capabilities of the cloud, thereby augmenting the authentication rate. Despite some advantages, physical layer fingerprinting in wireless communication confronts several challenges that warrant careful consideration. While it offers inherent security features, reduced credential dependency and real-time authentication, its practical implementation faces noteworthy obstacles. These challenges encompass the intricate demand for sophisticated signal processing algorithms, the susceptibility to potential attacks that could compromise security, the sensitivity to environmental variations impacting reliability, and the overarching need for stringent measures to counteract these issues.

2.3.5 Blockchain-based Authentication

Blockchain-based authentication in IoT involves registering each device on a blockchain network with a unique digital identity. Devices use their private keys to create digital signatures for secure communication, which can be verified by recipients using the sender's public key stored on the blockchain. This approach enhances security by removing single points of control and reducing the risk of unauthorized access. However, it may encounter challenges related to scalability and latency due to the blockchain's resource-intensive nature and consensus mechanisms [Abb+21] [SBA21].

In [Li+18], the authors of this solution challenge the conventional IoT device authentication, which heavily relies on a vulnerable intermediary institution, such as a Certificate Authority (CA) server. This setup is susceptible to single-point failures and internal attacks that can compromise authenticated device data. To mitigate these issues, blockchain technology is introduced as a secure, tamper-proof distributed ledger for IoT devices. Each device is assigned a unique ID recorded in the blockchain, enabling mutual authentication without the need for a central authority. Additionally, a data protection mechanism is devised by hashing essential data, like firmware, into the blockchain, facilitating the immediate detection of data state changes. In [Kha+20], the authors proposed a decentralized authentication and access control mechanism for IoT devices, aiming to enhance their security and ensure the safety and effectiveness of the system. The mechanism leverages fog computing and public blockchain technology to provide lightweight devices with a robust security solution. The system consists of three primary phases: initialization, device registration, and device authentication. During the initialization phase, systems and devices are registered to ensure unique identification. In the device registration phase, smart devices connect to the network, associating with their respective systems. Device authentication is carried out through blockchain-enabled fog nodes, allowing only authorized devices to join the network. The device-to-device communication phase facilitates secure interactions between device. In [Wan+19], the authors propose a private blockchain technology and smart contracts to handle new nodes joining the Internet of Vehicles (IoV) network. The contract node group, comprising verified cloud servers, roadside units (RSUs), and vehicle manufacturers, utilizes a Rayleigh consensus mechanism to approve or reject new joining requests. If over 51% of nodes grant their signatures, the new

node is accepted, adding a new block to the blockchain. Moreover, this solution broadcasts the identities of suspicious nodes to prevent future malicious attempts. Vehicle authentication uses Public Key Infrastructure (PKI) with cryptographic accumulators in two phases. First, the vehicle sends its ID and public key to an RSU, which validates and forwards the information to the Certificate Authority (CA). In the second phase, the CA verifies the data and generates a session key. The process concludes with the exchange of digital certificates. While this solution is highly efficient in authentication, larger networks may encounter packet loss during vehicle registration and key distribution.

2.3.6 Machine learning based Authentication

Machine learning-based authentication within the IoT context is a dynamic approach that utilizes advanced algorithms and models to verify devices [Ist+21]. When a device initiates authentication, it transmits data containing behavioral patterns, device attributes, or sensor readings to a machine learning model. This model processes the data, identifying unique patterns associated with the device's legitimate behavior. Over time, it continually learns and adapts to these patterns, making it more challenging for malicious devices to impersonate legitimate ones. This adaptability is a significant advantage, as it can identify new patterns without manual updates. Additionally, it can detect anomalies, such as unusual device behavior, enhancing security.

In [Das+18], the authors propose a novel approach to IoT authentication, treating it as a multi-label classification problem based on the physical I/Q samples of data packets from received signals. The focus is on wireless signal inputs prone to various impairments, such as frequency and timing offsets and dynamic channel changes. Traditional methods struggle to manually engineer features to address these challenges, making the paper advocate for the use of deep neural networks, specifically Long Short-Term Memory (LSTM) networks. The LSTM networks are chosen for their ability to learn temporal dependencies and higher-order correlations in the signal samples, providing rich and discriminative features. The authentication process involves transmitting a preamble, and the LSTM classifier effectively detects and processes these symbols, demonstrating resilience to adversarial attacks in a low-power IoT device testbed. The approach leverages the unique capabilities of deep learning to handle complex

temporal dependencies in wireless signal authentication. In [QSS21], the authors introduced a novel security authentication scheme designed to combat spoofing attacks in IoT networks. Their approach leverages machine learning algorithms to enhance security measures. The authentication method harnesses the physical layer characteristics of wireless channels to differentiate sensors. It also employs neural networks to learn channel fingerprints without requiring knowledge of the communication network model. The authors propose a security framework based on channel differences, aimed at providing lightweight authentication. Furthermore, they introduce a detection approach using Long Short-Term Memory (LSTM) networks, particularly beneficial for sinks supporting intelligent algorithms.
Machine learning-based authentication in IoT initially learns unique device patterns, enabling accurate future authentication. Despite its potential for enhancing IoT security, it grapples with pattern recognition complexities, particularly in large scale and dynamic IoT networks. Additionally, acquiring extensive datasets for training may pose privacy concerns, and preserving user privacy can be a challenge. Furthermore, machine learning-based authentication might face limitations in terms of resource constraints on IoT devices, potentially restricting its application in resource-constrained IoT environments. In summary, it offers a robust and adaptable approach to IoT device authentication, but it needs to address certain practical limitations.

2.3.7 Observations

In Table 2.1, we've outlined a comparison of the key contributions discussed in this section. Each contribution falls within a distinct category of the authentication methods mentioned earlier. This comparison primarily focuses on the features of authentication solutions that we identify as key limitations. Addressing these limitations is crucial for implementing a robust solution for Industrial IoT. For sake of clarity, these features can be explained as follows: The zero-touch feature indicates if no pre-configuration of the device has been done by any entity other than its manufacturer. Specifically, it indicates if the two entities seeking to identify each other must have a registration phase where certain parameters are shared. Multi-hop support refers to the fact that a new joining device is able to connect to the central coordinator of the network through a multi-hop communication method, rather than being limited to

direct communication with its nearby devices. Heterogeneous mutual authentication refers to whether the authentication scheme enables various types of IoT devices and the network coordinator to authenticate each other reciprocally. Application layer dependency refers to the fact that the authentication mechanism operates on the application layer and does not depend on any lower layer of the network(such as the physical layer for instance). This simplifies the implementation of the authentication solution and enhances cross-protocol compatibility. Finally, the scalability and dynamicity features refer to whether the proposed solution can effectively handle a large number of IoT devices and maintain efficiency in scenarios involving high mobility.

Note that many other solutions were proposed for authentication in IoT networks [CL21; Sha+20a]. However, these solutions do not address the limitation presented in this thesis. To the best of our knowledge, there is no solution that considers the lack of a previous touch with the new node before the joining phase. The Configuration-Free and Consensus-Fueled feature of this solution demonstrated its novelty compared to the previous works.

	[MNC20]	[Jan+14]	[Por+14]	[Li+18]	[Das+18]	[Che+19]
Zero-Touch	✗	✗	✗	✗	✗	✗
Support multi-hop	✗	✓	✓	✓	✗	✗
Heterogeneous Mutual Authentication	✓	✓	✗	✓	✗	✓
Scalability	✗	✗	✗	✓	✓	✓
Dynamicity	✗	✗	✗	✓	✓	✓
Application layer dependent	✗	✓	✓	✓	✗	✗

Table 2.1: A comparison with existing works in terms of protocol features

2.4 Detection Methods in IoT

Malicious nodes detection and trust management in IoT have been widely studied with various proposed solutions [Zar+17] [BWH18] [Haj+19]. Multiple elements are taken into consideration in the existing intrusion detection systems, like the strategy placement, the target security threats and the detection method. In what follows, we elaborate each section

separately while giving its main advantages and drawbacks.

2.4.1 Strategy placement

In terms of strategy placement, a solution can be centralized, distributed or hybrid. A centralized solution is placed and executed in one entity, the network's coordinator for example. On the other hand, a distributed solution is executed on every node, while a hybrid solution combines the two previous solutions. In [Abh+18], authors introduce a centralized Intrusion Detection System (IDS) for clustered IoT networks that aims to identify compromised gateways that degrade network performance by corrupting forwarded packets. It utilizes packet drop probability for monitoring gateways, introduces an algorithm for optimizing system parameters and tracks gateways through the downlink channel. This solution primarily concentrates on physical layer attacks while neglecting possible attacks at other layers. In [SCM21], authors present a solution that addresses the vulnerability of Software-Defined Networking (SDN) in wireless sensor networks. They introduce a lightweight, online change point detector that can operate in both centralized and distributed modes. This solution was tested in IEEE 802.15.4 networks. In this approach, the centralized detector achieves high detection rates and can identify the type of attack, while the distributed detector provides information for pinpointing the nodes responsible for the attack.

Centralized solutions may suffer from the vulnerability of a single point of failure. However, such solutions are more efficient for constrained environments due to the reduced execution overhead on the nodes level [SCL20][RWV13][Rah+20]. A distributed solution is more efficient than the centralized one in terms of latency and network overhead [Zho+20] [LY21] [Col+18]. A hybrid solution is a trade-off of overhead and accuracy in specific applications [HHN20].

2.4.2 Target security threats

In term of security threats, the detection system targets specific types of attacks based on which we consider a node as malicious. The most common studied attacks are: denial-of-service (DoS), routing attacks (e.g., selective forwarding attacks, sinkhole attacks, Sybil attacks, wormhole attacks), man-in-the-middle attack, jamming attacks and others[Has+19a] [KBL18]

[Ala+17b] [Kal+22]. In [Cer+15], authors introduce a system designed to detect and isolate sinkhole attacks in IoT networks. It utilizes dynamic clustering for data transmission and employs reputation and trust mechanisms to identify suspicious nodes. In [Kas+13], the authors proposed a DoS detection system integrating an IDS into the 6LoWPAN network of the ebbits framework. It employs an IDS probe to monitor 6LoWPAN network traffic, detecting DoS attacks. In the case of a jamming attack, the DoS protection manager receives an alert from the IDS. To validate the detection, it checks interference levels, loss rates, and the absence of updated information in network managers. The system's advantages include wired connectivity for immunity to wireless attacks, centralized processing on a Linux host, and reduced false positives by leveraging information from other network managers. The architecture is designed to meet the security requirements of real-time industrial environments.
These solutions may be efficient in terms of detection, however they are designed only for a specific type of attack.

2.4.3 Detection method

In term of detection methods, they represent the detection mechanism used in the system. It can be anomaly-based, signature-based, specification-based or others. Anomaly-based IDS in IoT identify intrusions and misuses by comparing network behavior to a normal pattern, highlighting significant deviations as potential issues. In [Lee+14], the authors utilized energy consumption as a metric to evaluate node behavior, constructing models for typical energy usage in mesh-under and route-over routing schemes. Nodes monitor their energy consumption at a rate of 0.5 seconds, and the IDS identifies a node as malicious, eliminating it from the 6LoWPAN route table upon detecting deviations from expected energy usage. Despite asserting its lightweight nature tailored for low-capacity networks, the authors omitted results on false positive rates, a crucial aspect for forming more accurate conclusions about the effectiveness of the proposed approach. In [KH17], authors designed and assessed three IoT IDS mechanisms. Neighbor-Based Trust Dissemination (NBTD), is centralized, with the border router managing trust values based on inputs from a Destination-Oriented Directed Acyclic Graph (DODAG). Clustered Neighbor-Based Trust Dissemination (CNTD) uses a distributed approach, assuming cluster-based segmentation of the DODAG. These clusters are monitored

by cluster heads which aggregate trust values from nodes. If a node's reputation surpasses a threshold, it is blocked, notifying the border router. Tree-Based Trust Dissemination (TTD) aligns with CNTD's topology but reduces node monitoring, focusing on parent nodes only. Leaf nodes are not monitored. However, this solution and other reputation based solutions suffer from attacks where nodes try to increase their reputation values in a malicious way. Moreover, such systems may generate some false alarms due to their strict criteria.
Signature-based IDS rely on a database of known attacks, where system activities or network behavior are compared to predefined attack signatures. In [OKR14], the authors aimed to decrease the computational burden of comparing packet payloads to attack signatures, especially for IoT nodes with limited capacity. Their approach relies on a multiple pattern-detection algorithm designed to expedite the process by employing auxiliary shift values, thereby minimizing unnecessary matching operations. These solutions effectively identify known threats but remain limited in their ability to detect new or modified attacks. Hybrid detection methods combine elements from signature-based and anomaly-based approaches, aiming to enhance overall effectiveness by addressing the limitations of individual techniques.

The main drawback of both the anomaly-based and signature-based solutions is that they are heavy for constrained devices due the size of the profile or the signature stored at the coordinator level. Moreover, the stored model must be updated frequently in order to be aligned with the network evolution and the new possible faced attacks.
In the context a joining phase, the exchanged messages are lightweight, and the role of a join proxy is very simple. Therefore, the join procedure is limited in term of elements and the only way to verify the legitimacy of a join proxy is based on an opinion of the new joining node. Plus, the coordinator has no knowledge of the join procedure handled by a join proxy, therefore, it has no knowledge about its behaviour unless the joining node successfully joins the network.

2.4.4 Machine Learning IDS

Among the most explored approaches are Machine Learning-based detection systems [BG15] [Da +19][Cha+19][VSO17][Agr+22]. In [Ge+19], the authors introduce an intrusion detection scheme for IoT networks that leverages deep learning, specifically utilizing a feed-forward neu-

ral network model for binary and multi-class classification of various attacks. The framework involves phases such as feature extraction, feature preprocessing, training, and classification. However, the critique centers around the lack of information on false positive rates, making it challenging to assess the reliability of the proposed scheme in real-world scenarios. Additionally, the focus on generic features and the absence of considerations for resource constraints in resource-limited IoT devices could impact the system's adaptability and scalability in practical deployments. In [DC18], the authors recommend using fog computing in IoT systems to strengthen intrusion detection, with the goal of improving efficiency and reducing data transmission to the cloud. They introduce a distributed deep learning approach designed to identify both known and novel intrusion attacks. The focus is on the advantages of fog computing, such as scalability, independent local attack detection, and expedited data training. The proposed architecture involves a master IDS collaborating with distributed IDSs, ensuring parameter updates and synchronization. While demonstrating commendable accuracy in multi-class detection, it is important to examine the mentioned extended training time and assess its practical implications in real-world IoT scenarios. Moreover, the paper could benefit from a more comprehensive discussion of potential challenges or limitations inherent in the proposed fog-based intrusion detection system. In [Xia+16], the authors delve into PHY-layer authentication, focusing on leveraging radio channel information like received signal strength indicators for detecting spoofing attacks in wireless networks. The interactions between a legitimate receiver and potential spoofers are framed as a zero-sum authentication game. The receiver optimizes its utility by selecting a test threshold in the hypothesis test, considering Bayesian risk in spoofing detection. Simultaneously, spoofers strategize to minimize the utility of the receiver by determining attack frequencies. The paper derives the Nash equilibrium for the static authentication game and explores a repeated PHY-layer authentication game for dynamic radio environments. Recognizing challenges in obtaining precise channel parameters, the paper introduces spoofing detection schemes using Q-learning and Dyna-Q, achieving optimal test thresholds through reinforcement learning. Practical implementation and evaluation via experiments in indoor settings, including simulations, validate the effectiveness of the proposed strategies.

These methods rely on factors like data availability, feature diversity, and the presence of classification labels. During join procedures in IoT, lightweight protocols are employed due to

the resource-constrained nature of the environment. Consequently, there is a lack of feature extraction and data acquisition during this phase, making it challenging to apply conventional intrusion detection methods. Moreover, when a proxy node participates in a joining phase, the network's coordinator is unable to classify its behaviour as malicious or honest.

2.4.5 Observations

Even though the detection system is very large and well studied topic, the existing solutions do not address the limitation presented in this thesis. To the best of our knowledge, until this date, there is no solution proposed for the detection of malicious nodes in the context of a joining phase.

2.5 Summary

In this chapter, we have initially elaborated the communication protocols used in IoT. Then, we have presented an overview of the IIoT protocol 6TiSCH, along with its Constrained Join Protocol (CoJP), which relies on the conventional approach of pre-sharing a key between the joining node and the network's coordinator. Later, we have provided a broad overview of the security aspects in IoT and discussed its inherent security limitations and challenges. Following that, we have outlined the current research efforts addressing the primary issues during the joining phase: authentication and the detection of malicious proxy nodes. It became evident that most existing authentication methods are ill-suited for the unique constraints of large-scale and dynamic Industrial IoT networks. Similarly, previous work on malicious node detection and trust management falls short in addressing the challenges posed by the joining phases, characterized by limited elements.

3 Consensus-based Mutual Authentication Scheme for Industrial IoT

In this chapter, we detail our proposed solution for a mutual authentication between a new joining node and the network coordinator. Our solution assumes no pre-shared key previously configured by an operator. End devices must only contain a pre-installed certificate configured by the manufacturer.

We assume that the network coordinator Crd is a fully trusted entity that manages the network security. To the opposite, a node in the network may act in a malicious way at this phase, thus it cannot be fully trusted. We assume however that no more than one third of the nodes in the network are malicious, similarly to Byzantine fault tolerance assumption [LSP19].

For the rest of this chapter, refer to Table 3.1 for details on the parameters used.

3.1 Our Architecture

Our solution is proposed for large scale and dynamic networks. We consider networks where the coordinator is one central entity responsible to control the network's functions and maintain its security. It manages the network communication, devices connectivity depending on the implemented application. The nodes are all the devices connected to this coordinator through direct or indirect links. We consider a mesh topology where multi-hop connectivity is possible. A Proxy node is a node of the network in a direct link with a joining node, and playing the role of intermediate between this new node and the coordinator during the joining phase.

Table 3.1: Table of notations

Notation	Description
Z_p	Finite field of order p
E	Elliptic Curve over Z_p
G	Cyclic group of order p issued from E
B	Generator of G
$Q(x)$	Polynomial of degree m
$P_i(x_i, y_j)$	A point generated from $Q(x)$
Crd	Coordinator
Sk_{crd}	Private key of Crd
Pk_{crd}	Public key of Crd
$f1$	Mapping function from G to Z_p
$f2$	Mapping function from Z_p to G
w	Random element in Z_p
S	Secret key
a_i	Coefficient of $Q(x)$
$Sk_{NewNode}$	Private key of the joining node
$Pk_{NewNode}$	Public key of the joining node
σ_i	Signature of P_i with Sk_{Crd}
HT	A hashtable at the coordinator level
E_{Node_i}	A Proxy node in contact with the joining node
$S_{E_{Crd}}$	Set of N Proxy Nodes to be contacted by the joining node
P_{Node_i}	Packet of points collected by E_{Node_i}
P	Set of points' packets coming from N contacted Proxy nodes
C_X	Set of all possible combinations in P
SC_i	Set combining the points for each element in C_X
FSP_i	A subset of $m+1$ distinct points randomly chosen from each SC_i
S'	Retrieved Group key
C	A challenge consisting of a random series of bits
El	Random element in G
CS	Session key (H(El))
CT	The challenge C encrypted with CS
EB	Enhanced Beacon

3.2 Prerequisites

3.2.1 Shamir Secret Sharing

Shamir's secret sharing method [Sha79] consists in dividing a secret into parts and sharing them in a way that a minimum number of shares is needed to reconstruct the secret. For that, a finite field Z_p is adopted to generate elements used as coefficients for a polynomial $Q(x)$ of degree m. The secret here consists of the value $Q(0)$ as represented in Figure 3.1. To redefine

this polynomial, at least $m+1$ distinct points generated from this polynomial are needed.

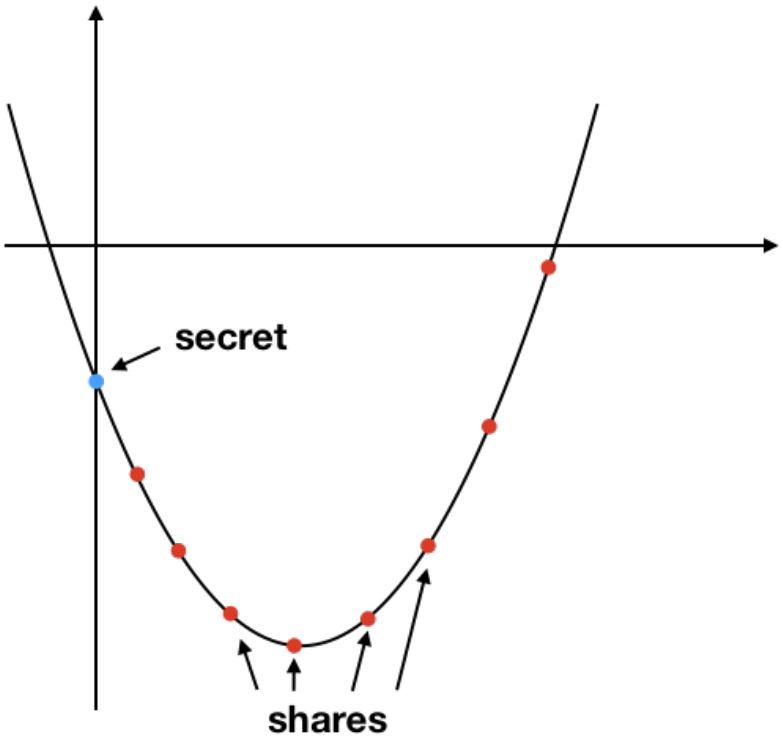


Figure 3.1: Illustration of Shamir Secret Sharing.

Given $m+1$ points, $\{P(x_i, y_i)\}, i \in \{1, .., m+1\}$, generated from a polynomial $Q(x)$, Lagrange method is used for polynomial interpolation as formulated in Equation (3.1).

$$f(x) = \sum_{i=1}^{m+1} y_i \prod_{j=1, j\neq i}^{m+1} \frac{x - x_j}{x_i - x_j} \tag{3.1}$$

3.2.2 Elliptic Curve Discrete Logarithm Problem

Given E an elliptic curve over a finite field Z_p of order p. Let two points P and $Q \in E$ and x is an integer. The elliptic curve discrete logarithm problem is defined as, given P and Q, to find x such as $Q = x \cdot P$. This problem can be considered as a one-way function $f : X \rightarrow Y$ where it is possible to calculate $y = f(x)\ \forall x \in X$ but it is very hard to calculate x given $y \in Y$. This challenge, coming from complex mathematical structures and the absence of known efficient algorithms (no known algorithm capable of solving the problem in polynomial time)

underscores the cryptographic strength of elliptic curve cryptography. This trapdoor function is well invested in cryptography [SS98].

3.2.3 Byzantine Fault Tolerance

Byzantine Fault Tolerance (BFT) is a principle in distributed computing aimed at mitigating faults in systems where components may behave maliciously or provide inaccurate information. The term "Byzantine" refers to arbitrary and malicious behavior, as illustrated in the Byzantine Generals' Problem—a theoretical challenge in consensus algorithms. In a BFT system, the objective is to achieve consensus among nodes or participants, even if some exhibit faults or act maliciously. This is particularly crucial in security-sensitive environments like blockchain networks and distributed databases. Practical Byzantine Fault Tolerance (PBFT), a specific BFT implementation, focuses on establishing agreement among distributed nodes by assuming that at least two-thirds of them are honest [CL+99]. In PBFT, nodes collaborate to propose and agree on command sequences, ensuring consensus when a significant majority of nodes operate faithfully, addressing the challenges of Byzantine Fault Tolerance.

3.3 Main Idea

Our approach takes into consideration the imbalance of resources in the network. The Crd has more capabilities than the network's resource-constrained nodes. For this reason, we propose to use two different mechanisms to achieve mutual authentication.

At the first step, the Crd authenticates the new node which asks to join the network. This phase relies on a certification-based authentication. Having the capability to access Certificate Authorities through chains of trust, the Crd can verify a certificate provided by a joining node.

In a second step, the joining node proceeds with the Crd authentication. It is based on a consensus: multiple nodes existing in the network prove for the joining node the legitimacy of the Crd. This is done by revealing a secret shared between the Crd and the network's nodes using Shamir secret sharing. Hence, multiple nodes contacted by the joining nodes, called Proxy nodes, must collaborate together in order to reveal this secret contained at the Crd.

Therefore, a collusion of many nodes in the network points towards its secret.

Once both sides are mutually authenticated, a key is established between them for the subsequent exchanges.

3.4 Setup Phase

While bootstrapping the network, the Crd executes the following steps in order to prepare the parameters used by our protocol:

- Let G be a cyclic group of order p issued from an elliptic curve and B is its generator.
- Let Sk_{Crd} and Pk_{Crd} a pair of private and public keys respectively, used by the Crd and the nodes to sign and verify exchanged messages.
- Define two mapping functions $f1 : G \rightarrow Z_p$ and $f2 : Z_p \rightarrow G$ such that $\forall e \in G$, we have $f2(f1(e)) = e$.
- Compute the group key $S = w \cdot \mathrm{B}$, where w is a random element in Z_p,
- Let $Q(x) = a_0 + a_1 x + a_2 x^2 + ... + a_m x^m$ of degree m. The coefficients a_i, where $i \in \{1, .., m\}$, are random elements in Z_p and $a_0 = f1(S)$.
- The Crd creates a hashtable HT to register authentication session identifiers along with session keys for each new node willing to join the network.

In an industrial context, we certainly have nodes that are already in the network. Therefore, each node i existing in the network is configured by the coordinator with a tuple $(\sigma_i, P_i, Pk_{Crd})$, where:

- $P_i = (x_i, y_i)$, where $y_i = Q(x_i)$, is a distinct random point generated from the secret polynomial $Q(x)$.
- σ_i is the signature of P_i as:

$$\sigma_i = Sign(H(x_i || ',' || y_i), Sk_{Crd})$$

where $Sign$ is a signature scheme and H is a hash function.

3.5 Joining Node Authentication

The Crd needs to authenticate the new nodes looking to join the network. We propose to use a certificate-based authentication as the Crd has the necessary computational and storage resources. Since we consider an industrial context, we can assume that the Crd has prior knowledge of the type of nodes that may join its network. Therefore, the Crd only stores certificate chains of trust related to those types of nodes. Moreover, we can assume that in each joining node, a certificate is installed by the manufacturer after production, as part of the non volatile data, and no other configuration is needed.

Our proposed certificate-based authentication of the joining node is established as follows:

- Proxy nodes send Enhanced Beacons EB to the joining nodes on a regular basis.
- The joining node collects EBs advertising the network. It does so for some time T, trying to get in touch with more nodes advertising this network.
- The joining node sends its certificate, as a join request, to all contacted Proxy nodes.
- The Proxy nodes forward the request to the Crd.
- The Crd verifies the certificate and authenticates or not the joining node.
- In case it is authentic, the Crd saves in HT, a hash of the public key of the joining node $H(PK_{NewNode})$, as an authentication session identifier.
- The Crd allows Proxy nodes to continue the communication with this joining node by sending back a response containing the joining node's public key $PK_{NewNode}$.

Note that, proceeding in the execution of the protocol using $PK_{NewNode}$ to encrypt the further exchanges: (1) proves that the joining node owns its corresponding secret key, and (2) allows at the same time to secure the communication with the Proxy nodes.

Figure 3.2 illustrates this phase.

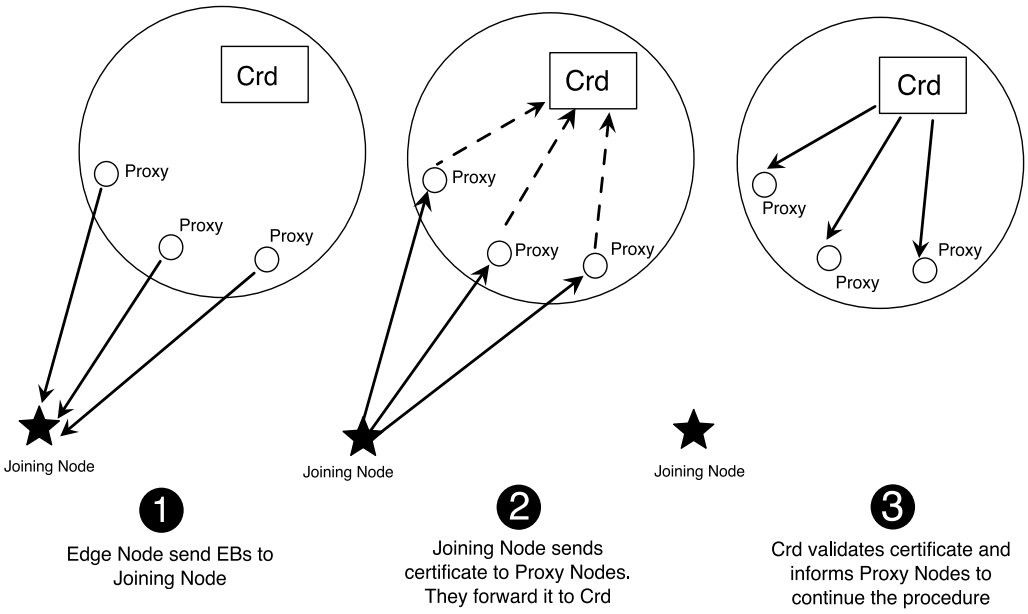


Figure 3.2: Steps for authenticating the joining node by the coordinator

3.6 Coordinator Authentication

In this section, we present the whole Crd authentication. Given a set $S_{E_{Crd}} = \{E_{Node_1}, E_{Node_2}, ..., E_{Node_N}\}$ of N Proxy nodes to contact, the joining node performs the following steps:

- The joining node requests from each Proxy node E_{Node_i} a packet $P_{Node_i} = \{P_{i,1}, P_{i,2}, ..., P_{i,m}\}$ of m points where $P_{i,k} = (x_k, y_k), k \in \{1, .., m\}$.
- Each E_{Node_i} receiving the joining node request, asks for $m-1$ distinct points $P_{i,k}$ from random nodes in the network.
- Each node j receiving E_{Node_i}'s request, sends its own point $P_j = (x_j, y_j)$ and the signature σ_j provided by the Crd.
- For each received (P_j, σ_j), the Proxy node verifies the signature σ_j using the Crd's public key Pk_{Crd}.
- The Proxy node E_{Node_i} forms a packet of points P_{Node_i} which contains the collected points (verified in the previous step), as well as E_{Node_i}'s own point.
- The Proxy node sends P_{Node_i} to the joining node encrypted with $PK_{NewNode}$.

- The joining node decrypts the packets with its private key $SK_{NewNode}$ and forms a set $P = \{P_{Node_1}, P_{Node_2}, ..., P_{Node_N}\}$ coming from N received Proxy nodes.
- The joining node constructs a set $C_X = \{C_1, C_2, ..., C_X\}$ of all possible combinations in P, where $X = C^2_{n(P)}$ is the number of combinations without repetition of two sets of points $P_{Node_i} \in P$.
- $\forall\ C = \{P_{Node_i}, P_{Node_j}\}$, where $P_{Node_i}, P_{Node_j} \in P$, define:

$$SC_i = P_{Node_i} \cup P_{Node_j} = \{P_{i,1}, ..., P_{i,m}, P_{j,1}, ..., P_{j,m}\}$$

- Let $FSP_i \subset SC_i$ be a subset of $m+1$ distinct points randomly chosen from each SC_i, defined as:

$$FSP_i = \{P_1 = (x_1, y_1), ..., P_{m+1} = (x_{m+1}, y_{m+1})\}$$

- FSP_i is used to reconstruct the secret $Q_i(0)$ using Lagrange polynomial interpolation as follows:

$$Q_i(0) = \sum_{k=1}^{m+1} y_k \prod_{z=1, z\neq k}^{m+1} \frac{-x_z}{x_k - x_z} \tag{3.2}$$

We note that the polynomial interpolation (3.2) is based on the points collected from nodes already active in the network. We recall that these points were initially provided by the Crd and generated from the polynomial $Q(x)$ defined in the setup phase. Therefore, an interpolation polynomial on these points should provide the same value $Q(0)$ defined by the Crd.

Considering all the combinations done on the points collected by the joining node and provided by the Crd, we will end up with $X = C^2_{n(P)}$ repetitive values $Q_i(0) = Q(0)$ computed during the previous step.

Being able to repeatedly retrieve the same value $Q(0)$ allows to achieve a consensus through multiple nodes in the network, directing the joining node towards the Crd's identity.

Nevertheless, malicious Proxy nodes may be among the Proxy nodes collecting points. Obviously malicious Proxy nodes aim to deviate the interpolation's results, leading to some values $Q_i(0) \neq Q(0)$ among the X interpolations performed by the joining node. Whereas, as long as the rate of malicious nodes in the network is less than 33%, the consensus can be achieved by

considering the most frequent value among the X interpolations. Hence, the success of the consensus is based on the majority of honest nodes contacted during the joining phase.

3.7 Key Establishment

The Crd authentication phase allowed the joining node to discover $Q(0)$. Therefore, the joining node proceeds to verify and establish a common key with the Crd through the following procedure based on the El Gamal exchange:

- The joining node retrieves the group key as:

$$S' = f2(Q(0)) = f2(f1(S)) = S$$

- The joining node generates the following parameters: a random element r in Z_p, a challenge consisting of a random series of bits $C \in \{0,1\}^*$ and a random element $El \in G$;
- The joining node sets the session key $CS = H(El)$ and computes $CT = enc(C, CS)$, where $enc()$ is a symmetric encryption algorithm;
- The joining node calculates $Sig = Sign(H(r \cdot B || r \cdot S + El || CT), SK_{NewNode})$ where $Sign$ is a secure signature scheme, H is a hash function and $SK_{NewNode}$ is the private key of the joining node;
- The joining node sends $(r \cdot B, r \cdot S + El, CT, Sig, PK_{NewNode})$ to Crd;
- The Crd receiving $PK_{NewNode}$, retrieves the authentication session by looking up for $H(PK_{NewNode})$ in HT. Then, it verifies the signature Sig using $PK_{NewNode}$;
- We recall that the group key has been generated as $S = w \cdot B$ where $w \in Z_p^*$ was a random value chosen by the Crd during the setup phase. Thus, the Crd calculates:

$$El' = (r \cdot S + El) - w \cdot (r \cdot B) = El$$

- The Crd recovers $C = dec(CT, H(El'))$, where $dec()$ is a symmetric decryption algorithm;
- The Crd sends recovered C to the joining node as a response to the challenge and saves $H(El')$ as a session key for the joining node's session identifier in HT;

- The joining node considers the key establishment has succeeded if it receives the response C before a time T.

Figure 3.3 illustrates the protocol exchange sequence.

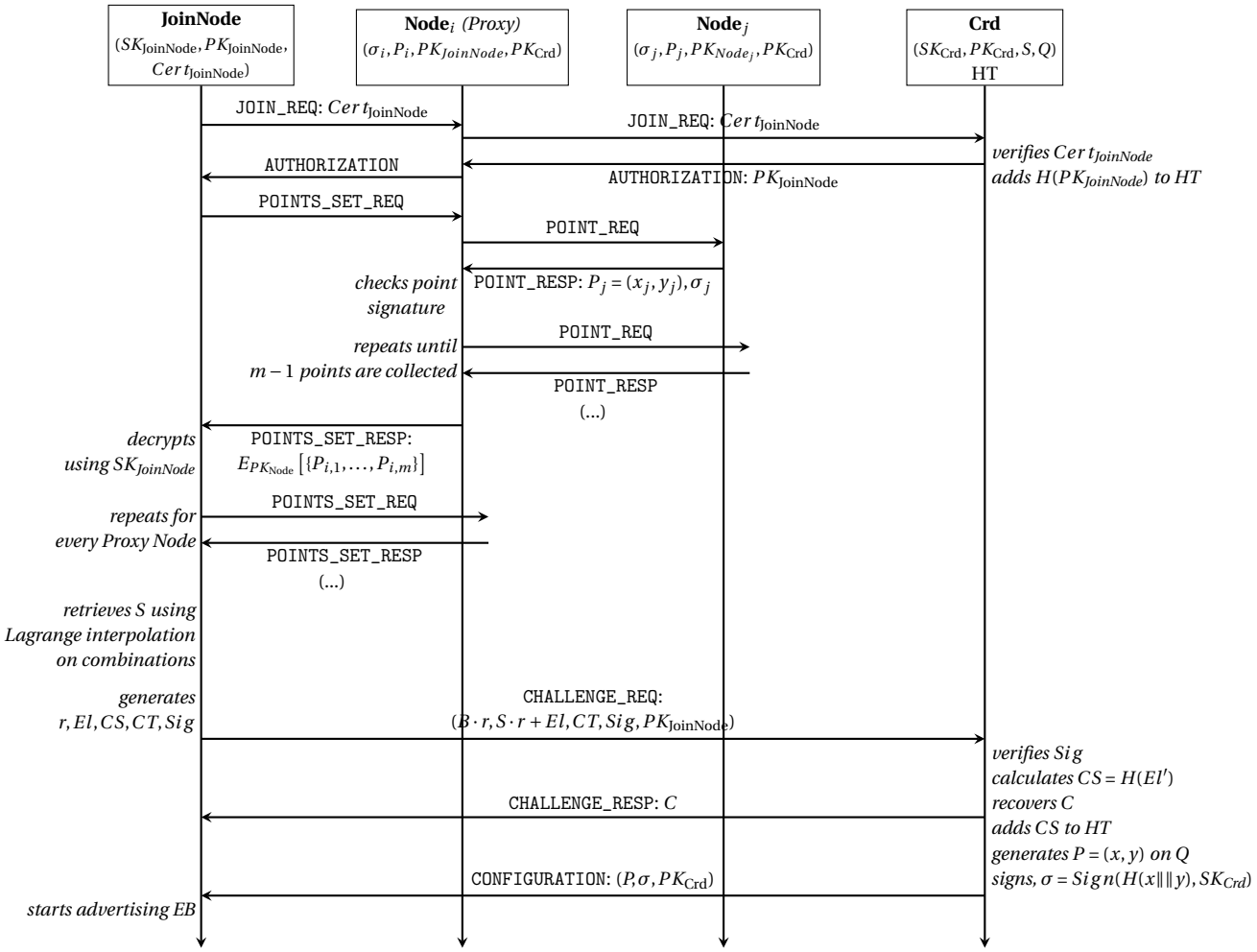


Figure 3.3: Overview of the proposed mutual authentication protocol. The protocol includes 3 main phases : 1) authentication of the joining node; 2) authentication of the Coordinator by collecting points, making a consensus and submitting a challenge; and 3) establishment of a common key.

3.8 Collect strategies: Global vs. Local

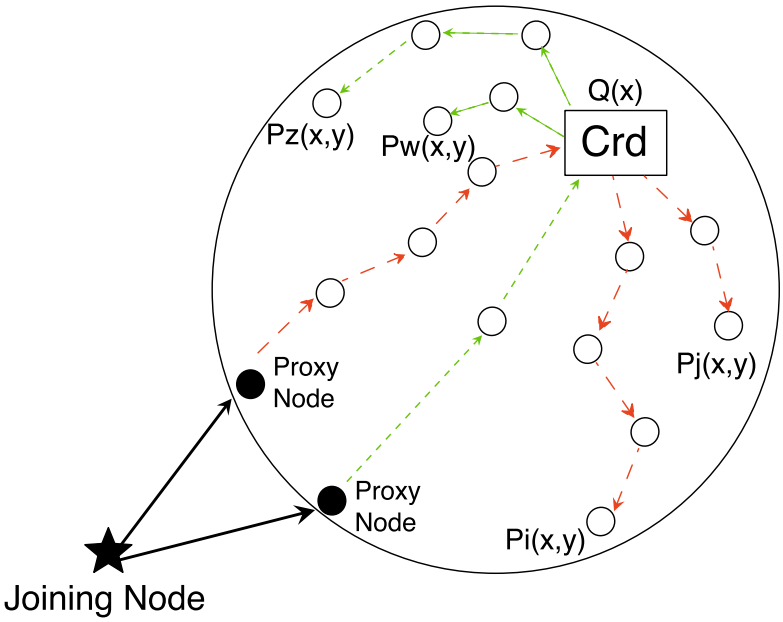
In the previous sections, we described all the steps to be executed for a joining phase. In this section, we focus on the collect phase when a Proxy node has to collect coordinates of points from the network. This phase might have a high impact on the communication overhead of our protocol, especially in multi-hop networks. Therefore, we discuss in this section two different collect strategies to be executed by a Proxy node. In the next section, we evaluate the performance of our solution using both strategies, in order to compare their efficiencies and shed light on the strong and weak points of each of them.

3.8.1 Global mode

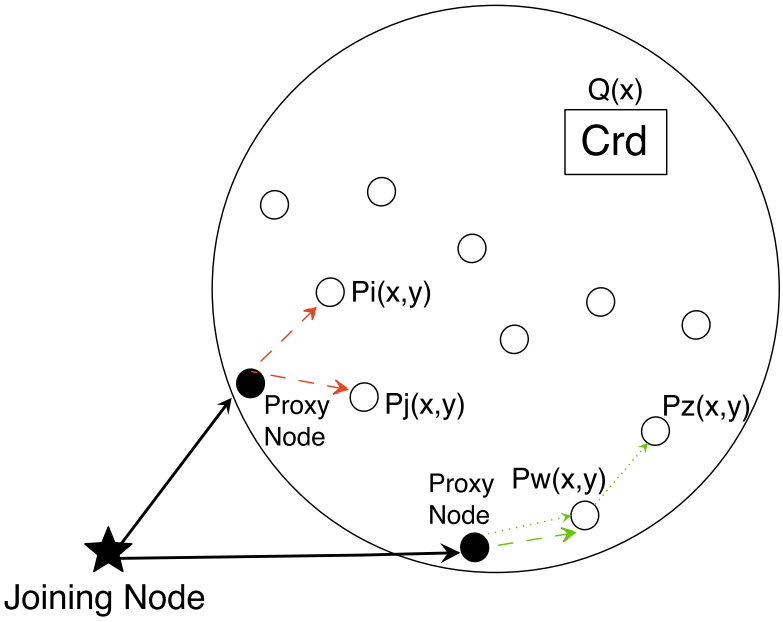
In the *Global mode*, a Proxy node reaching the collect phase delegates the point collection to the Coordinator. To this end, it sends a request all the way to the Coordinator which is responsible to collect the packets of points and send them back to the requesting Proxy node. Therefore, the following steps are executed:

- The Proxy node starts by sending a message `POINTS_SET_REQ` to the coordinator, requesting a packet of points.
- The coordinator chooses $m-1$ random nodes from the network and sends each one a message `POINT_REQ`, requesting its point coordinates.
- Each requested node sends back to the coordinator a message `POINT_RESP` containing its point coordinates.
- When receiving a message `POINT_RESP`, the coordinator forwards it to the Proxy node.
- The Proxy node ends up by receiving $m-1$ messages `POINT_RESP`.
- The Proxy node adds its own point to this packet and sends it in a message `POINTS_SET_RESP` to the joining node.

We note that the coordinator chooses the nodes in the network in a random manner, making this collect random and independent of the position of the coordinator and the requesting Proxy node in the network. Figure 3.4a illustrates this Global collect mode.



(a) Global mode



(b) Local mode

Figure 3.4: Different collect strategies for multi-hop networks. (a) In the *Global mode*, each Proxy node requests a set of points from the Coordinator which is then responsible for requesting points from randomly selected nodes. The example shows distinct sets of points being requested by two Proxy nodes with the corresponding multi-hop paths (in green and red). (b) In the *Local mode*, each Proxy node requests points from direct neighbors (in red). If the number of neighbors is insufficient a flooding strategy is adopted to collect points from nodes further away (in green).

In a multi-hop network, the decision of which path to follow between a given pair of nodes is delegated to the routing protocol. In addition, a message sent from one node to another will possibly traverse multiple wireless links.

3.8.2 Local mode

In the *Local mode*, a Proxy node reaching the collect phase sends a request to its direct neighbours. If the number of direct neighbours is limited, multi-hop requests are sent through selected neighbors until a sufficient number of distinct points is collected. Therefore, the following steps are executed:

- The Proxy node starts by sending a message `POINT_REQ` to random $m-1$ of its direct neighbour nodes, requesting a point from each.
- Each requested node sends back to the Proxy node a message `POINT_RESP` containing its point coordinates.
- The Proxy node adds its own point to this packet and sends it in a message `POINTS_SET_RESP` to the joining node.

Figure 3.4b illustrates this Local collect mode. Note that in case the requesting Proxy node has a number of neighbours $Nr < m-1$, it resorts to a flooding strategy to collect points from nodes that are further away. We do not detail this strategy further in this work.

3.9 Attack Models

In this section, we present the attacks that we consider during the authentication phase. We distinguish two types of attackers, the malicious insiders and the active attackers. The malicious insiders are the malicious nodes inside the network having authorized access and exploit their privileges in order to carry out malicious activities threatening the network security. The active attackers are outside the network and do not have authorized access to this network, therefore, they may conduct different type of attacks in order to gain access, reveal shared data or interrupt the network service.

3.9.1 Malicious insiders

The main vulnerability that we may have is the existence of malicious insiders, or malicious nodes in the network, controlled by remote nodes, or programmed to perform malicious behaviours. These malicious insiders can be either nodes in the network or the Proxy nodes contacted by the joining node.

When Proxy nodes are malicious insiders, these malicious nodes may have two purposes, either fail the authentication or lead the joining node to authenticate a wrong Crd. For that, a malicious Proxy node can conduct these types of attacks:

- **Individual attack:** A joining node requests a Proxy node to collect points from other nodes in the network in order to reconstruct the polynomial and reveal the secret. A malicious Proxy node creates its own polynomial and generates the requested number of points from it, instead of contacting other nodes, trying to fail the interpolation done by the joining node.
- **Collaborative attack:** All malicious Proxy nodes in the network have a sort of agreement between themselves. They have the same polynomial from which they generate points to be sent to the joining node. Hence, they all work together looking for leading the interpolation done by the joining node to one wrong Crd.
- **Impersonation attack:** A Proxy node tries to impersonate the joining node or the Crd while playing the role of intermediate between them.

When other nodes in the network are malicious insiders, their main purpose is to fail the authentication. Therefore, when a malicious node is contacted by a Proxy node in order to collect its point for the consensus, this malicious node may send a false point. However, during the collect phase, points collected from nodes are signed by Crd. A contacted malicious trying to conduct an impersonation leading the joining node to another impersonated coordinator, will have to sign its sent point with the real coordinator private key. This signature is verified at the Proxy node level. A contacted node in the network does not have access to the coordinator's private key. Moreover, forging a digital signature is verified as a very challenging task [KP17]. Note that other types of attacks inside the network threatening a multi-hop wireless network, such as jamming attack or nodes selfishness, etc., are not considered in the scope of this work.

3.9.2 Active attackers

For the active attackers, we consider the attackers outside the network and targeting the communication between the Proxy nodes and the joining nodes. The main two phases in our authentication mechanism vulnerable to such type of attacks are the phases of the coordinator authentication and the key establishment. However, during the coordinator authentication, the packets sent from a proxy node to the joining node are encrypted with the public key of the joining node. During the key establishment phase, the packets sent from the joining node to Proxy Nodes are signed by this joining node. Therefore, the risks of eavesdropping and Man-in-the-middle attacks are excluded from these phases. Moreover, attackers can conduct multiple type of attacks like DoS, jamming attacks targeting to intercept the communication between the joining node and the Proxy node. We consider these type of attacks out of scope of this work.

3.10 Analysis: Impact of the possible attacks on our solution

In this section, we analyse theoretically the capability of our solution to face the attacks that we consider in our attack model. We recall that in order to authenticate the Crd, the joining node requests packets of points from multiple Proxy nodes. After that, these packets of points are combined two by two to achieve a consensus based on Shamir's secret sharing scheme. However, the existence of malicious nodes among the contacted Proxy nodes leads to incorrect calculated results for some of these combinations.

In the worst case scenario that we are considering, where the third of these Proxy nodes are malicious nodes, we have for any number of Proxy nodes $N > 3$, three categories of combinations of packets: (1) a combination of packets collected by two honest Proxy nodes; (2) a combination of a packet collected by an honest Proxy node and another packet collected by a malicious one; and finally (3) a combination of packets collected by two malicious Proxy nodes.

For the first category, the calculation always leads to the correct secret. For the second one, each calculation leads to a wrong secret that is not repeated in the other calculations. For the

third category, the calculation results depend on the type of the conducted attack. In terms of percentage of repetition of each secret resulting from the combinations, we notice that in the first category of combinations we have $C^2_{\frac{2\cdot N}{3}} / C^2_N$. The percentage of each secret found in the second category is $1/C^2_N$. Secrets found in the third category may be repeated or not depending on the attack scenario.

- **Individual Scenario:** For the third category, in the case of an individual attack, a different wrong secret is calculated for each different combination. The percentage of each secret resulting from the combination is $1/C^2_N$.
- **Collaborative Scenario:** Also for the third category, in case of a collaborative attack, the same wrong secret is calculated for all the combinations of nodes belonging to this category. The percentage of this secret is equal to $C^2_{\frac{N}{3}} / C^2_N$.

In all cases, we clearly see that the percentage of the repetition of the secret found through combinations of the first category always represents the majority. Therefore, even under individual and collaborative attack scenarios, our protocol always achieves the consensus for any number of Proxy nodes $N > 3$ as long as the rate of malicious Proxy nodes does not exceed $N/3$. Combining each pair of packets for Lagrange interpolation instead of doing the calculation multiple times while choosing random points from all the combined packets, leads to lower probability of attack success.

- **Impersonation Scenario:** In order to conduct an impersonation attack, an attacker faces multiple challenges on multiple phases of the joining procedure. For an impersonation attack during the key establishment phase, a malicious Proxy node cannot impersonate the joining node since the parameters sent by the joining node during this phase are signed with its private key. Likewise, a malicious Proxy node cannot impersonate the Crd since it needs to be able to recover the value w (known only by the Crd), given $S = w \cdot B$. This is necessary to get the session key and to send back the challenge waited by the joining node. However, succeeding to lead this attack is equivalent to solving elliptic curve discrete logarithm problem known to be hard in multiplicative cyclic groups as described in 3.2.2.

Total number of nodes	100
Degree m of $Q(x)$	[2-10], default value=2
Rate of malicious nodes	{10,20,33}%, default value=33%
Simulation rounds	1000

Table 3.2: Simulation parameters.

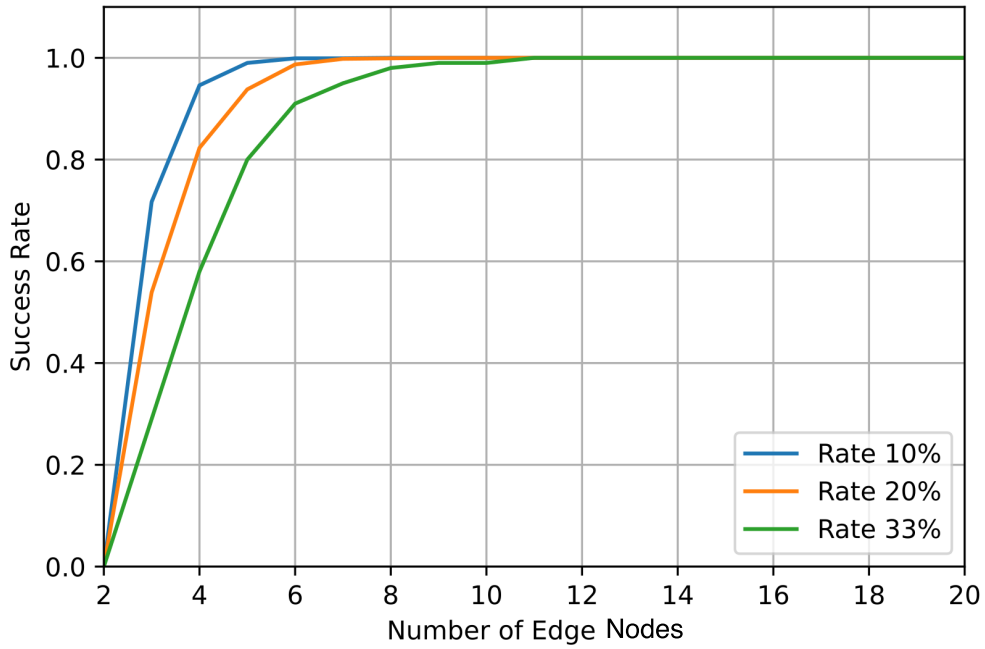
3.10.1 Security Protocol Evaluation

In order to evaluate the performance of our protocol, we conducted an experiment in a distributed system. In our experiment, we consider one Crd, multiples nodes and joining nodes. Each one of these entities is implemented as a thread executing its tasks defined in our protocol and communicating with other entities. Nodes can be either honest or malicious. If a node is malicious, it can be configured to perform an individual or a collaborative attack (Section 3.9). The Proxy nodes executing the consensus are randomly chosen among the nodes launched in our simulation. The simulation parameters (Table 3.2) are varied in order to evaluate their impact on the success rate of the authentication. The results are presented in the following subsections.

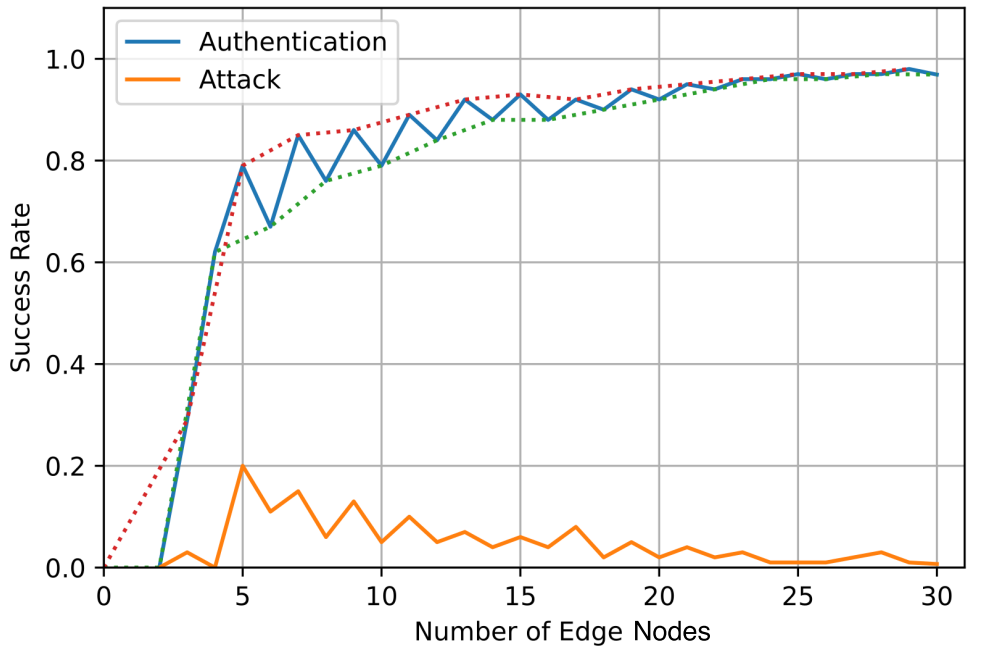
Security Robustness

Figure 3.5a represents the variation of authentication success rates according to the number of Proxy nodes, for different rates of malicious nodes conducting an individual attack. As we can see, in the worst case (33% of malicious nodes in the network), we reach a rate of 0.8 of successful authentications starting from a number of Proxy nodes equal to 5. This value starts to converge to 1 starting from a number of Proxy nodes equal to 8 and less if we consider a rate of 10% or 20% of malicious nodes. Note that for a rate of malicious nodes in the network higher than 33%, to attain the same success rate, a higher number of Proxy node must be contacted. Since this solution is proposed for an industrial context, we consider that the availability of Proxy nodes is not a constraint. Moreover, for an individual attack, the cases where the authentication does not succeed, represent the cases where the consensus has not been achieved, not that an attack has succeeded.

Figure 3.5b represents the same variation as Figure 3.5a in a scenario where malicious nodes (33% of the network) conduct a collaborative attack. Although we notice the convergence of



(a) Individual attacks.



(b) Collaborative attacks.

Figure 3.5: Variation of authentication success rates.

authentication success rates, we also observe an instability in the rates achieved in the case where we have an even and an odd number of Proxy nodes. Indeed, if we separately consider the success rates curve of even numbers of Proxy nodes (green curve) and odd numbers (red curve), we notice that the rate in both curves is increasing. However, the convergence of the red curve is much quicker than the green one. To explain this behavior, we calculated (in Table 3.3) the rate of how many times, in average, the number of malicious Proxy nodes has exceeded the half of the contacted Proxy nodes during the experiment. We note that in the collaborative attack, when the number of malicious Proxy nodes exceeds the half of contacted Proxy nodes, the authentication fails since we achieve a consensus leading to malicious nodes' secret. Now if we check the rates in table 3.3, we can clearly notice that we exceed more often the half when an even number of Proxy nodes is contacted compared to when an odd number of Proxy nodes is contacted. This means that, the collaborative attack succeeds more for even numbers than odd numbers. Therefore, the convergence of authentication success rate becomes slower for even numbers as noticed in Figure 3.5b.

	# Proxy nodes								
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Avg. Exceed Time	0.55	0.26	0.37	0.2	0.32	0.15	0.23	0.13	0.2

Table 3.3: Percentage of times malicious exceeding half Proxy nodes.

Impact of degree m

Given a polynomial of degree m and a number N of Proxy nodes to be contacted to collect $m-1$ points each, the number of messages N_{msg} exchanged during one Crd authentication is:

$$N_{msg} = (m-1) \times N \times 2 + N \times 2 = 2 \times m \times N \tag{3.3}$$

As we can see, the communication overhead depends on the degree m and the number of Proxy nodes to contact.

In Table 3.4, we represent the impact of the degree m on authentication success rates and the number of messages exchanged. As we can notice, increasing the value of m increases the number of messages exchanged but does not have an impact on success rates. Thus, bigger

values of m do not make our protocol more robust. Hence, we can reduce the communication overhead by adopting the smallest value of m.

	Degree m								
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Success Rate (%)	80	79	80	79	77	77	77	78	77
# Messages	20	30	40	50	60	70	80	90	100

Table 3.4: Degree m modification with five Proxy nodes.

3.11 Discussion

This section provides further discussion of the some corner cases.

First, in the case of a network recently booted, the number of nodes may be very limited. This may prevent a complete authentication procedure based on a consensus of multiple nodes. In such case, we suggest that the first few nodes to join the coordinator of the network are pre-configured and fully trusted. Hence, the evolution of the network is made by a consensus of trusted nodes.

Second, the joining node must get in touch with multiple Proxy nodes in the network in order to have a safe enough join procedure. In the case of a dynamic and large scale network like the Industrial IoT case, the availability of multiple Proxy nodes is supposed to be continuous. However, in an temporary isolated case, the joining node can wait for a while to receive Enhanced Beacons from a sufficient number of Proxy nodes.

Third, in order to get a robust performance of our protocol, we suppose that the malicious nodes in the network do not exceed the two third of its size. This assumption, inspired from the Byzantine Fault Tolerance, is a state of a network where our protocol performs efficiently and is not a condition for the correctness of the protocol. We demonstrated through the evaluation of our protocol that if a significant number of Proxy nodes are contacted, this assumption can be relaxed. In particular, in the case of an individual attack, the existence of only some honest nodes between the Proxy nodes is sufficient to have a successful authentication. In the case of a collaborative attack, a majority of honest nodes between the Proxy nodes can ensure a successful authentication. Therefore, depending on the expected number of Proxy nodes and

the type of attacks to be conducted, new probabilities can be calculated and the assumption considering the maximum rate of malicious nodes in the network can be modified for a higher value.

Fourth, the coordinator authentication phase is based on revealing a secret S at the end of the consensus. This secret by itself does not represent any security element of the network and its only purpose is to identify the network coordinator. Moreover, only the coordinator, the entity that created this secret, will be able to establish a key with the joining node. Therefore, revealing this secret by the joining node and the knowledge of this secret by the nodes in the network do not have an impact on the safety of this network. Therefore, there is no need to update this secret after one or multiple joining phases. Additionally, the update or revocation of the key established between the coordinator and the joining node at the end of the joining phase, as well as the link-layer key and the configured security parameters, is considered out of scope if this work.

3.12 Summary

In this chapter, we have proposed a novel mutual authentication scheme for Industrial IoT. The essence of our proposal is to establish mutual authentication, taking into account the inequalities in resources of end devices and coordinator. Indeed, end device authentication is achieved based on certificates while coordinator authentication relies on a consensus executed by active nodes in the network and adapted for their constraints. Finally, a session key is established between these two entities. Notably, our scheme stands apart from many existing solutions in that it does not necessitate any prior configuration of new nodes during network deployment.

We have conducted both theoretical analyses and simulations to assess the resilience of our protocol in attack scenarios, assuming the presence of up to 33% of malicious nodes in the network. The simulation results align with the theoretical analysis, demonstrating the robustness of our protocol against attacks.

In the next chapter we will adopt 6TiSCH protocol as an application to our solution. We will also provide an evaluation covering communication, latency, and energy consumption.

4 Application to 6TiSCH and Performance Evaluation

In chapter 3 we presented our solution for Configuration-Free mutual authentication in Industrial IoT. In order to validate our solution on a real Industrial IoT application, we adopted the 6TiSCH framework. As described in Section 2.1.2, 6TiSCH defines a complete IP-based network stack that targets industrial deployments. This chapter presents a more advanced and realistic evaluation of our security protocol in the context of 6TiSCH, focusing on the cost of communications and cryptographic operations. These evaluation methods are the most related to our study scenario, and inspired from the methodologies of evaluation in the literature [Kri+18] . We adopt the 6TiSCH terminology in the remaining of this section: the joining node is called the Pledge, the Edge node (or Proxy Node) is the Edge Join Proxy (JP) or (Proxy JP), and the Coordinator is the Join Registrar/Coordinator (JRC).

4.1 6TiSCH Integration

To make our experiments realistic and reflecting the operations in a typical 6TiSCH environment, we target a mesh network topology. For this purpose our network layer uses the RPL protocol to perform routing. The JRC is the root of the RPL network. It starts sending RPL DIO messages seeking to construct the network tree. As a consequence, all used paths are embedded in a routing tree (RPL's DODAG) which is rooted at the JRC. At the data-link and physical layers, we employ IEEE 802.15.4e operating in TSCH mode.

As part of the network boot, the JRC generates the polynomial $Q(x)$ of degree $m = 3$. In this

experiment, we opt for $m = 3$ rather than $m = 2$. This choice is made because, in the case of $m = 2$, the Proxy node only needs to collect a single point and include its own point in the packet. However, such a configuration may not provide a representative enough evaluation of communication. The coefficients of $Q(x)$ are elements generated from a finite field Z_p. The JRC then configures the existing network nodes by sequentially sending each of them the coordinates of a different point generated from $Q(x)$. Beside that, we propose to modify the Enhanced Beacon (EB) of 6TiSCH in order to advertise the degree m of $Q(x)$.

The Joining phase of the protocol starts as soon as a Pledge receives a TSCH Enhanced Beacon from a Proxy JP. All the message exchanges described in Section 2.1.2 between the Pledge, the Proxy JP and the JRC are all executed at the application layer. In order to compare their performance, the two collect strategies presented in Section 3.8 are implemented. When operating in *Global mode*, a Proxy JP launches the collect phase by sending a `POINTS_SET_REQ` message to the JRC which is then responsible to collect the packets of points from randomly selected nodes. Since we use RPL in non-storing mode, every message sent between two nodes in the network will go over a multi-hop path through the JRC.

When operating in *Local mode*, a Proxy JP requests points by sending `POINT_REQ` messages to its IPv6 link-local neighbors, relying on single-hop communications instead.

4.2 Implementation

We implemented our scheme above the Contiki-NG operating system as it offers a complete and well-tested 6TiSCH network stack [Oik+22]. Aiming to implement our scheme on well constrained IoT nodes that reflect the reality of IoT devices, we chose to use the Zolertia Z1 platform. It is based on a second generation MSP430F2617 low power micro controller, designed on a 16-bit RISC architecture and featuring 8KB of RAM and 92KB of Flash memory. It is equipped with a CC2420 IEEE 802.15.4 compliant radio transceiver.

Moreover, to implement the cryptographic primitives of our protocol, we relied on the `micro-ecc` library [Mac]. It consists of energy-efficient implementation of NIST curves, written in C language and supported by embedded systems. In our implementation, it is used to generate elements from an elliptic curve, for the modular operations and the key estab-

lishment. We use the standard curve `Secp160r1`. As a consequence, each coordinate is represented as a 160 bits word and a point occupies 320 bits / 40 bytes. Given that the maximum size of an IEEE802.15.4 frame is about 127 bytes, in our implementation the `POINTS_SET_RESP` message is sent as separate `POINT_RESP` messages, one per point, in order to avoid relying on 6LoWPAN fragmentation.

We observe that activating 6LoWPAN fragmentation represents a resource-intensive feature to such platform, significantly consuming both RAM and Flash memory space [Sta18]. Consequently, opting to deactivate this feature and instead relying on packet fragmentation at the application layer, achieved by transmitting each point in separate frames, emerges as a more efficient solution. Additionally, to enhance the representation and evaluation of communication costs, we have chosen to deactivate the transmission of signatures associated with each point. This avoids the need to send two separate messages for each point, contributing to a more streamlined communication process.

4.3 Simulation

We conduct our performance evaluation in a fully-controlled environment through simulation. To this end, we used the Cooja network simulator [Ost+06]. It decouples the emulation of mote firmwares from the simulation of radio communications between them. The main benefit of this approach is that the emulator runs the unmodified mote firmware, as if it was run on a real device.

Using Cooja, we can define for a simulation our own topology, parameters and models. The main parameters used are summarized in Table 4.1. Our test network consists in a mesh of 25 motes of type Zolertia Z1 organized in a 5×5 grid. Every node runs our prototype implementation based on Contiki-NG and communicates using IEEE 802.15.4 in TSCH mode. RPL Lite, a stripped-down implementation of RPL is used as routing protocol, in non-storing mode. Our prototype relies on UDP as transport protocol instead of using CoAP as it would have been impossible to add the CoAP library given the limited memory resources of the chosen platform.

The radio propagation model used is the Unit Disk Graph Model (UDGM). Assuming the

Platform and topology	
Mote type	Zolertia Z1 (MSP430)
Number of nodes	25
Topology	5 × 5 Grid
Radio propagation/error model	UDGM
Distance between nodes	$\delta = 1$
Communication/interference ranges	$\delta_c = 1.25$ / $\delta_i = 2.5$
Simulation parameters	
Collect strategies	Global, Local
Number of rounds	10
JRC position	Center, Corner
Network stack	
Physical Layer	IEEE 802.15.4
MAC Layer	TSCH
TSCH Scheduler	6TiSCH
TSCH Slotframe length	1
Routing	RPL Lite (non-storing mode)
Tuned network parameters	
`QUEUEBUF_CONF_NUM`	4
`NETSTACK_MAX_ROUTE_ENTRIES`	25
`UIP_CONF_BUFFER_SIZE`	160
`NBR_TABLE_CONF_MAX_NEIGHBORS`	4
Security	
Library	`micro-ecc`
Polynomial degree m	3
Curve	`Secp160r1`
Size of point coordinates	320 bits

Table 4.1: Parameters of Contiki-NG's 6TiSCH network stack, security layer and simulation model.

distance δ between nodes in the grid is unity, we configured the communication range to $\delta_c = 1.25$ and the interference range to $\delta_i = 2.5$. Even though we did not introduce explicit communication errors in the radio communication model, interferences and collisions are possible and taken into consideration.

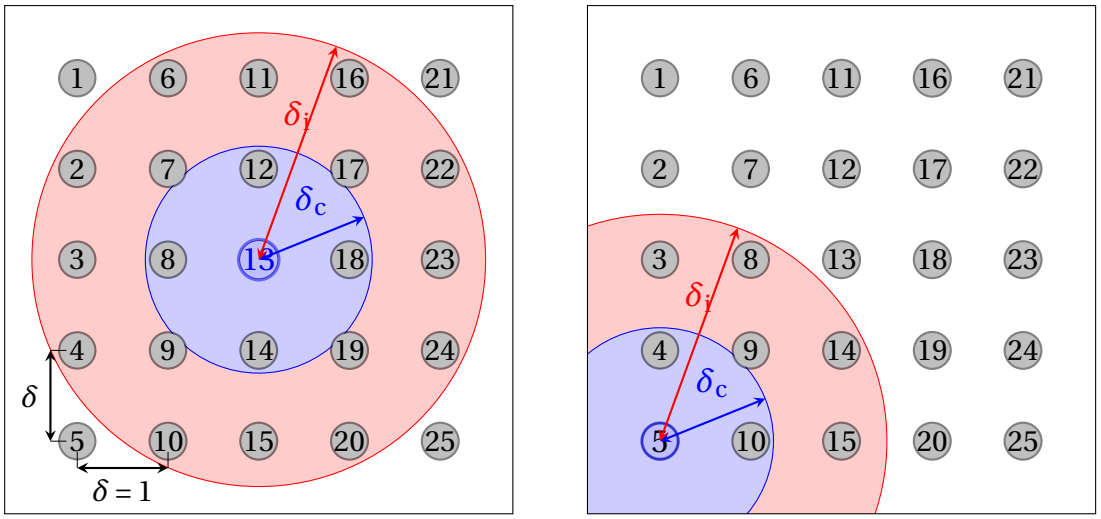
Since we opted for a very constrained mote platform, especially with regards to both Flash and RAM spaces, care was needed when defining the Contiki-NG network stack parameters. To support 6TiSCH operations, some parameters must be reduced on the Z1 platform, as mentioned in the official documentation of Contiki-NG [Duq18]. The use of the

`micro-ecc` library made the memory consumption on the motes even higher, which forced us to limit some of the parameters even further. To be more specific, the number of packets in the link-layer queue, `QUEUEBUF_CONF_NUM`, was limited to 4 and the size of the IPv6 buffer, `UIP_CONF_BUFFER_SIZE`, to 160 bytes. The number of entries in the neighbor table, `NBR_TABLE_CONF_MAX_NEIGHBORS`, was limited to 4. and the number of routing entries, `NETSTACK_MAX_ROUTE_ENTRIES`, was limited to 25.

Those limits motivated the selection of a 5×5 grid topology and the limited radio range. Indeed, this choice limits the size of the neighborhood to 4 and as there are 25 nodes in the network, only 24 entries must be maintained in the JRC/RPL root routing table. We note that these are limitations of the 6TiSCH implementation on the Z1 platform, not of our protocol.

Table 4.2 represents the space consumption of Flash and RAM by our solution on Zolertia Z1 platform (the case of a joining node). The first line represents the space consumed by the RTOS and the communication stack while operating in a network. The Protocol execution represents the memory cost of sending collect requests and handling received packets. Lagrange interpolation is the cost of including micro-ecc library and executing the interpolation 10 times. Cryptography operations are the operations of packets decryption and key establishment at the joining node level (micro-ecc library is already included for Lagrange interpolation, other libraries are included to compliment this part).

We considered different scenarios for our experimental evaluation. First, we evaluated the two collect strategies implemented in our prototype : global and local modes. Second, as the position of the JRC/RPL root in the network affects the average path length, we considered two different placements, as shown in Figure 4.1 : one where the JRC is at the center of the grid (node 13, Figure 4.1a) and another one where it is located in a corner (node 5, Figure 4.1b). For every scenario considered, we ran the simulations 10 times to account for the random selection of JP. Finally, we configured the JRC to wait until the RPL network has converged before starting the security protocol boot phase in order to avoid interference of routing with our evaluation.



(a) JRC at the center

(b) JRC in a corner

Figure 4.1: The 5 × 5 grid topology used in the simulations with two different JRC placements. The blue (resp. red) disk depicts the communication (resp. interference) range. The distances between nodes and the ranges are to scale.

4.4 Evaluation

4.4.1 Communication overhead

This section evaluates the overhead of our protocol in terms of communication. Recall that in Section 3.10.1 we evaluated the number of messages as a function of the polynomial degree m and the number N of Proxy JP. Here, we extend this analysis by accounting for the multi-hop nature of the network. To this end, we quantify the number of data frames required at the link-layer to carry the protocol messages exchanged at the application layer. The number of

Operation	Flash Consumption	RAM Consumption
RTOS+Network Stack	**63665**	**6324**
Protocol Execution	**844**	**220**
Lagrange Interpolation	**4104**	**160**
Cryptography Operations	**5582**	**180**

Table 4.2: Space consumption in bytes of Flash and RAM by our solution on Zolertia Z1 platform

data frames depends on the number of hops traversed by the application layer messages and on possible re-transmissions due to collisions and interferences.

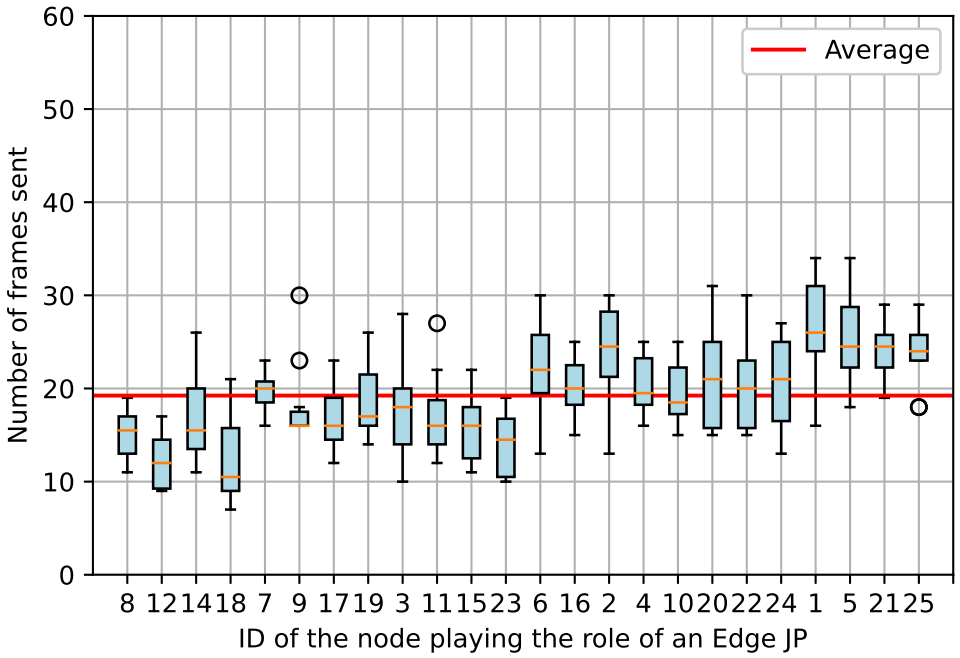
To conduct this evaluation, our simulation scenario assumes that each node in the network acts as a Proxy JP at different times during the simulation, gathering points on behalf of a requesting Pledge. This approach provides a clear representation of the collection phase's cost, reflecting its dependence on the placement of the Proxy JP. Additionally, executing the collection phase with multiple Proxy JPs concurrently on a resource-constrained platform may result in buffer overhead at the JRC level, especially in a global collect mode. In a less constrained environment, without limitations, and where we do not specifically aim to observe for evaluation purposes, this assumption becomes unnecessary for the implementation. We count the number of data frames exchanged during that phase. Given that the contacted JP are picked randomly, we repeat the simulation 10 times for each Proxy JP.

We perform this evaluation for the two collect strategies described in Section 3.8.

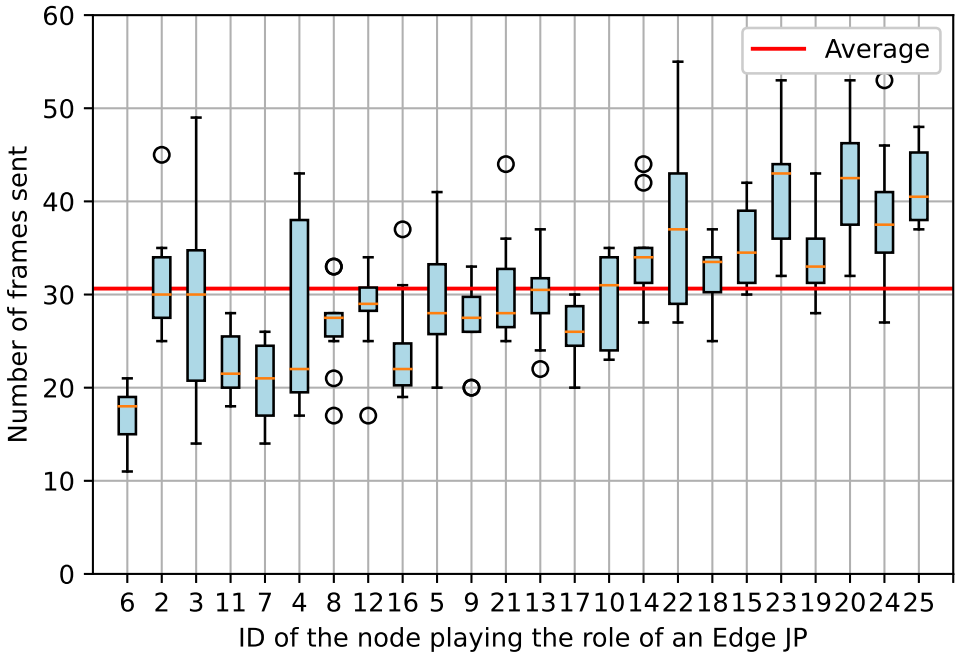
Global mode, JRC at the center

We first consider the *Global mode* in the scenario where the JRC is placed at the center of the grid. Figure 4.2a represents the number of data frames exchanged during the collect phase of each Proxy JP. Each tick on the x-axis corresponds to the ID of the node acting as Proxy JP. The y-axis provides the number of frames exchanged. We used box plots to show the variability of this metric over 10 simulation runs. The nodes on the x-axis are ordered according to increasing distance from the JRC. The Manhattan distance was used as the nodes can only communicate at the data-link layer with their direct horizontal and vertical neighbors as a combined consequence of the grid structure of the topology and the limited communication range. We can indeed observe a trend where the nodes closer to the JRC require less frame exchanges during the global collect strategy than the nodes that are further away. In addition to this, we observe that on average, collecting $m = 3$ points requires 19 frames.

The number of frames could be calculated as shown below in Equation (4.1), where $E[\|P_{\text{JRC}\leftrightarrow\text{JP}}\|$ is the expected distance between a node and the JRC. The 1st part of the formula corresponds to the request for $m-1$ nodes sent from the Proxy JP to the JRC, the second part to the request



(a) JRC at the center



(b) JRC in a corner

Figure 4.2: Number of frames sent during a global collect phase when each node is playing the role of a Proxy JP separately. The box plots summarize the distribution of the number of frames obtained over 10 runs of the simulation.

sent by the JRC to $m-1$ JP and the corresponding responses and the 3rd part to the forwarding of the $m-1$ responses from the JRC to the Proxy JP.

$$\begin{aligned} E[N_{\text{frames}}] &= E[\|p_{\text{JP}\leftrightarrow\text{JRC}}\|] \\ &\quad + 2\times(m-1)\times E[\|p_{\text{JRC}\leftrightarrow\text{JP}}\|] \\ &\quad + (m-1)\times E[\|p_{\text{JP}\leftrightarrow\text{JRC}}\|] \\ &= (3m-2)\times E[\|p_{\text{JP}\leftrightarrow\text{JRC}}\|] \end{aligned} \tag{4.1}$$

Global mode, JRC in a corner

In order to evaluate the impact of the JRC position on the communication overhead, we conducted a second series of simulation where the JRC is located in a corner of the grid. Figure 4.2a represents the number of frames sent during the collect phase, for each Proxy JP. The nodes ID are again ordered by increasing Manhattan distance from the JRC. The results show that positioning the JRC at the corner of the network instead of the center requires in average more frames (30 versus 19) since many nodes are further away from the JRC.

The average hop count in a 5×5 grid with central JRC is 2.5 while it becomes 4.167 when the JRC is in a corner. Using Equation (4.1), this means the expected frame count is 17.5 with a central JRC and 29.167 with a JRC in a corner. This seems to be in-line with the simulation results.

Local mode

We finally consider the *Local mode*. Here, the JRC placement does not matter as the messages are exchanged only with directly connected nodes, without using the routing tree. The number of frames sent per collect phase for each Proxy JP are presented in Figure 4.3. We observe that the local mode is much more efficient than the global one in term of communication with an average of 4 frames sent per collect phase and per Proxy JP. Moreover this value has very low variance and is independent of the position of the Proxy JP relative to the JRC.

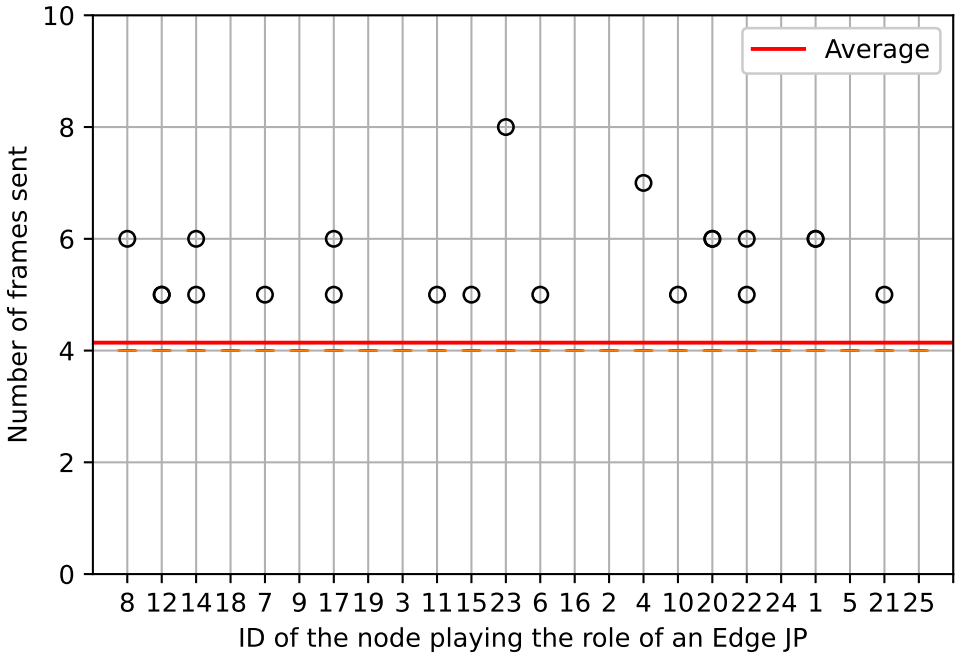


Figure 4.3: Number of frames sent during the *Local* collect phase for each node playing the role of a Proxy JP separately.

4.4.2 Latency and energy consumption

This section evaluates the collect phase of our protocol implementation from the point of view of latency and energy consumption. This is done for both global and local strategies. More specifically in *Global mode* we are interested for each Proxy JP by the interval of time between the transmission of the first `POINTS_SET_REQ` message and the reception of the last corresponding `POINT_RESP` message. In *Local mode*, we consider the interval between the first `POINT_REQ` and the last `POINT_RESP` messages.

To estimate the energy spent during that interval of time, we integrated the Energest [Dun+07] module in the firmware of our implementation. It is a lightweight, software-based energy estimation solution that tracks the time spent by different hardware components such as the CPU and radio transceiver in various power states.

For the radio transceiver, it distinguishes three states : listening, transmitting and OFF. For the CPU it counts four states: ON, Low Power Mode (LPM) and OFF. The estimation of the current

draw of the Zolertia Z1 mote in each of these states, according to its datasheet, is presented in Table 4.3.

The total energy consumed during the interval of interest is calculated by Equation (4.2) where $U = 3\,\text{V}$ is the Zolertia Z1 power supply voltage, PS is the set of Energest power states, I_s is the current draw in state s and T_s is the total time spent in state s.

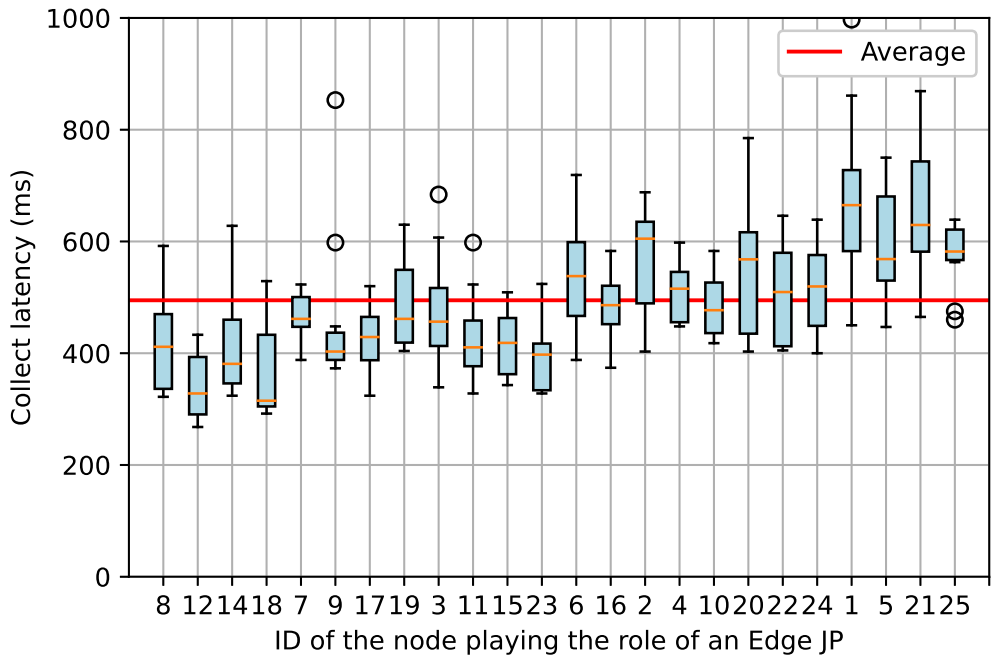
$$E = \sum_{s \in \text{PS}} U \times I_s \times T_s \tag{4.2}$$

For the *Global* collect strategy with the JRC in the center of the grid, the measured latency and energy consumption, separately for each node playing the role of a Proxy JP, are reported in Figures 4.4a and 4.5a, respectively. For the *Local* mode, the time and energy measurement results are reported in Figures 4.4b and 4.5b respectively. Comparing these results together, it is immediately apparent that the *Local* mode is much faster and energy efficient than the *Global* mode. The average latency to collect a set of $m = 3$ points is 180 ms with the *Local* strategy against 500 ms with the *Global* strategy. The same conclusion can be drawn for the energy with the *Local* mode requiring 2.05 mJ on average to collect $m = 3$ points versus 5 mJ for the *Global* mode. To put these results in perspective, let's consider that a node is powered by a pack of 3 standard 2000 mAh NiMh batteries, providing 25.92 kJ. Each collect represents only between 7×10^{-6} % and 1.9×10^{-5} % of the battery capacity.

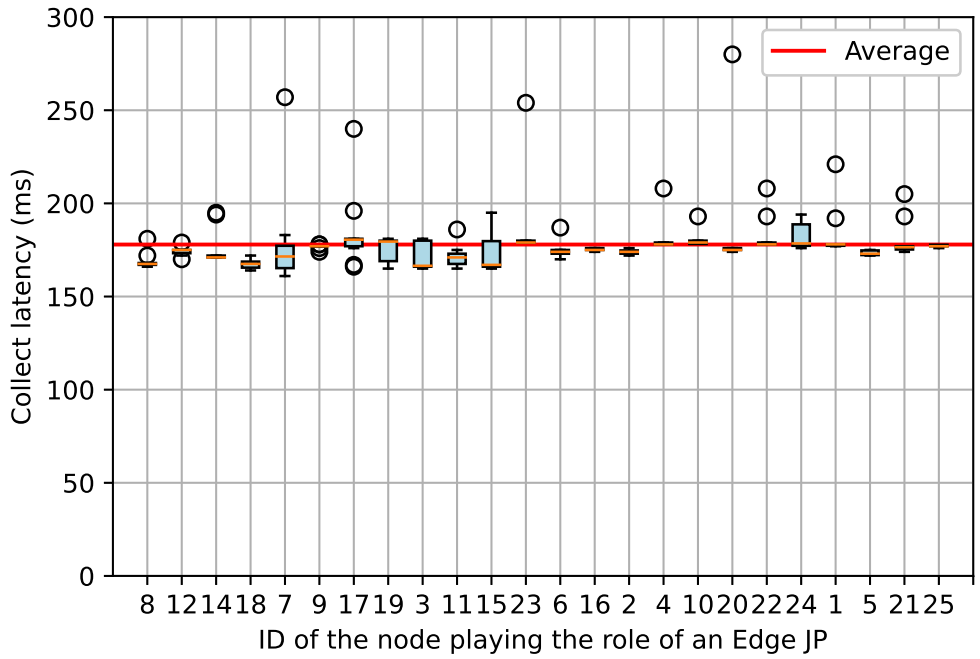
We note that in this section we only consider the energy spent on the communication part of our protocol. The evaluation of the computing complexity considering the interpolation cost at the pledge level and the encryption operations are considered in separate sections.

Power states (PS)	**Current consumption**
Radio Listening Rx	18.8 mA
Radio Transmitting Tx	17.4 mA
CPU active	0.5 mA
CPU LPM	0.5 μA

Table 4.3: Energest power states and corresponding current draw.

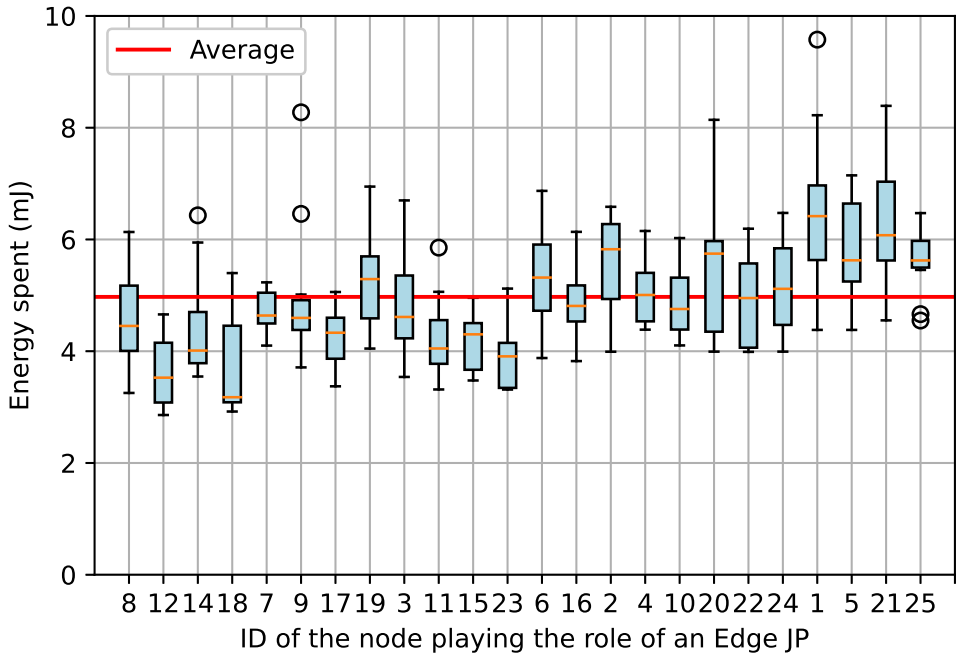


(a) Global collect

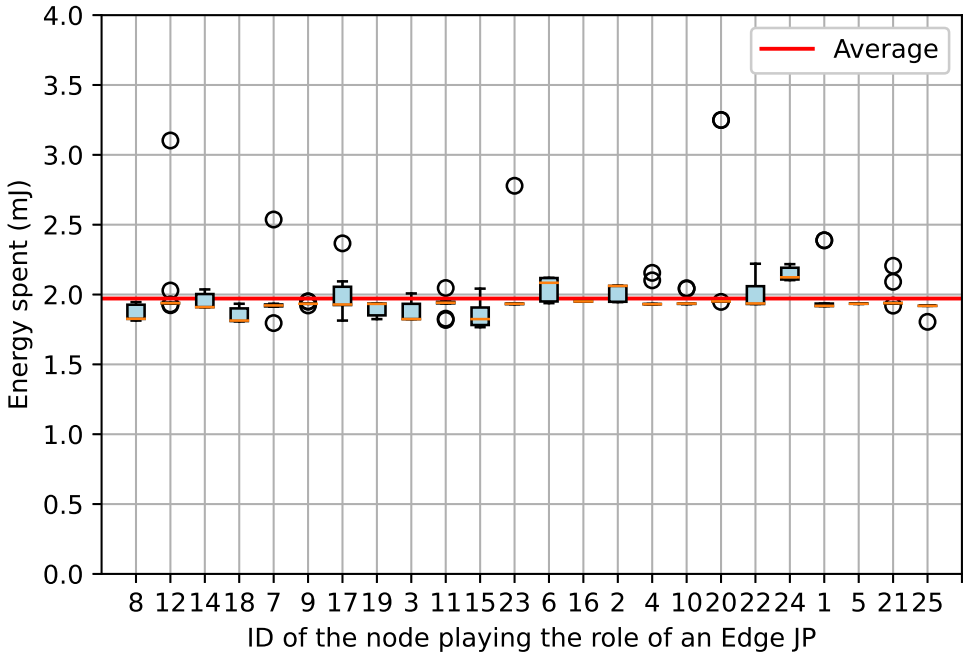


(b) Local collect

Figure 4.4: Time spent during the collect phase for each node playing the role of a Proxy JP separately (ms).



(a) Global collect



(b) Local collect

Figure 4.5: Energy spent during the collect phase for each node playing the role of a Proxy JP separately (mJ)

4.4.3 Interpolation cost

In this section we evaluate the cost incurred by a pledge to perform Lagrange polynomial interpolation described in section 3.2.1. As described in section 3.6, once a pledge has received multiple packets of points from different Proxy JPs, it attempts at recovering the secret key through interpolation. It does so using points from pairs of Proxy JPs. The pledge has to do all/many combination of packets from different Proxy JP. Then on each combination, Lagrange interpolation will be executed on a randomly selected subset of $m+1$ points.

To focus on the cost of the sole interpolation step, we run it on a simulated node and measure the duration of computing the interpolation as well as the resulting amount of energy consumed, using Energest, similarly to Section 4.4.2.

We note that Lagrange Interpolation is based on modular operations in Z_p executed on the collected points where the coordinates of these points are generated from Q(x) as described in section 3.4.

Table 4.4 provides the energy consumed due only to the computing (CPU Active + LPM) for each number of Lagrange interpolation executed on the pledge. We see that one interpolation costs around 1.57 mJ. The cost of one interpolation is not so high, however, we see that having a large number of interpolations makes the total cost considerable. Therefore, it is better to stay on a limited number of interpolations in order to reduce the cost of this phase.

	Number of interpolations					
	1	**3**	**5**	**10**	**20**	**50**
Duration of usage of the CPU (ms)	1049	3147	5245	10490	20981	52454
Energy consumed (mJ)	1.57	4.72	7.87	15.74	31.47	78.68

Table 4.4: Energy consumption on the pledge level

Hence, to reduce the overhead of the complexity for this phase, a pledge can stop the interpolation calculations after a limited number X of iterations. Therefore, X can be calculated as follows: according to Section 3.10, a pledge being in contact with N Proxy JP will have a

total number of combinations equal to C_N^2. From this C_N^2, $C_{\frac{2\cdot N}{3}}^2$ are the number of combinations between two honest nodes and $C_{\frac{N}{3}}^2$ are the number of the combinations between two malicious ones. The rest of the combinations are between an honest node and a malicious one. Hence, in the most extreme type of collaborative attack, we have two types of results that may be repeated, either the $C_{\frac{N}{3}}^2$ malicious ones, or the $C_{\frac{2\cdot N}{3}}^2$ correct ones. Therefore, once the pledge reaches a number of iterations X where it already exceeded $C_{\frac{N}{3}}^2$ similar results, it can stop since only the honest calculation can exceed this value. For example, for a number of Proxy JPs equal 5, the total number of combinations C_N^2 is 10, the number of combinations between two honest nodes $C_{\frac{2\cdot N}{3}}^2$ is 3 and the number of combinations between two malicious nodes $C_{\frac{N}{3}}^2$ is 1. Therefore, in this case, the pledge can stop the interpolations once it reaches a number of iterations $X > 1$ (X=2) where the results are similar.

4.4.4 Encryption Cost

In this section we evaluate multiple encryption operations executed in our protocol on multiple type of nodes. The results are represented in Table 4.5. Let's first consider the Proxy JP. As described in section 3.6, a Proxy JP receiving points from multiple nodes in the network (m-1 nodes), must verify the signature of each, using Pk_{JRC}, in order to check the integrity of the collected data. Elliptic curve digital signature algorithm ECDSA is used for this signature costing 22.15 mJ (line 1 of Table 4.5). Moreover, a hybrid encryption (Elliptic curve El Gamal + AES-128) is used to encrypt the packet of points sent between each Proxy JP and the pledge. This avoids any modification on the packet being sent over a public channel between the Proxy JP and the pledge. El Gamal is used to encrypt a generated session key with Pk_{pledge} costing 19.14 mJ (line 2 of Table 4.5). The cost of each algorithm operation is presented in the same row. It consists of two Elliptic Curve multiplications and one Elliptic Curve point addition. The sent data are encrypted with this session key using AES-128 costing 0.02 mJ (line 3 of Table 4.5).

At the pledge level, a decryption of the packets is executed at first. A hybrid decryption (Elliptic curve El Gamal + AES-128) is used as well. El Gamal decryption algorithm is used consisting of one Elliptic Curve multiplication and one Elliptic Curve addition. This decrypts the session key that will then be used to decrypt the data. Then, after the reveal of the secret, a session

Node type	Operation	CPU usage duration	Energy consumed
Proxy JP	Signatures verifications	14 771 ms	22.15 mJ
	El Gamal Encryption:	12 760 ms	19.14 mJ
	* EC Multiplication (1)	6350 ms	9.525 mJ
	* EC Multiplication (2)	6348 ms	9.522 mJ
	* EC Addition	62 ms	0.093 mJ
	AES-128 Encryption	12 ms	0.02 mJ
	Total	27543 ms	41.31 mJ
Pledge	EL Gamal Decryption:	6420 ms	9.63 mJ
	* EC Multiplication	6347 ms	9.5205 mJ
	* EC Addition	72 ms	0.108 mJ
	AES-128 Decryption	12 ms	0.02 mJ
	Key Establishment:	12 768 ms	19.15 mJ
	* EC Multiplication (1)	6351 ms	9.5265 mJ
	* EC Multiplication (2)	6347 ms	9.5205 mJ
	* EC Addition	66 ms	0.099 mJ
	Challenge Encryption	3 ms	0.005 mJ
	Signing	6567 ms	9.85 mJ
	Total	25770 ms	38.65 mJ

Table 4.5: Energy consumed for cryptography operations on each node type.

key is established as described in section 3.7. This costs 19.14 mJ. In the end, a challenge is generated and encrypted with the session key using AES-128. The whole parameters are signed with Sk_{pledge} using ECDSA before being sent costing 9.85 mJ. Overall, we see that the signatures (signing and verification) and the key establishment are the most consuming operations. Moreover, comparing the cryptography costs between a Proxy JP and the pledge, we see that they are somehow equal (41.31 mJ for the Proxy JP and 38.65 mJ for the pledge). The cryptography operations executed at the JRC level were not considered in this evaluation since we consider that the JRC is not a constrained device.

We note that in our implementation we used the Zolertia Z1 device, considered from the most constrained devices unsuitable for cryptography operations [Bau+16].

4.4.5 Total consumption

In this section we aim to provide an overview of the total energy consumed for the whole joining phase by the main nodes participating in this task, that is the Pledge and Proxy JPs. We

consider all the phases from the initial request till the key establishment.

We assume a network where a Pledge sends requests to 5 Proxy JPs that each respond with a packet of points. According to the results presented in Figures 3.5a and 3.5b, the average success rate for this number of Proxy JP is around 80% for the worst case where one third of the network nodes are malicious. The total number of interpolations for all possible combinations is equal to 10. For the collect phase, we used the mean value extracted from Figures 4.5a and 4.5b to calculate the energy spent by 5 Proxy JPs in both the *Global* and the *Local* modes. We also presented the mean value for the number of frames sent, extracted from Figures 4.2a and 4.3. Table 4.6 represents the total cost of communication for such a joining phase, comparing the global mode and the local mode. We see that the results validate the efficiency of the local mode over the global mode.

Node type	Operation	Global mode	Local mode
5 Proxy JPs	Points collect	25 mJ *(≈100 frames)*	10.25 mJ *(≈20 frames)*
	Message exchanges with Pledge	0.6 mJ *(10 frames)*	0.6 mJ *(10 frames)*
Pledge	Message exchanges with Proxy JP	0.91 mJ *(5 frames)*	0.91 mJ *(5 frames)*
All	**Total**	26.51 mJ *(≈115 frames)*	11.76 mJ *(≈35 frames)*

Table 4.6: Comparison of the total energy consumed for communication.

Table 4.7 represents the total cost of computation for the executed operations in such scenario. We consider in this evaluation Lagrange interpolations at the pledge and the cryptography operations at both the Proxy JPs and the Pledge.

Operation	Energy consumed
Cryptography operations on 5 Proxy JPs	206.57 mJ
Lagrange Interpolation on the Pledge	15.74 mJ
Cryptography operations on the Pledge	38.65 mJ
Total	260.95 mJ

Table 4.7: Comparison of the total energy consumed for computation.

Tables 4.6 and 4.7 provide an overview of the energy consumed in a network for an average joining phase. We observe that the cryptography cost is considerably higher than the com-

munication cost (260.95 mJ versus 11.76 mJ for the local mode and 26.51 mJ for the global mode).

4.4.6 Comparison with CoJP

We compare our solution to the default authentication mechanism adopted in 6TiSCH. As mentioned in section 2.1.2, the joining phase of 6TiSCH is called CoJP and based on a simple Request-Response between the pledge and the JRC going through a multi-hop of intermediate nodes. This exchange is executed at the application layer using CoAP and is secured by an End-to-End security protocol called OSCORE. OSCORE is based on a pre-shared key that they claim to be already shared and relies on AES-128 symmetric encryption. In CoJP, they do not consider a mutual authentication phase since it is assumed that a key is pre-shared between the pledge and the JRC. Hence, they consider only a pre-shared key validation phase, executed at the same time with the key establishment phase. However, in our proposed solution we ensure both schemes: mutual authentication and key establishment. We utilize in this evaluation the same scenario of 5 Proxy JPs adopted in section 4.4.5. For the mutual authentication phase, since it is not executed in CoJP, there is no way to compare its cost with the same phase of our protocol. However, for the cost of our protocol in this phase we refer to section 4.4.5 . For the key establishment phase of each protocol, we represent the energy consumed for computing and communication tasks, by all the nodes participating in the joining phase (Proxy JPs and joining node). The comparison is presented in Table 4.8

Our solution is more energy consuming than a symmetry based authentication protocol since it ensures a complete mutual authentication and key establishment phase based on zero-configuration. However, it is more autonomous and suitable for large scale and dynamic networks where the configuration of each arriving node is a significant issue.

Phase	Aspect	Our Solution	CoJP
Mutual Authentication	Computing	Consensus	PSK
	Communication	+ Certificate	
Key Establishment	Computing	260.95 mJ	0.006 mJ
	Communication	1.36 mJ	1.36 mJ

Table 4.8: Comparison between our proposed solution and CoJP, for the total energy consumed in the network for a joining phase

4.5 Possible Improvements

We showed in this chapter that, even though it requires additional message exchanges, increasing the latency and energy consumption incurred by the nodes, applying our proposal to 6TiSCH is a practical way to ensure mutual authentication without relying on pre-shared keys.

Moreover, in our experiments we adopted the Zolertia Z1 mote. The constrained memory of this mote obligated the limitation of the network size and number of neighbours per node as mentioned in Section 4.3. This type of node is considered one of the most constrained devices (Class 0 according to RFC 7228 [BEK14]). Therefore, in an industrial environment where nodes are supposed to be less constrained, the topology of the targeted IoT network can scaled up and the processing time and energy consumed for cryptography operations will be more limited. Moreover, hardware accelerations supporting cryptography operations, including ECC, tends to be more common even in low-power microcontrollers [Kie+21]. Such support would also help in reducing the time and energy spent in performing these operations.

4.6 Summary

In this chapter, we have conducted a comprehensive evaluation of our security protocol within the context of 6TiSCH, focusing on communication costs and cryptographic operations. We have implemented our scheme using the Contiki-NG operating system, which provides a robust 6TiSCH network stack. To reflect real-world IoT device constraints, we utilized the Zolertia Z1 platform. Our performance evaluation took place in a controlled environment through simulation using the Cooja network simulator.

We have assessed our solution in both Global and Local mode scenarios: one with the JRC at the grid's center and another with the JRC in a corner. We have compared our approach to the default authentication mechanism in 6TiSCH, noting that our method incorporates mutual authentication and key establishment, unlike CoJP, which relies on pre-shared keys.

Our findings indicate that, despite the additional message exchanges that may slightly increase latency and energy consumption, implementing our proposal in 6TiSCH offers a practical means of ensuring mutual authentication without the need for pre-shared keys.

By focusing on resolving the security issue during the joining phase, the detection of proxy nodes has gained significant importance. In the following chapter, we will delve into the detection system for malicious proxy nodes within the context of the joining phase.

5 Detecting Malicious Proxy Nodes during IoT Network Joining Phase

In chapter 3 we presented an authentication solution aiming to secure the joining phase in IoT networks. Our solution consists of a consensus approach, where multiple Proxy nodes play intermediary roles between the joining node and the network. However, this introduces a vulnerability to potential attacks orchestrated by these Proxy nodes, as there is no way for the joining node to independently verify the legitimacy of the information provided by them. Consequently, malicious Proxy nodes could potentially misdirect a joining node to a different network. Thus, within the context of the joining phase in IoT networks, the detection and mitigation of malicious nodes emerge as a critical concern. In this chapter, we present a solution to detect malicious proxy nodes in the joining phase of an IoT network. For the rest of this chapter, refer to Table 5.1 for details on the parameters used.

5.1 Our Architecture

In this section, we propose a detection system solution for large-scale and dynamic IoT networks. Figure 5.1 represents an architecture of our target system. We consider the following architecture in our system:

- **Coordinator:** It is responsible to control the network's functions and maintain its security. It manages the network communications, devices connectivity and data flows depending on the implemented application. It receives, through join proxies, the Join

Table 5.1: Table of notations

Notation	Description
Z_p	Finite field of order p
Crd	Coordinator
$Node_i$	A random node in the network
k	A joining phase of order k
S_{E_k}	Set of the N Proxy nodes in contact with a joining node in joining phase k
R_{E_k}	Set of Proxy nodes reported the joining node after a joining phase k
E_{Node_i}	a Proxy node in contact with a joining node
$Sk_{NewNode}$	Private key of joining node
$Pk_{NewNode}$	Public key of joining node
Ntw	Network
NR_i	Number of reports against a $Node_i$
NP_i	Number of participation of a $Node_i$
NH_i	Number of honest participation of a $Node_i$
TX_i	$NH_i \div NP_i$
Pun_i	Boolean parameter indicating if a$Node_i$ is punished or allowed to participate in further joining phases.
$T1$	Threshold T1
$T2$	Threshold T2
LG	Log table
I_{OF}	Intensity of ON-OFF attack
I_{FP}	Intensity of False Positive attack
I_{FN}	Intensity of False Negative attack
TN	Total number of nodes in the network
TM	Total number of malicious nodes in the network
TH	Total number of honest nodes in the network
λ	Rate of malicious nodes in the network

Requests of new devices and is responsible of giving the permissions to access the network. Until receiving a request, the coordinator has no knowledge about the Join Requests handled by the join proxies. We note that the coordinator is a fully trusted entity.

- **Nodes:** Are all the devices connected to this coordinator through direct or indirect links. Therefore, we consider a mesh topology where multi-hop connectivity is possible.

- **Proxy node:** It is a node in the network in a direct link with a joining node. It is a join

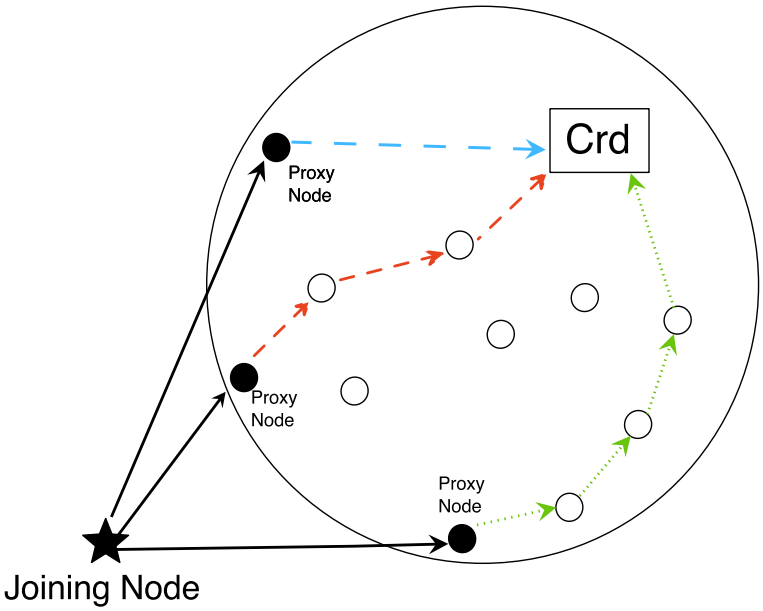


Figure 5.1: Architecture of our detection system.

proxy node and plays the role of intermediate between a new node and the coordinator during the joining phase. Proxy node sends advertisement packets and receive Join Requests from new devices. We consider the cases where one or multiple Proxy nodes are participating in the same joining phase. It can not be fully trusted since it can act maliciously during the joining phase.

- **Joining node:** The new node sending its request to one or multiple Proxy nodes in order to join the network, after receiving advertisements from each of them. This joining node can not be fully trusted since it can be a malicious node.

5.2 Proposed Solution

5.2.1 Trust Model

In this section, we propose a solution for the coordinator of a network, to detect the malicious Proxy nodes in its network. This detection mechanism is considering multiple factors:

- **Number of reports NR_i:** is a whole number with minimum value 0. It represents the number of times $Node_i$ has been reported to the Crd as malicious participant

in a joining phase. We note that the joining node can only report a Proxy node if its authentication succeeded.

- **Number of participation NP_i:** is a whole number with minimum value 0. It represents the number of times a $Node_i$ participated as a Proxy node in a joining phase.

- **Number of honest participation NH_i:** is a whole number with minimum value 0. It represents the number of times a $Node_i$ participated as a Proxy node in a joining phase without that the joining node reports it to the Crd.

All these parameters are stored at the Crd level, and are updated after each joining phase.

5.2.2 Main Idea

We assume that a joining node must be able to detect the malicious Proxy nodes that participated in its joining phase. This depends on the authentication mechanism operating. In Section 5.2.4, we elaborate our detection mechanism proposed for the consensus-based authentication scheme in 6TiSCH protocol. After a successful joining phase, the joining node has a secure end-to-end communication established with the coordinator. Therefore, the joining node reports the detected malicious nodes for the coordinator of the network. At the coordinator level, a mechanism is established in order to classify these reported nodes as malicious, or consider that as a false report.

This mechanism is based on comparing the Proxy node parameters with two thresholds T1, and T2. T1 is used to count the number of reports considering a Proxy node until it reaches a set threshold. Its purpose is to detect an excessive number of reports against a Proxy node before investigating its behaviour. T2 is used to deal with possible attacks on the detection system and distinguish between malicious Proxy nodes conducting an ON-OFF attack, or consider these reports as False Positive attacks against it. Finally, if the coordinator classifies a Proxy node as malicious, it punishes this Proxy node by preventing it from participating again in further joining phases.

5.2.3 Setup phase

During the network bootstrapping process, the Crd performs the following sequential steps to prepare the parameters utilized by our protocol:

- The Crd maintains a log table LG, in order to record the log of each joining phase in the network. For any joining phase k, each log contains the following information:
  - The public key $PK_{NewNode}$ of the joining node. This public key is derived from the certificate received in the $JoinRequest$.
  - A set $S_{E_k} = \{E_{Node_1}, E_{Node_2}, ..., E_{Node_N}\}$ of the N Proxy nodes that forwarded the $JoinRequest$ of the joining node to the Crd in the joining phase k.
  - A set $R_{E_k} \subset S_{E_k}$, the subset of Proxy nodes reported as malicious by the joining node to the Crd after the joining phase k.
- The Crd maintains a table of scores Sc, in order to save several parameters for each node in the network. This table contains the following parameters for each $Node_i$:
  - An identifier i of each $Node_i$ in the network.
  - The number of participations NP_i counting how many times $Node_i$ participated in a joining phase.
  - The number of reports NR_i indicating how many times $Node_i$ was reported as malicious during its previous joining phases.
  - The number of honest participations NH_i counting how many times $Node_i$ participated in a joining phase without being reported.
  - The metric TX_i=NH_i/NP_i
  - The flag Pun_i indicating if a $Node_i$ is punished or allowed to participate in further joining phases.

5.2.4 Report of malicious nodes

During a joining phase k, a joining node sends a $JoinRequest$ to all the N Proxy nodes it received network advertisement packets from. The joining node is able to detect the mali-

cious Proxy nodes after the joining phase. Different solutions can be implemented at the joining node level in order to identify the malicious nodes, depending on the authentication mechanism operating and the type of attack conducted by a malicious Proxy node. In the case of a selective forwarding attack, a malicious Proxy node receiving a $JoinRequest$ will drop it instead of forwarding it to the coordinator. Therefore, labeling a Proxy node without a $JoinResponse$ as malicious provides a straightforward approach to identifying potential malicious Proxy nodes at the joining node level.

In the case of our authentication mechanism based on consensus proposed for 6TiSCH (see Section 3), a consensus of multiple nodes in the network is accomplished in order to redirect the joining node to the correct coordinator of the network. Receiving multiple packets coming from multiple Proxy nodes, these packets are combined at the joining node level to form all possible pairs of combinations. The joining node employs Lagrange polynomial interpolation on each combination to reconstruct the coordinator-defined secret polynomial. Having a majority of honest Proxy nodes between these nodes will lead to both the identification of the correct coordinator of the network and the detection of the malicious nodes. The malicious nodes are the ones sending packets containing fake information in order to falsify the consensus or to lead the joining node to a malicious coordinator. The following process is executed at joining node level in order to detect the malicious Proxy nodes:

- $index$ is a set of unordered couples, where each combination of two Proxy nodes E_{Node_p} and E_{Node_q} are saved along with $q(0)$, their corresponding result of Lagrange interpolation.
- After all the interpolations for all the combinations are executed, one result $Q(0)$ is the most frequent interpolation result.
- $\forall E_{Node_p}, E_{Node_q} \in index$ where $q(0) = Q(0)$, E_{Node_p} and E_{Node_q} are considered as honest Proxy nodes and added to the set GE of honest Proxy nodes.
- $\forall E_{Node_i} \in S_{E_k} \wedge E_{Node_i} \notin GE$, we add E_{Node_i} to R_{E_k}, the set of malicious Proxy nodes for the joining phase k.
- Upon a successful authentication, the joining node sends the set R_{E_k} to the Crd

5.2.5 Malicious nodes detection

At the end of each joining phase k, the pledge establishes an end-to-end communication with the Crd of the network. Therefore, upon the reception of R_{E_k} the following steps are executed by the Crd:

- The Crd increments the number of participations of the N Proxy nodes in the joining phase k, as follows:
  $\forall Node_i \in Ntw$, where Ntw represents the network,
  $$NP_{i_k} = \begin{cases} 1 & \text{if } Node_i \in S_{E_k} \\ 0 & \text{Otherwise} \end{cases}$$
- The Crd increments the number of honest participations of the non-reported Proxy nodes in the joining phase k, as follows:
  $\forall Node_i \in Ntw$
  $$NH_{i_k} = \begin{cases} 1 & \text{if } Node_i \in S_{E_k} \wedge Node_i \notin R_{E_k} \\ 0 & \text{Otherwise} \end{cases}$$
- The Crd updates the table Sc by calculating for each $Node_i$ the following parameters:
  - The number of participations is the sum of participations in the previous joining phases $NP_i = \Sigma NP_{i_k}$
  - The number of honest participations is the sum of honest participations in the previous joining phases $NH_i = \Sigma NH_{i_k}$
  - The number of reports can be represented as follows: NR_i=NP_i-NH_i
  - The report of honest participations over total participations can be calculated as follows:
    TX_i=NH_i/NP_i
  - The $Node_i$ is punished by the Crd if the value of its parameters exceed the system thresholds as follows:
    $$Pun_i = \begin{cases} 1 & \text{if } NR_i \geq T1 \wedge TX_i < T2 \\ 0 & \text{Otherwise} \end{cases}$$

Therefore, at the end of a joining phase k, this mechanism allows the Crd to consider a malicious $Node_i$ if Pun_i=1. This node must be punished and excluded from further joining

phases. The punishment mechanism is to be defined according to each operating protocol. In Section 5.2.6, we propose a punishment mechanism for 6TiSCH protocol.

5.2.6 Proxy node punishment

After detecting a Proxy node as malicious, the coordinator must take actions in order to reduce its impact on further joining phases. Therefore, it must prevent it from playing the role of a join proxy, while keeping its role as a node in the network. This punishment mechanism can be implemented depending on the operating communication protocol.

In the case of 6TiSCH protocol, this can happen by increasing the value of the $proxypriority$ parameter contained in the Enhanced Beacon EB sent by this join proxy node to the joining node [DR21]. Hence, a joining node receiving such EB will not consider sending it a $JoinRequest$.

However, according to 6TiSCH IETF RFC [Vuč+21b], the joining node takes the packets received from a Proxy node as legitimate without any security verification. Therefore, a joining node can not verify the legitimacy of all the information received in the EB; a malicious Proxy node may set its $proxypriority$ value lower than what it must be. Hence, in order to prevent a punished or excluded malicious Proxy node from disturbing a joining phase by sending fake EB, the following steps are executed:

- As explained in chapter 3, in our authentication protocol proposed for 6TiSCH, once receiving an EB, a joining node sends a $JoinRequest$ to the Crd through multiple Proxy nodes, containing its certificate.
- The Crd receiving a $JoinRequest$, generates a symmetric key ks and sends in a $JoinResponse$ the following parameters:
  - $Enc(ks; Pk_{NewNode})$, the encryption of ks with the public key of the joining node $Pk_{NewNode}$ retrieved from its certificate. $Enc()$ is an asymmetric encryption algorithm.
  - $HMAC(ks; networkID, panID, proxyID,$ $proxypriority)$, where $HMAC()$ is a hash-based message authentication code

algorithm, ks is the key for hashing and the hashed content is $networkID$, $panID$, $proxyID$ and $proxypriority$, four parameters of the network already sent in the EB.

- For each received $JoinResponse$, the joining node retrieves $ks' = Dec(ks; Sk_{NewNode})$ where $Dec()$ is an asymmetric decryption algorithm and $Sk_{NewNode}$ is the private key of the joining node.
- The joining node calculates the $HMAC'$ of $networkID$, $panID$, $proxyID$ and $proxypriority$ received in each EB using the retrieved symmetric key ks'.
- The joining node checks if $HMAC = HMAC'$ for each received EB and $JoinResponse$.
- The joining node considers the EB having their $HMAC$ matching with its corresponding $JoinReponse$ by getting the same hash while using ks'.
- In the case where multiple ks' retrieved, the joining node considers the majority Proxy nodes having the same ks'.

These steps allow the joining node to filter the received EB. In the case where a punished or excluded Proxy node is sending fake EB to falsify the authentication process, this will be detected by this filter since these EB will be unique compared to the ones coming from legitimate Proxy nodes having the same information.

5.3 Attack Models

We here present the attacks that we consider on our detection system. We may have attacks conducted by two different entities: The malicious Proxy nodes trying to falsify the detection system, and the joining nodes that may be malicious and reporting honest nodes or not reporting detected malicious ones. These different attacks are classified as follows:

- **ON-OFF attack**: A malicious node may act maliciously sometimes and honestly other times, in order to falsify the detection system. In this way, it avoids the repetitive report at the coordinator level in order to keep its number of reports value $NR_i < T1$. Moreover, it acts honestly in order to keep the value $TX_i \geq T2$.

- **False Positive attack**: A joining node that just joined the network has to report for the coordinator the set of malicious Proxy nodes R_{E_k}. A malicious joining node may report in this case honest nodes in order to increase their value NR_i. This also decreases their value TX_i.

- **False Negative attack**: Same as the False Positive attack, but this time the malicious joining node does not report a malicious node in order to keep its value NR_i low and avoid its detection. This also keeps their value TX_i as high as possible.

Other type of attacks are considered in our solution. However, they do not really impact our detection system regarding the countermeasures already implemented. For example, the Self Promoting attack and the Ballots attacks [Kou+20] do not impact our detection system since we do not use any reputation value where each node must give its feedback after a service.

5.4 Security Analysis

In this Section, we study the effectiveness of our solution in detecting malicious nodes. First, we present a theoretical analysis of the detection of a specific malicious node $Node_i$, by the coordinator, after a number x of joining phases in the case where no attack is targeting our detection system. After that, we extend our analysis by considering attack scenarios (namely, ON-OFF and False Positive attacks). In this solution, we assume that the node's position is not taken into account during the joining phases. Consequently, the selection of a Proxy node by a joining node is purely random, and therefore, the rate of malicious and honest nodes among these Proxy nodes is entirely random.

Definition 1. *We define* $\mathrm{NR_{ix}}$ *the number of reports for a specific node* $Node_i$*, after a number* x *of joining phases in a network.*

Definition 2. *We define* $\mathrm{PM_i(k)}$*, the probability of reporting a specific malicious node* $Node_i$*, in one random joining phase* k*.*

Definition 3. *We define* $\mathrm{PM_{ix}}$*, the probability of detecting a specific malicious node* $Node_i$*, after a number of joining phases x in a network.*

Definition 4. *We define* $\mathrm{PH_i(k)}$*, the probability of reporting a specific honest node* $Node_i$*, in one random joining phase k.*

Definition 5. *We define* $\mathrm{PH_{i}}_x$*, the probability of detecting a specific honest node* $Node_i$*, after a number of joining phases x in a network.*

Definition 6. *We define* λ*, the rate of malicious nodes in the network.*

Definition 7. *We define* TN*, the total number of nodes in the network.*

Definition 8. *We define* $\mathrm{TM} = \mathrm{TN} \cdot \lambda$*, the total number of malicious nodes in the network.*

Definition 9. *We define* $\mathrm{TH} = \mathrm{TN} - \mathrm{TM}$*, the total number of honest nodes in the network.*

Definition 10. *We define* $Ntw = \{Node_1, Node_2, ..., Node_{TN}\}$ *the set of all the nodes in the network.*

5.4.1 Attack-Free detection system

First, we analyse the robustness of our detection system in the case where we have malicious nodes performing attacks on the authentication scheme but not on the detection system.

Lemma 1. *In a network where we have malicious nodes (of rate* λ*) who perform attacks against the authentication system but not the detection system, we have:*

$$PM_{ix} = P(NR_{ix} \geq T1) = P(NP_{ix} \geq T1) \tag{5.1}$$

where $P(NR_{ix} \geq T1)$ *and* $P(NP_{ix} \geq T1)$ *represent the probabilities that* NR_{ix} *and* NP_{ix} *reach the threshold* $T1$.

Proof. As described in Section 5.2.5, a node is considered a malicious one (i.e $Pun_i = 1$) if $(NR_i \geq T1 \wedge TX_i < T2)$.

We also recall that:

$$NR_i\text{=}NP_i\text{-}NH_i$$
$$TX_i\text{=}NH_i/NP_i$$

In the case where no attacks are conducted on the detection system, if $Node_i \in Ntw$ is malicious, then the number of its honest participations $NH_i = 0$. Consequently,

$$NR_i\text{=}NP_i\text{-}NH_i = NP_i$$
$$\text{and } TX_i\text{=}NH_i/NP_i\text{= }0$$

In that case, we notice that the condition $TX_i < T2$ is always satisfied. Therefore,

$$Pun_i = 1 \text{ only if } NP_i \geq T1$$

Hence, after a number of joining phases x in a network,

$$PM_{ix} = P(NR_{ix} \geq T1) = P(NP_{ix} \geq T1) \tag{5.2}$$

□

Theorem 1. *In one random joining phase k, where we have a set of N Proxy nodes* S_{E_k} *in contact with the joining node, we express the probability that a specific malicious node* $Node_i \in Ntw$ *is selected among the N Proxy nodes in* S_{E_k} *and is reported upon successful authentication, as follows:*

$$PM_i(k) = \sum_{n=1}^{n_{max}} (C_{TM}^{n} \cdot C_{TH}^{N-n}/C_{TN}^{N})(C_{TM-1}^{n-1}/C_{TM}^{n}) \tag{5.3}$$

Where n is the number of malicious nodes among the N contacted Proxy nodes, and n_{max} represents the maximum possible value of n while ensuring successful authentication.

Proof. In order to report a specific malicious node $Node_i$ at the end of a random joining phase k, multiple conditions must be satisfied. First, the authentication must succeed. Therefore, as

explained in chapter 3, the maximum number of malicious Proxy nodes n_{max} must be strictly inferior than the half of the number N of Proxy nodes in S_{E_k}. In the case where the number N is odd, $n_{max} = \lfloor N/2 \rfloor$, where $\lfloor N/2 \rfloor$ represents the floor function which rounds $N/2$ down to the nearest integer. In the case where the number N is even, $n_{max} = (N/2) - 1$.

Given a network Ntw of TM malicious nodes and TH honest ones, the probability of selecting a number n of malicious nodes among N contacted Proxy nodes is calculated as follows:

$$C_{TM}^{n} \cdot C_{TH}^{N-n} / C_{TN}^{N} \tag{5.4}$$

Second, in order to be reported, the specific malicious Proxy node $Node_i$ must be selected among the n malicious Proxy nodes in S_{E_k}. This probability is expressed as:

$$C_{1}^{1} \cdot C_{TM-1}^{n-1} / C_{TM}^{n} \tag{5.5}$$

Therefore, in order to express the probability that we have n malicious nodes selected for the joining phase k among which the specific malicious Proxy node $Node_i$ appears, we need to multiply the two previous probabilities (i.e., eq.5.4 and eq.5.5).

Finally, the probability of the presence of at least one malicious node and at most n_{max} ones among the N Proxy nodes in S_{E_k} (i.e., $1 \leq n \leq n_{max}$), making the authentication succeeding, is expressed as follows:

$$PM_i(k) = \sum_{n=1}^{n_{max}} (C_{TM}^{n} \cdot C_{TH}^{N-n} / C_{TN}^{N})(C_{TM-1}^{n-1} / C_{TM}^{n}) \tag{5.6}$$

Theorem 2. *Given a network Ntw, where malicious nodes may conduct attacks only on the authentication system and not on the detection system, the probability that a specific malicious node $Node_i$ is detected after x joining phases in the network is expressed as follows:*

$$PM_{ix} = 1 - \sum_{j=0}^{T1-1} C_{x}^{j} (PM_i(k))^{j} (1 - PM_i(k))^{x-j} \tag{5.7}$$

Proof. According to lemme 1, the chances that a node $Node_i \in Ntw$ will be considered

as malicious depends only whether the number of reports against that node exceeds the threshold T_1 or not. Therefore, declaring $Node_i$ as malicious after x joining phases means that node has at least successfully been reported T_1 times in a sequence of x joining phases.

In that case, this probability of $Node_i$ exactly matches the definition of binomial distribution with parameters x and $PM_i(k)$ where we express the distribution of the number of successes (i.e., reporting $Node_i$ according to theorem 1 conditions) in a sequence of x independent experiments.

To express this binomial distribution, we first express the probability of reporting $Node_i$ exactly j times after x successive joining phases as follows:

$$C_x^j (PM_i(k))^j (1 - PM_i(k))^{x-j} \tag{5.8}$$

Where the probability of successfully reporting $Node_i$ in one joining phase k is $PM_i(k)$ (as shown in theorem 1).

After that, we express the probability of reporting a specific malicious node $Node_i$ a number of times $j \in \{0..T1-1\}$, as follows:

$$\sum_{j=0}^{T1-1} C_x^j (PM_i(k))^j (1 - PM_i(k))^{x-j} \tag{5.9}$$

Calculating the probability of reporting a specific malicious nodes $Node_i$ more than a threshold $T1$ after a number of joining phases x, refers to the opposite of this previous probability (eq.5.9). This probability is then expressed as follows:

$$PM_{ix} = 1 - \sum_{j=0}^{T1-1} C_x^j (PM_i(k))^j (1 - PM_i(k))^{x-j} \tag{5.10}$$

□

5.4.2 ON-OFF Attack

In this Section, we analyse the effectiveness of our detection system in the case where ON-OFF attacks are conducted by the malicious nodes against that system.

Theorem 3. *Let I_{OF} be the intensity of the ON-OFF attack conducted by the malicious Proxy nodes. It represents the percentage of their malicious behaviour from all their intervention. Let f be a range of error to be added to I_{OF} in order to cover its upper and low bound of oscillations. In fact, we add this error because the defined intensity can not be exactly the same at any time t.*

Therefore, for one random joining phase k, having a number of Proxy nodes N in contact with the joining node, we express the probability that a specific malicious node $Node_i$ is reported as follows:

$$\begin{aligned} PM_i(k) = (I_{OF} \pm f) \times & \\ & \sum_{n=1}^{n_{max}} (C_{TM}^{n} \cdot C_{TH}^{N-n} / C_{TN}^{N})(C_{TM-1}^{n-1} / C_{TM}^{n}) \\ & + (I_{OF} \pm f) \times \\ & \sum_{n=(n_{max})+1}^{N} \Bigg((C_{TM}^{n} \cdot C_{TH}^{N-n} / C_{TN}^{N})(C_{TM-1}^{n-1} / C_{TM}^{n}) \\ & \times \sum_{j=n-n_{max}}^{n-1} C_{n-1}^{j} (1 - (I_{OF} \pm f))^{j} (I_{OF} \pm f)^{n-1-j} \Bigg) \end{aligned} \tag{5.11}$$

Proof. In order to detect a specific malicious node $Node_i$, after a random joining phase k, multiple conditions must be satisfied. This includes the probability that $Node_i$ is selected between the N Proxy nodes in S_{E_k}, $Node_i$ acted maliciously in this joining phase k, the authentication succeeded and $Node_i$ is reported. Hence, Equation 5.11 covers multiple probabilities together. We split them into parts in order to clarify each part.

The first part represents the same Equation in 5.3 multiplied by the coefficient $(I_{OF} \pm f)$. Equation 5.3 satisfies the conditions mentioned earlier, except the condition where $Node_i$ is acting malicious. The multiplication with the coefficient $(I_{OF} \pm f)$ satisfies this condition. In this case, two probabilities are calculated, the one of upper bound considering an intensity of ON-OFF attack of $(I_{OF} + f)$, and the one with lower bound considering $(I_{OF} - f)$.

The second part is specific for the case of ON-OFF attack. The malicious Proxy nodes in S_{E_k} may be acting honestly in this joining phase k. Therefore, we must count in our probability calculation the cases where the number of malicious nodes in S_{E_k} exceeds n_{max}. This is

expressed as follows:

$$\sum_{n=(n_{max})+1}^{N} (C_{TM}^{n} \cdot C_{TH}^{N-n} / C_{TN}^{N})(C_{TM-1}^{n-1} / C_{TM}^{n}) \tag{5.12}$$

However, while counting these cases going from $n_{max}+1$ to N, we must multiple this by the corresponding probability that between these malicious Proxy nodes, a number of Proxy nodes are acting honestly in this joining phase k, in a way that the majority of the N contacted Proxy nodes are acting honestly during this joining phase k. This is expressed as follows:

$$\sum_{j=n-n_{max}}^{n-1} C_{n-1}^{j} (1-(I_{OF} \pm f))^{j} (I_{OF} \pm f)^{n-1-j} \tag{5.13}$$

Finally, this part is multiplied by the coefficient $(I_{OF} \pm f)$ as well, satisfying the condition that this $Node_i$ is acting malicious in this joining phase k. □

Theorem 4. *In a network, after a number x of joining phases happened, we calculate the probability that a specific Proxy node $Node_i$ is detected as malicious. In the case of an ON-OFF attack conducted against the detection system, the conditions are that NR_i reaches $T1$ and $TX_i < T2$. This can be represented as follows:*
While ($T2 > (1-(I_{OF}-f))$):

$$PM_{ix} = (1 - \sum_{j=0}^{T1-1} C_{x}^{j} (PM_i(k))^{j} (1-PM_i(k))^{x-j}) \tag{5.14}$$

Proof. As explained for Equation 5.7, we calculate the probability of reporting a specific malicious nodes $Node_i$ more than a threshold $T1$ after a number of joining phases x. Moreover, in order to satisfy the second condition we must have:

$$TX_i < T2$$

However,

$$TX_i = NH_i / NP_i = 1 - NM_i / NP_i = (1-(I_{OF} \pm f))$$

since the percentage of honest participation for a $Node_i$ represents the opposite of I_{OF}, with a range of oscillations going from $(I_{OF}-f)$ to $(I_{OF}+f)$;
Therefore, the second condition is satisfied when:

$$T2 > 1-(I_{OF}-f)$$

since $T2$ must be higher than the opposite of the lower bound of I_{OF}. To ensure a proper functioning of the detection system and getting the expected rate of detection presented in equation 5.14, the value of $T2$ must respect this condition. □

5.4.3 False Positive

We now analyse the effectiveness of our detection system while a False Positive attack is conducted by a part of the joining nodes.

Theorem 5. *Let I_{FP} be the intensity of the False Positive attack conducted by the joining nodes. It represents the percentage of joining nodes conducting a False Positive attack from all the upcoming joining nodes. Therefore, for one random joining phase k, having a number of Proxy nodes N in contact with the joining node, we calculate the probability that a specific honest node $Node_i$ is reported. This represents the False Positive attack success and it can be calculated as follows:*

$$\begin{aligned} PH_i(k) = I_{FP} \times (\sum_{n=0}^{n_{max}} (C_{N-n-1}^{n_{max}-n-1} / C_{N-n}^{n_{max}-n})) / n_{max} \\ \times \sum_{n=0}^{n_{max}} (C_{TM}^{n} \cdot C_{TH}^{N-n} / C_{TN}^{N})(C_{TH-1}^{N-n-1} / C_{TH}^{N-n}) \end{aligned} \tag{5.15}$$

Proof. In order to consider a specific honest node $Node_i$ as malicious, after a random joining phase k, multiple conditions must be satisfied. This includes the probability that $Node_i$ is selected between the N Proxy nodes in S_{E_k}, the authentication succeeded, the joining node in the joining phase k is malicious and $Node_i$ is chosen to be reported. Hence, Equation 5.15 covers multiple probabilities together. We split them into parts in order to clarify each part. First, in order to have a successful authentication, we must have to the maximum n_{max} of the nodes in S_{E_k} are malicious. This can be calculated as follows:

$$\sum_{n=0}^{n_{max}} (C_{TM}^{n} \cdot C_{TH}^{N-n} / C_{TN}^{N}) \tag{5.16}$$

Second, the probability that the specific honest node $Node_i$ is between the honest Proxy nodes from the N Proxy nodes in S_{E_k} can be calculated as follows:

$$(C_{TH-1}^{N-n-1} / C_{TH}^{N-n}) \tag{5.17}$$

The condition that the joining node in the joining phase k is malicious can be expressed by the multiplication with I_{FP}. Finally, in a joining phase k, a malicious joining node may have zero, one or multiple choices of honest nodes from S_{E_k} to report. The probability that the honest $Node_i$ is chosen between these nodes can be expressed as follows:

$$(\sum_{n=0}^{n_{max}} (C_{N-n-1}^{n_{max}-n-1} / C_{N-n}^{n_{max}-n})) / n_{max} \tag{5.18}$$

□

Theorem 6. *In a network, after a number x of joining phases happened, we calculate the probability that a specific honest node $Node_i$ is considered as malicious. In the case of a False Positive attack conducted against the detection system, the conditions are that NR_i reaches $T1$ and $TX_j < T2$. The upper bound of the possible false positive success rate can be expressed as follows:*

$$PH_{i\,x} = (1 - \sum_{i=0}^{T1-1} C_x^i (PH_i(k))^i (1 - PH_i(k))^{x-i}) \tag{5.19}$$

Proof. As explained for Equation 5.7 and 5.14, we calculate the probability of reporting a specific honest node $Node_i$ more than a threshold $T1$ after a number of joining phases x. Moreover, as we tune $T2 = 1.0$, the second condition is always satisfied since $\forall\ TX_i$, $TX_i < 1$. Therefore, for the worst case scenario, if $T2 = 1.0$, the only condition is that NR_i reaches $T1$. Hence, in this case, we calculate the highest possible value for the False Positive success.

□

Theorem 7. *In a network, after a number x of joining phases happened, in the case of a False Positive attack conducted against the detection system, the conditions that a specific honest node $Node_i$ is not considered as malicious are NR_i doesn't reach $T1$ or $TX_i > T2$. This can be represented as follows:*
While ($T2 < 1 - I_{FP}$):

$$PH_{i\,x} = 0$$

Proof. In order to satisfy the second condition of not considering an honest $Node_i$ as mali-

cious, we must have:

$$TX_i > T2$$

However,

$$TX_i = NH_i/NP_i = 1 - NM_i/NP_i = (1 - (I_{FP}))$$

since the percentage of honest participation for a $Node_i$ represents the opposite of I_{FP}.

Therefore, the second condition is satisfied when:

$$T2 < 1 - (I_{FP})$$

To ensure a proper functioning of the detection system, the value of $T2$ must respect this limits. □

5.5 Summary

In this chapter, we have introduced a novel solution for detecting malicious proxy nodes during the IoT network's joining phase. This detection system is centralized at the network coordinator and relies on the participation logs of nodes in the network as proxy nodes, their honest involvement, and the number of reports received from joining nodes regarding these proxy nodes. Our solution accounts for various attack types launched by proxy nodes within the network or by joining nodes.

We have conducted a theoretical evaluation to assess the effectiveness of our solution based on the detection system parameters and network conditions. This evaluation considered multiple types of attacks, enhancing the comprehensiveness of our study.

In the next chapter, we will evaluate the robustness of this solution.

6 Detection system: A Performance Evaluation

In chapter 5, we presented a solution for malicious proxy nodes detection during IoT network joining phase. In this chapter, we aim to evaluate the security robustness of our proposed solution.

6.1 Experimental Setup

We conducted an experiment in a distributed system where we consider one coordinator, multiple nodes and one joining node at a time. We monitor a series of joining phases to the network after its boot. At a time t, one joining node is in the joining phase of this network. The joining phase is accomplished according to the consensus detailed in chapter 3. Same for the report and detection mechanism, they are implemented as described in Section 5.2. The simulation parameters are presented in Table 6.1. It consists of a network with a total number of nodes $TN = 100$ and a number of joining phases $Nb = 1000$ indicating the number of nodes to join the network. The number of Proxy nodes N in contact with the joining node is set to 5 and the rate of malicious nodes λ is set to 33% of the total nodes in the network (this value of λ is the worst case according to the byzantine fault tolerance assumption [LSP19]). We adopted these two last values based on a previous experiment in section 3.10.1 that shows an average authentication success rate of 80% with these values. This simulation model does not take the node's position into account, therefore, the selection of an Proxy node by a joining node is always purely random. We recall that a report from a node to the coordinator does not

Total number of nodes TN	100
Number of Proxy nodes N	5
Degree m of Q(x)	2
Rate of malicious nodes λ	33%
Number of joining phases Nb	1000
Threshold $T1$	{0,5,10}, default value=5
Threshold $T2$	[0-1.0], default value=0.5
ON-OFF attack intensity I_{OF}	{0,20,30}, default value=0
False Positive attack intensity I_{FP}	{0,22,33}, default value=0
False Negative attack intensity I_{FN}	{0,20,30,50}, default value=0

Table 6.1: Simulation parameters.

happen unless the authentication succeeds. Moreover, we consider multiple types of attacks in our detection system (ON-OFF, False Positive and False Negative). We vary as well the combinations of attacks and their intensities. For a clear evaluation of our detection system, and in order to keep the same rate of malicious nodes in the network, we do not execute any punishment mechanism on the detected malicious nodes in this experiment, and the new joining nodes are not counted as nodes from the network after they successfully do the joining phase.

In order to have a statistically significant representation of the evaluation of the detection system, we represent in each experiment result the average of 100 simulation rounds where each simulation is a series of 1000 joining phases.

6.2 Attack-Free scenario

In this Section we vary the main parameters of our detection system to see their impact on the detection of malicious Proxy nodes. We note that in this experiment no attacks are conducted against the detection system. Moreover, we compare the results to the theoretical evaluation in 5.4.1. Figure 6.1 represents the impact of the variation of threshold T1 on the detection rate, while threshold T2 is fixed to 0.5. The detection rate represented in the graph is calculated after each joining phase by dividing the number of malicious nodes detected by the total number of malicious nodes in the network. We clearly see the impact of T1 on the acceleration of the detection rate. For T1=1, the detection rate reached 100% after around 180 iterations, for T1=5 after around 410 and for T1=10 after around 650. For the remaining experiments, we will set

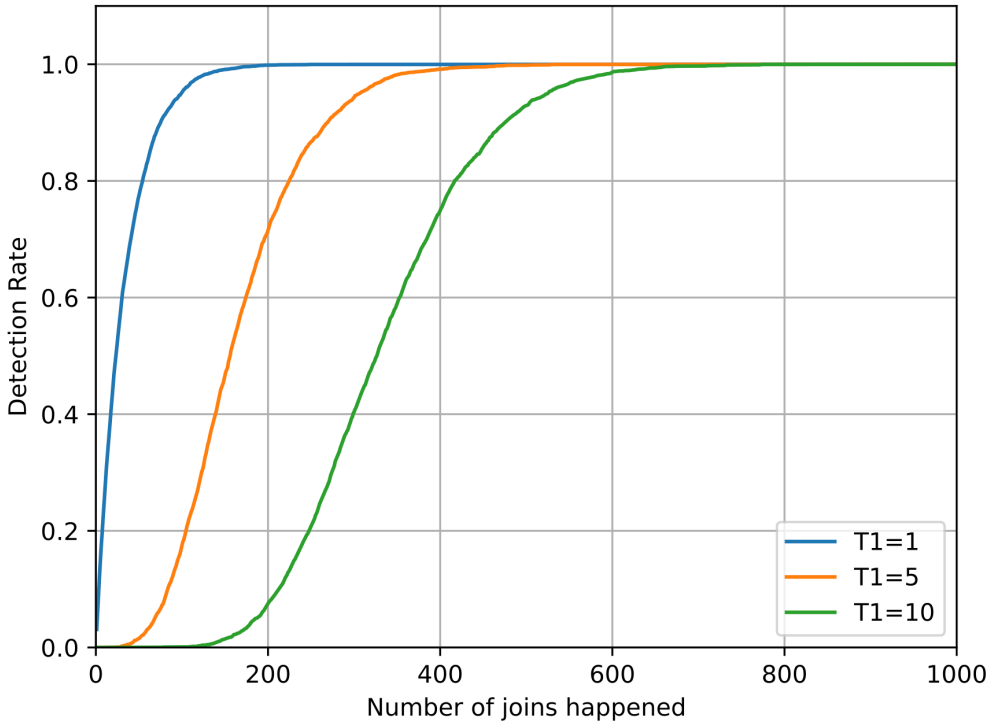


Figure 6.1: Variation of the detection rates with Threshold T1.

$T = 5$ as it produces relevant results, considering this size of the network.

Figure 6.2 represents the impact of the variation of T2 on the detection rate, while T1 is fixed to 5. We see that in the case of a scenario where no attacks are conducted on the detection system, T2 has no impact on the detection rate. This is consistent with what is represented in Equation 5.1.

In Figure 6.3, we vary the rate λ of malicious nodes in the network while fixing $T1 = 5$ and $T2 = 0.5$. We adopt three values of λ: 10%, 20% and 33%. For each one, we evaluate the detection rate in the network and the detection success of a specific node $Node_i$. That rate is calculated by dividing, for each number of joining phase x, the number of times this $Node_i$ is detected after this joining phase x in all the simulations, on the total number of rounds of simulations. We compare this curve to the curve representing the detection rate (which is the average detection rate of all the simulations rounds), and the curve representing PMi_x (the probability of detecting a specific malicious node $Node_i$), calculated with Equation 5.7 for

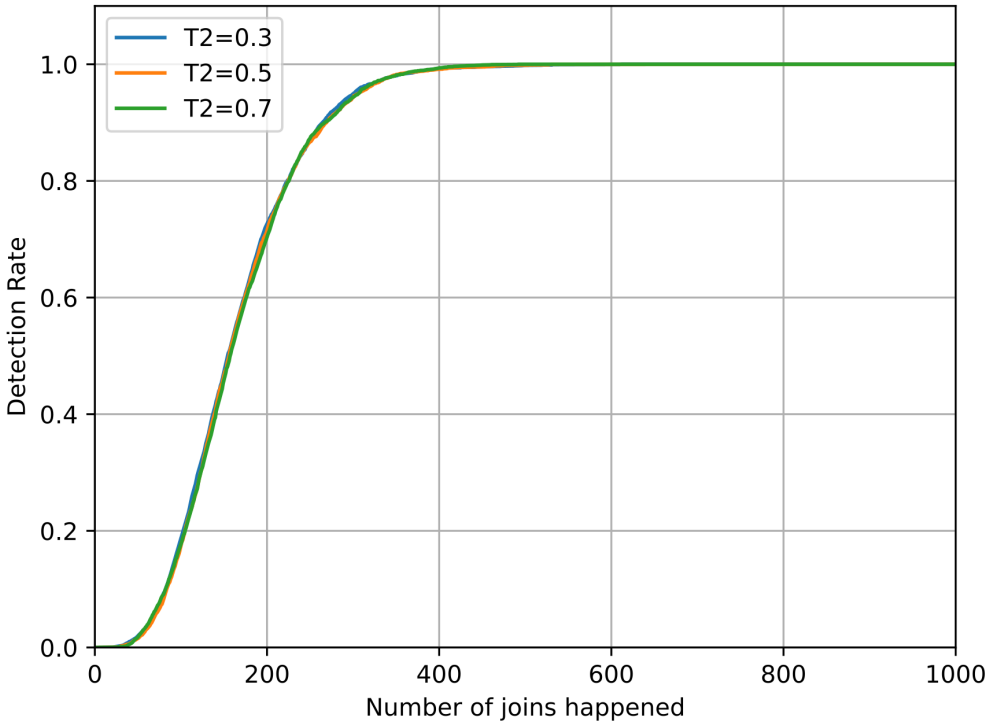


Figure 6.2: Variation of the detection rates with Threshold T2.

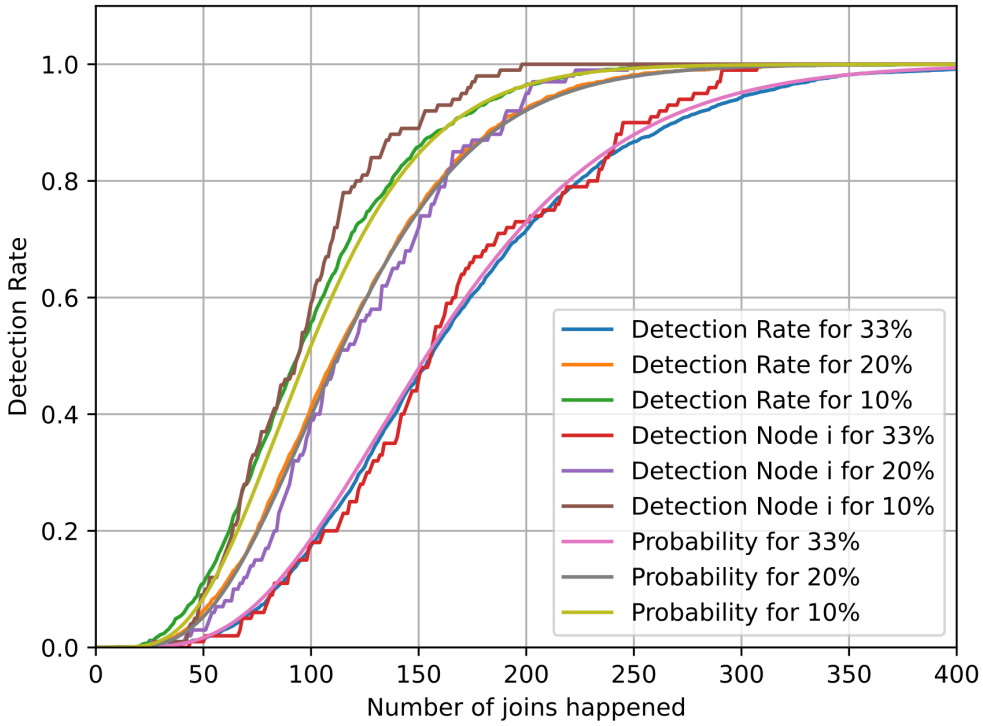


Figure 6.3: Variation of the detection rates compared to the probability in function of λ (T1=5 T2=0.5).

each number of joining phase x in the simulation going from 0 to 1000. We see that for each value of λ, the curve of the detection rate is similar to the the curve of probability. Moreover, the detection success of a $Node_i$ has the same distribution of these two curves.

6.3 ON-OFF Attack

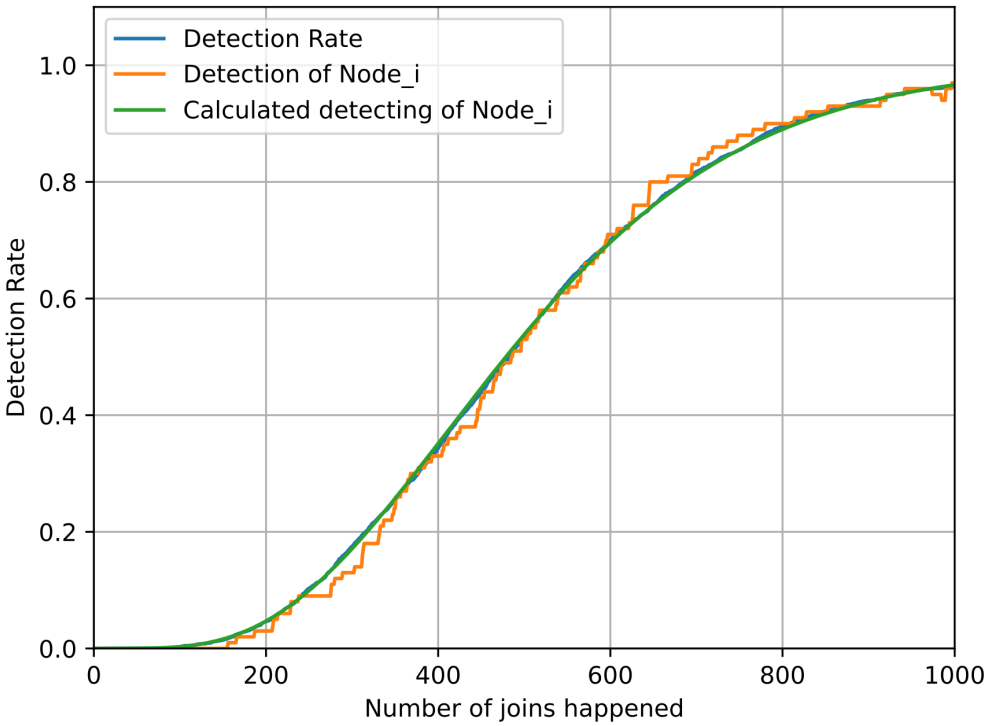
In this Section we evaluate the impact of an ON-OFF attack on our detection system. We fix in this Section $\lambda = 33\%$ and $T1 = 5$. We vary each time the intensity of the ON-OFF attack I_{OF} and the threshold $T2$ of the detection system. We recall that I_{OF} represents the percentage of time a malicious node acts maliciously on the total time of its participation as a Proxy node. Figure 6.4 and 6.5 represent the variation of the detection rate with the intensities of ON-OFF attack of 20% and 30% respectively. For each intensity we present two results with two values of $T2$: $T2 = 1 - (I_{OF} - f)$ representing the lower bound of $T2$ for the well functioning of the detection system according to the study presented in Section 5.4.2 and $T2 = 1 - I_{OF}$ representing a value

of $T2$ lower than threshold to be respected. We tune the value of f to 0.1 in this experiment. We compare this curve to the curve representing the detection success of a malicious node $Node_i$ and the curve representing PMi_x (the probability of detecting $Node_i$ in the case of an ON-OFF attack), calculated with Equation 5.14 (We highlight that the curves representing the probability calculation are not included in the figures illustrating the detection rate with a value of $T2$ not considering the value f. This omission is intentional, as in such cases, the probability cannot be reliably expected.). We see that the detection rate and the detection success of $Node_i$ in each graph are consistent with the corresponding calculated probability. Moreover, we clearly see the impact of usage of f on the detection rate.

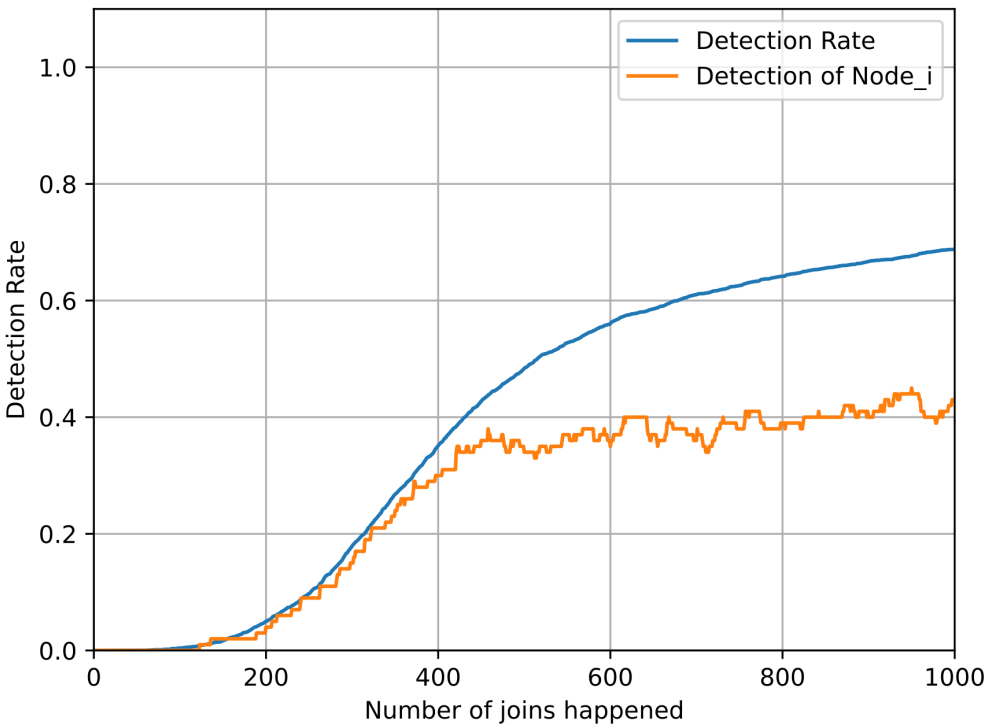
6.4 False Positive Attack

In this Section we evaluate the impact of the False Positive attack conducted by the joining nodes on the detection rate and success rate of the False Positive attack. This attack is conducted with different intensities. For example, a False Positive attack of intensity 50 % means that half of the 1000 joining nodes are sending false reports about honest nodes to the coordinator of the network after the joining phase. We recall that in such type of attacks, the joining node can only report a limited number of nodes between the Proxy nodes it is in contact with. It must show that it was in contact with a majority of honest nodes, which led it to successfully join the network. Figure 6.6 represents the variation of the detection rate and the False Positive success rate with the False Positive attack intensity I_{FP}. We fix $T1 = 5$ and $T2 = 1.0$. As explained in Section 5.4.3, $T2 = 1.0$ leads to the highest possible False Positive success rate. The detection rate converges to 100 % after around 400 joining phases. We see that the impact of different intensities of False Positive attack (I_{FP}=22% and I_{FP}=33%) on the detection rate is almost null. Comparing these results to Figure 6.1 for the same simulation parameters, we see that this attack conducted alone has no impact on the detection rate. Moreover, we see that the highest value of False Positive success rate is always lower than the probability of the highest False Positive success PH_{i_x} calculated using the formula in Equation 5.19.

Figure 6.7 represents the variation of the detection rate and the False Positive attack success rate while adopting the same intensities in Figure 6.6 but with $T2 < 1 - I_{FP}$ as explained in

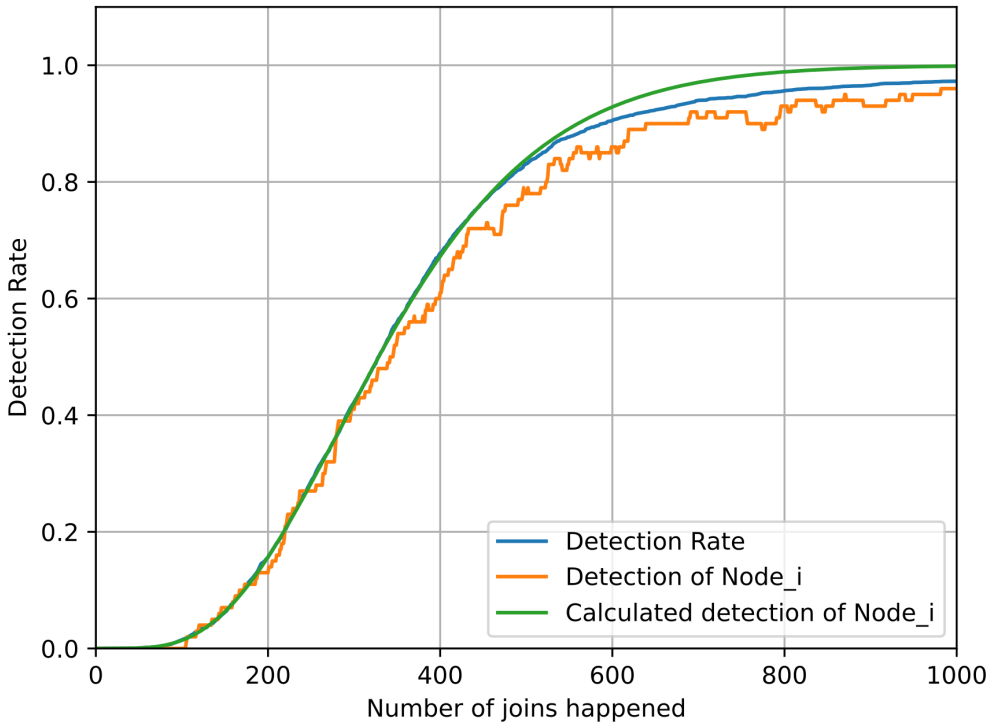


(a) T2=0.9 (T2=1-(0.2-0.1)

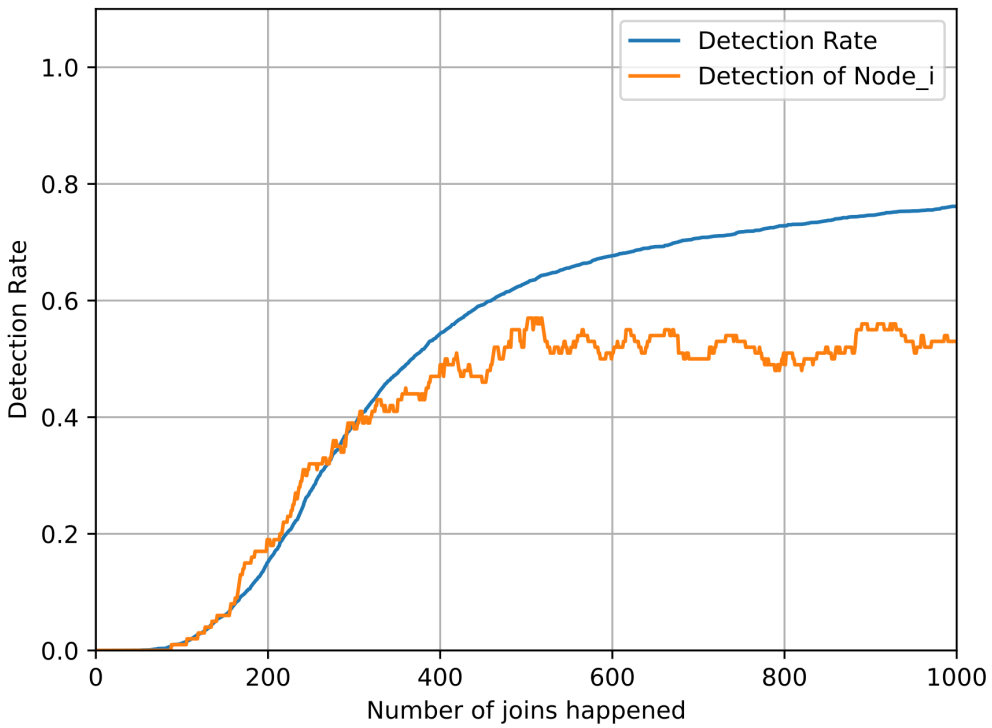


(b) T2=0.8 (T2=1-0.2)

Figure 6.4: Variation of the detection rates with threshold T2 for intensity I_{OF}=20%.



(a) T2=0.8 (T2=1-(0.3-0.1))



(b) T2=0.7 (T2=1-0.3)

Figure 6.5: Variation of the detection rates with threshold T2 for intensity I_{OF}=30%.

Theorem 7. We see that when tuning an appropriate value of $T2$ for an expected intensity of False Positive attack, the False Positive success becomes almost null.

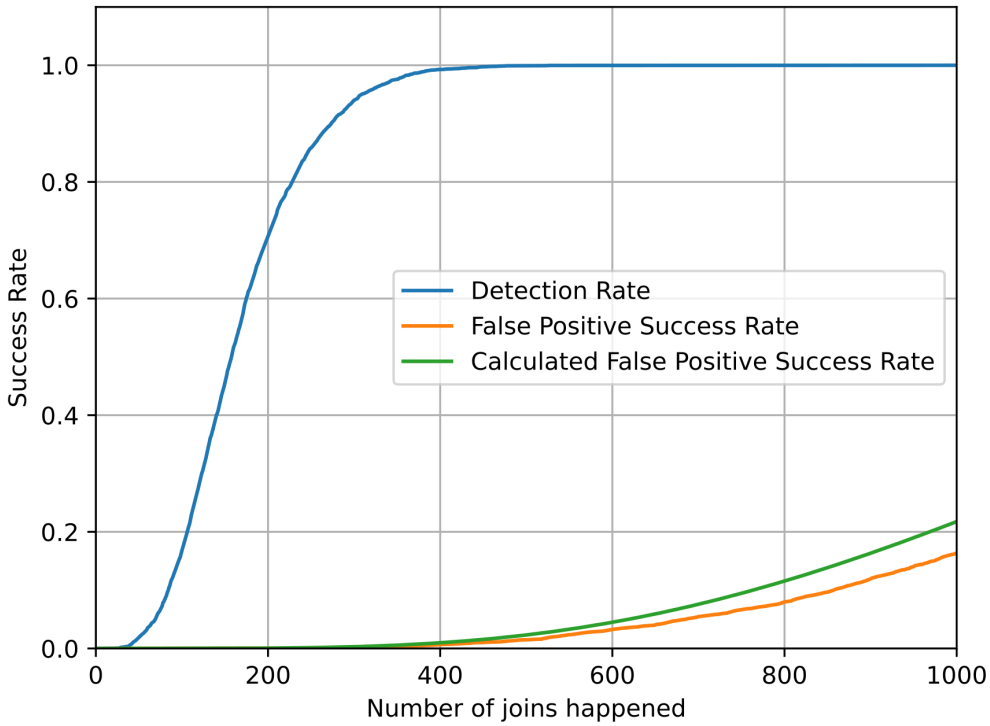
6.5 False Negative Attack

Figure 6.8 represents the variation of the detection rate while conducting False Negative attacks by the joining nodes. This attack is conducted with different intensities. For example, a False Negative attack of intensity 50 % means that the half of the 1000 joining nodes are not sending reports about the detected malicious nodes to the coordinator of the network after the joining phase. Comparing these results to the results in Figure 6.2 where the parameters are the same but no attacks are conducted, we see that this type of attack decelerates somewhat the detection speed, specifically when conducted with a high intensity ($I_{FN} = 50\%$).

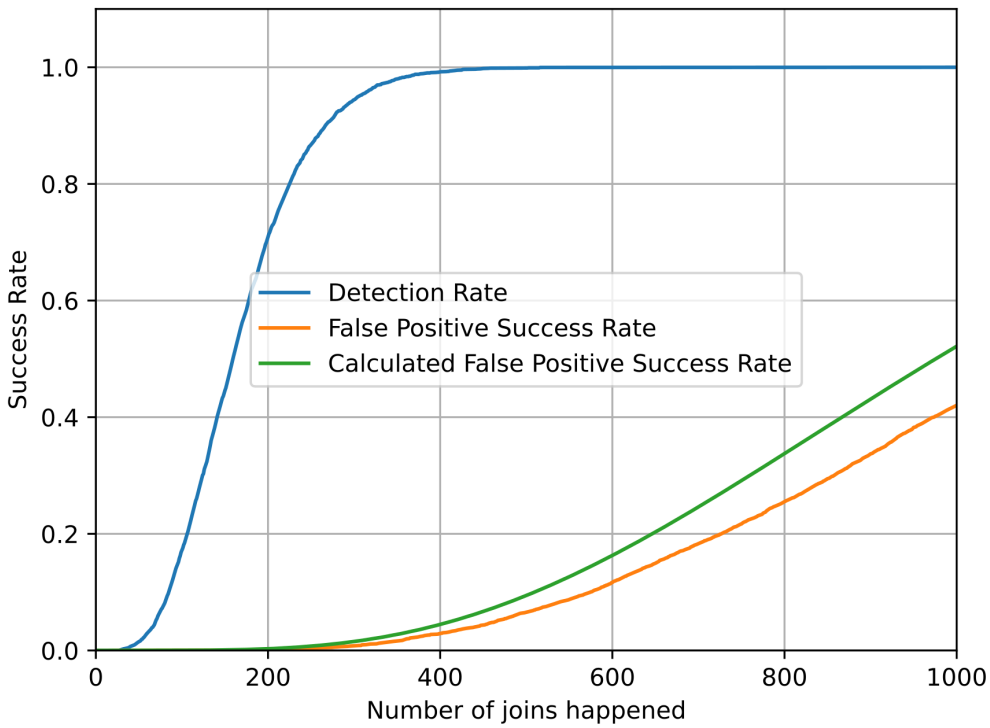
6.6 Combination of Attacks

In this Section we conduct simultaneously multiple attacks on the detection system. We aim therefore to evaluate the accuracy of the coordinator in distinguishing between two contradictory situations, like ON-OFF attack vs False Positive attack, or False Positive attack vs False Negative attack. Figures 6.9 represents an experiment with both False Positive and False Negative attacks conducted against the detection system, where $I_{FP} = 22\%$ and $I_{FN} = 20\%$. One joining node can not conduct both attacks at the same time. We vary the value of $T2$ between $T2 = 1$ and $T2 < 1 - I_{FP}$. The same experiment is conducted in Figure 6.10 with $I_{FP} = 33\%$ and $I_{FN} = 30\%$. We see that for both values of $T2$, the detection rate is similar to the detection rate in Figure 6.8 where a False Negative attack is conducted alone for the same values of intensities. For the False Positive success rate, for $T2 = 1$, we see that its value in figures 6.9b and 6.10a is lesser than the one where the False Positive attack is conducted alone for the same intensities in figures 6.6a and 6.6b respectively. For $T2 < 1 - I_{FP}$ we see that the False Positive success rate is almost null for both intensities.

In Figure 6.11, we conduct ON-OFF attack and False Positive attack simultaneously. We adopt $I_{OF} = 20\%$ and $I_{FP} = 22\%$. We variate the value of $T2$ from $T2 > 1 - (I_{OF} - f)$ to $T2 < 1 - I_{FP}$, where $f = 0.1$. The same experiment is conducted in 6.12 with $I_{OF} = 30\%$ and $I_{FP} = 33\%$. We

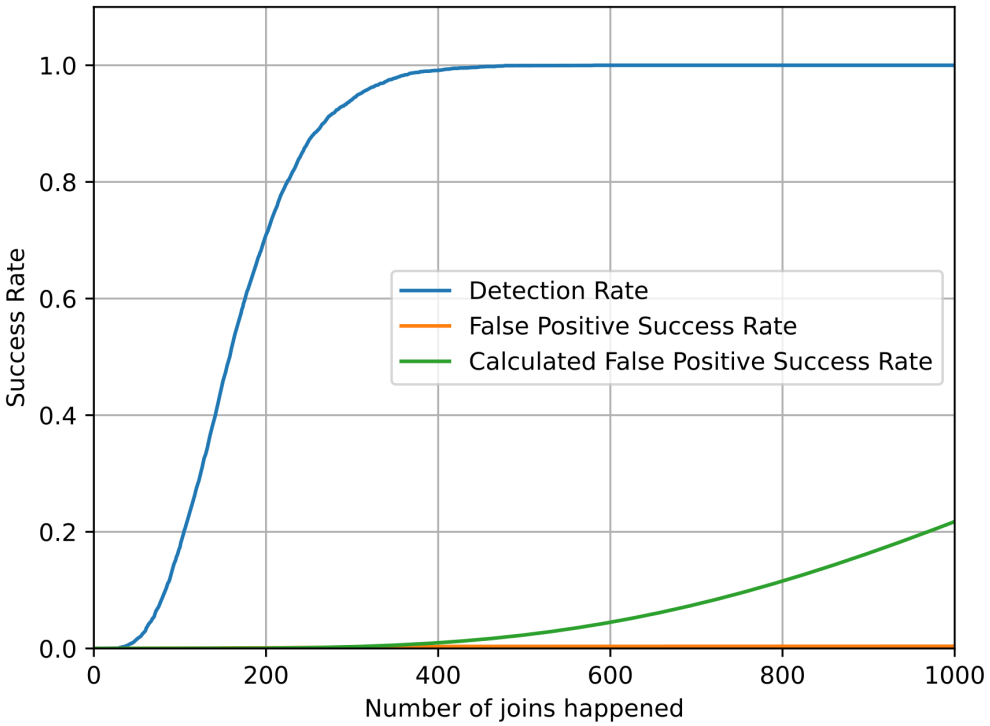


(a) $I_{FP} = 22$

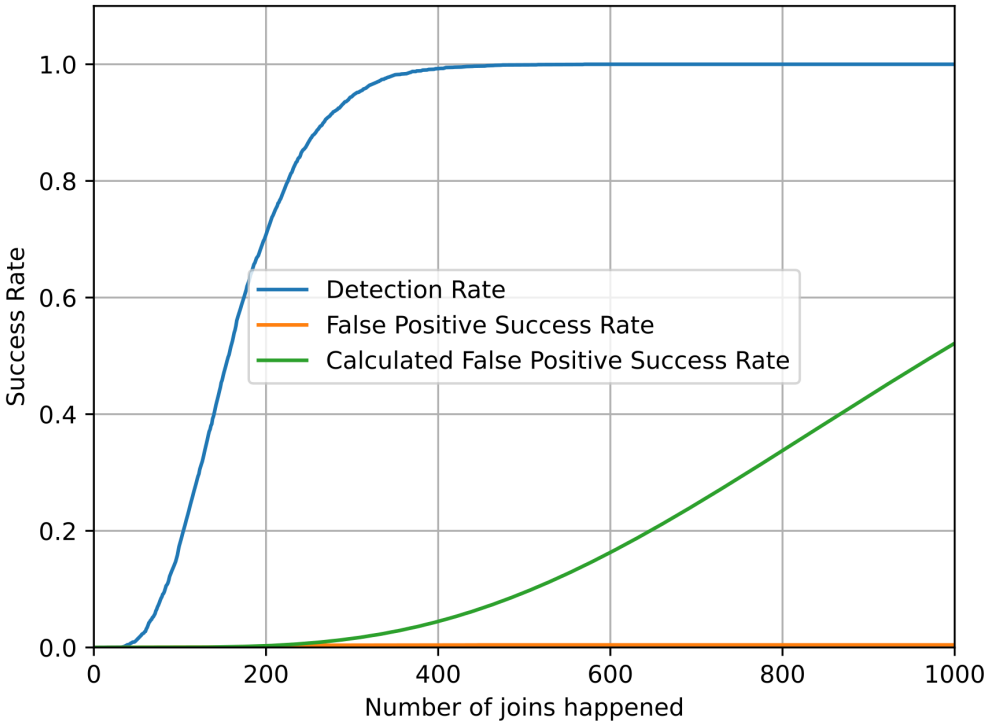


(b) $I_{FP} = 33$

Figure 6.6: Variation of the detection rate and false positive success compared to the probability for different False Positive attack intensities and $T2 = 1$.



(a) $I_{FP} = 22$ and $T2 = 0.7$



(b) $I_{FP} = 33$ and $T2 = 0.6$

Figure 6.7: Variation of the detection rate and false positive success compared to the probability for different False Positive attack intensities and different values of $T2$.

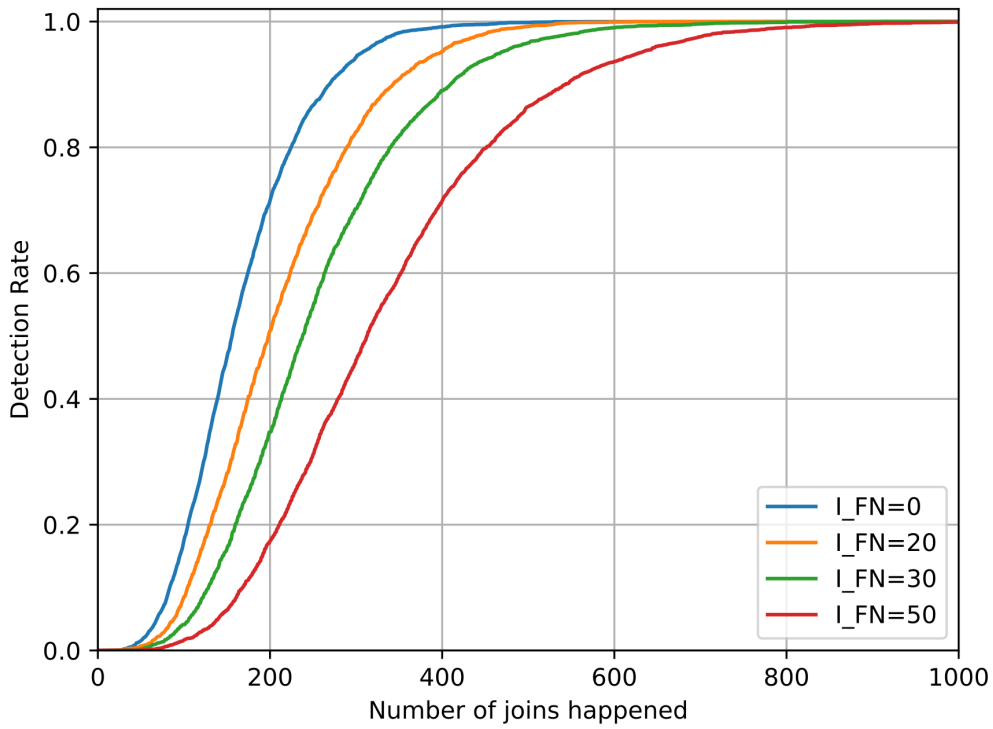


Figure 6.8: Variation of the detection rates while conducting a False Negative attack (T1=5 T2=0.7).

see that for $T2 > 1 - (I_{OF} - f)$, where $f = 0.1$, the detection rate is the highest possible, and False Positive rate is considerable. For $T2 < 1 - I_{FP}$, the detection rate is not good enough while the False Positive success rate is almost null.

6.7 Punishment

In this experiment, we aim to evaluate the evolution of the detection rate when the coordinator punishes the detected malicious nodes and excludes them from further joining phases. Figure 6.13 represents the same simulation presented in Figure 6.1 ($T1 = 5$, $T2 = 0.5$) while adding the punishment phase. Comparing the graphs in these two figures, having the same simulation parameters and thresholds , we see that the detection rate is faster, (after 250 joining phases with punishment VS. after 400 without punishment).

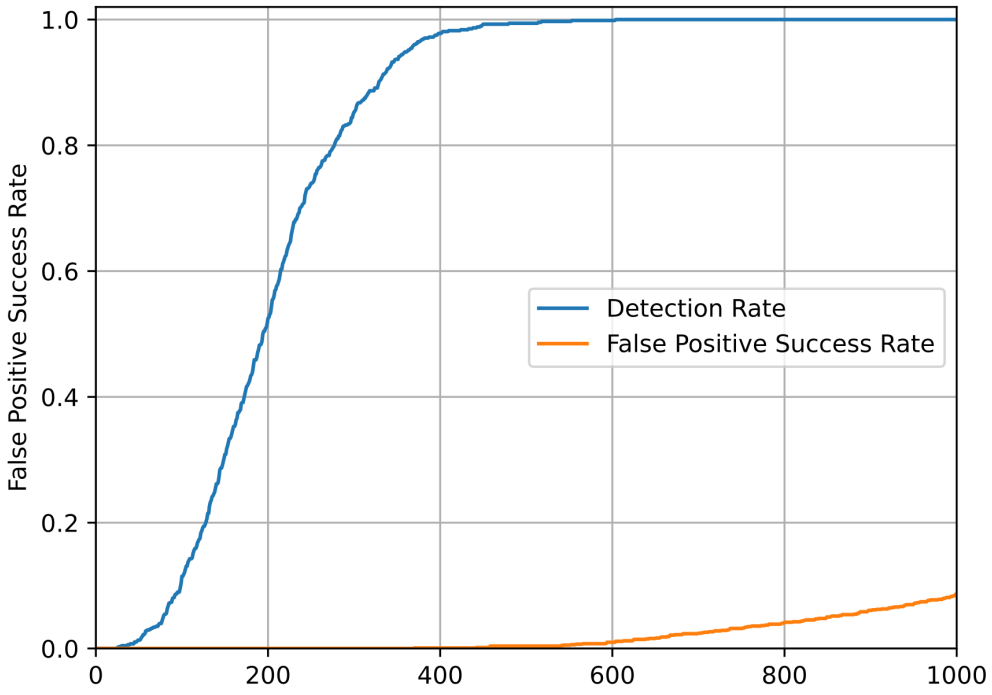
6.8 Discussion

We showed in this chapter that, even though we consider a limited number of factors in our detection system, our solution is practical and ensures a higher level of safety for the joining phases of an IoT network. This Section provides further discussion of the parameters used and some corner cases.

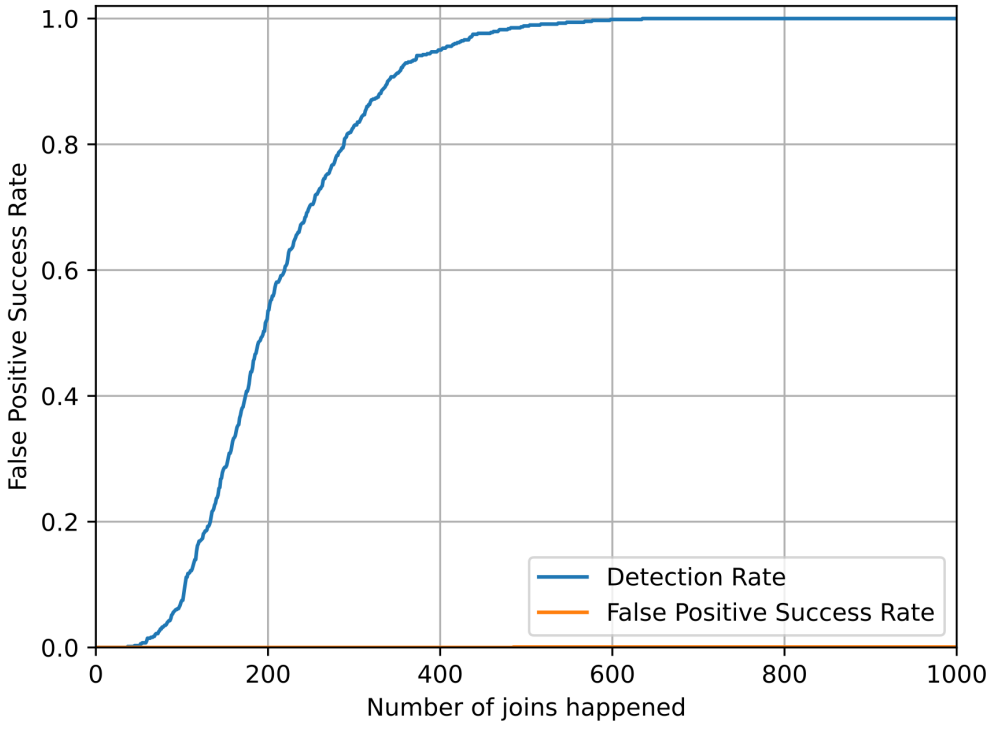
First, since this solution is proposed for the joining phase context, a limited number of factors can be considered in such detection system. However, this makes the solution more simple and more realistic to be implemented in different IoT networks protocols and topology. Moreover, the centralised aspect of this solution makes it more suitable for IoT where one entity is responsible most of the time for the communication and security of its down tree nodes.

Second, we assumption that the node's position is not considered during the joining phases, is realistic enough to reflect the general characteristics of IoT networks. In the general context of a large scale IoT network, the probability of a node being either malicious or honest is inherently random. By embracing this randomness in the selection of Proxy nodes by joining nodes, our model aligns with the unpredictable nature of nodes in typical IoT scenarios.

Third, the parameters $T1$ and $T2$ of our detection system must be tuned in an appropriate way

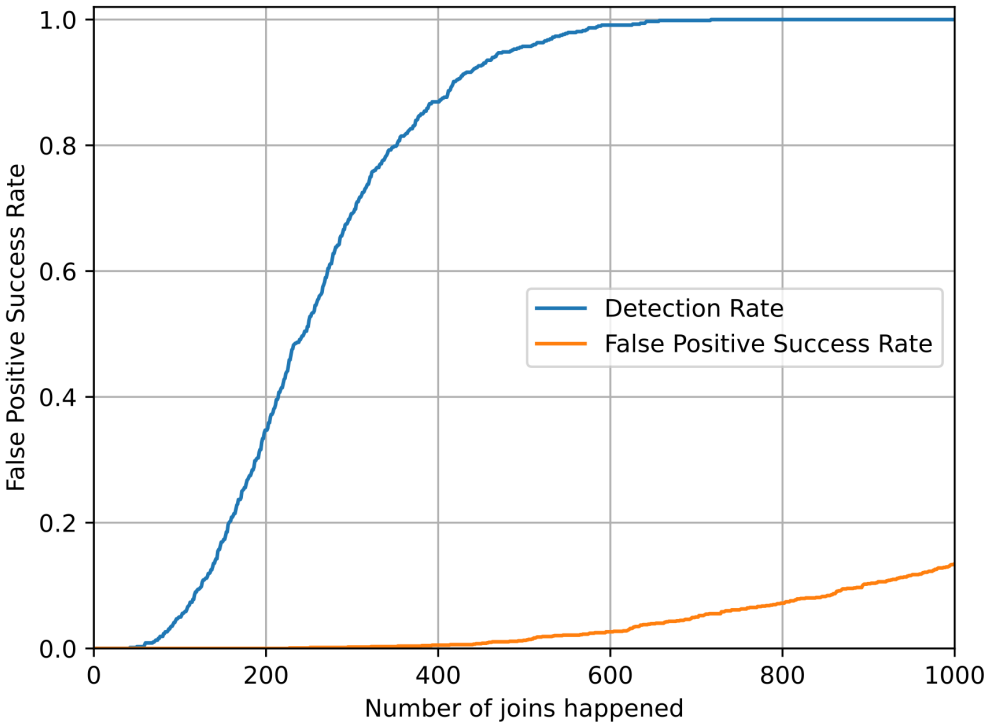


(a) $T2 = 1.0$

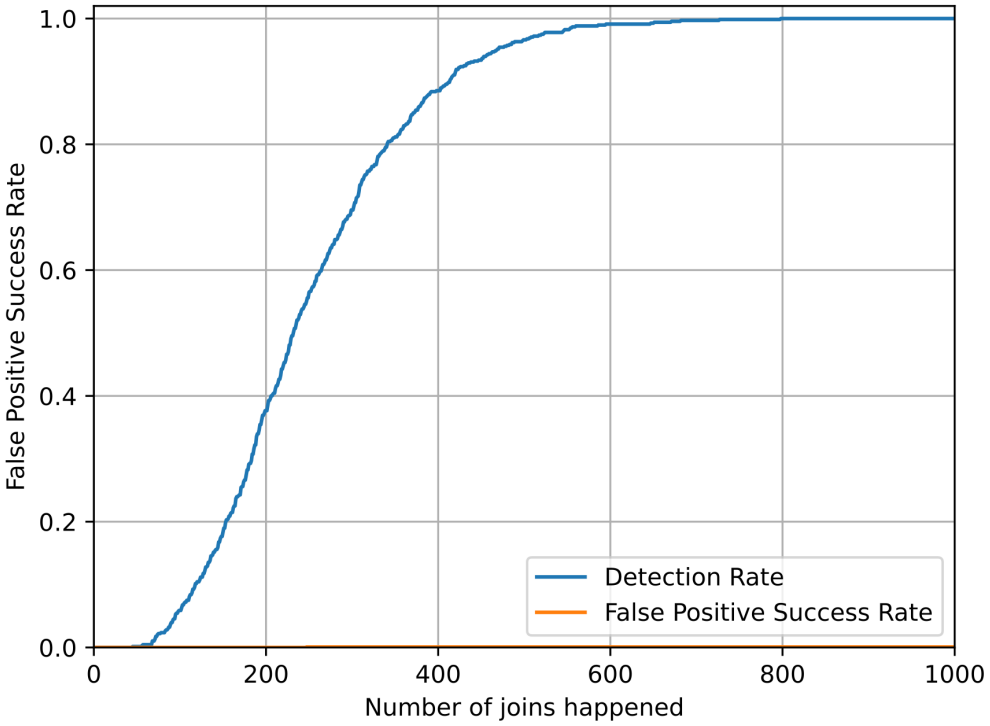


(b) $T2 = 0.7$

Figure 6.9: Variation of the detection rate and false positive success for $I_{FP} = 22\%$ and $I_{FN} = 20\%$ with different values of $T2$.

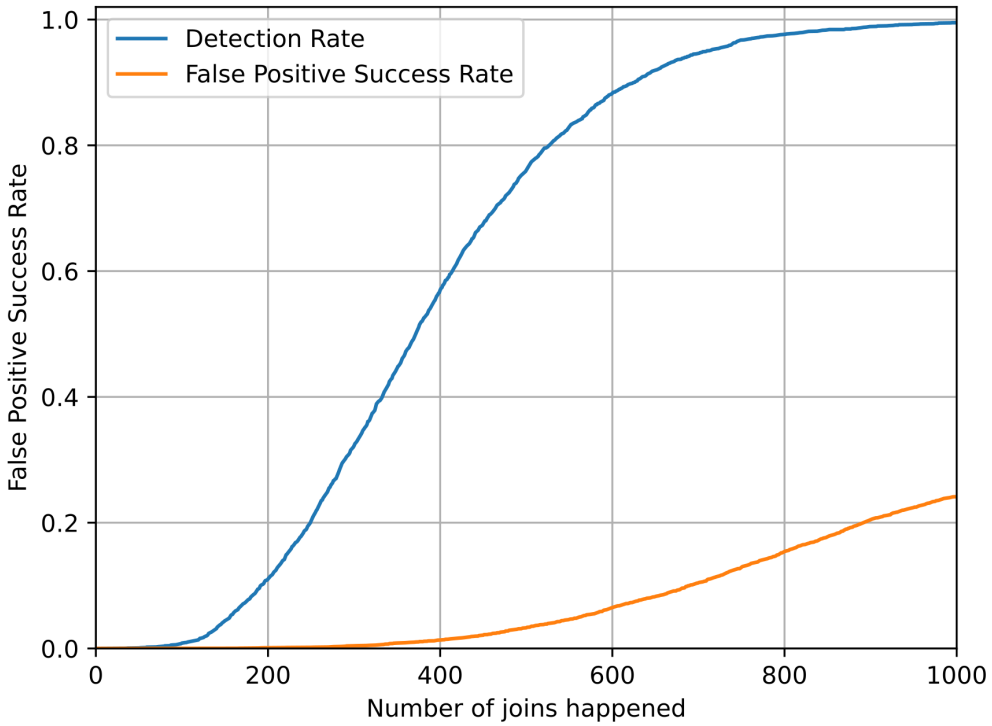


(a) $T2 = 1.0$

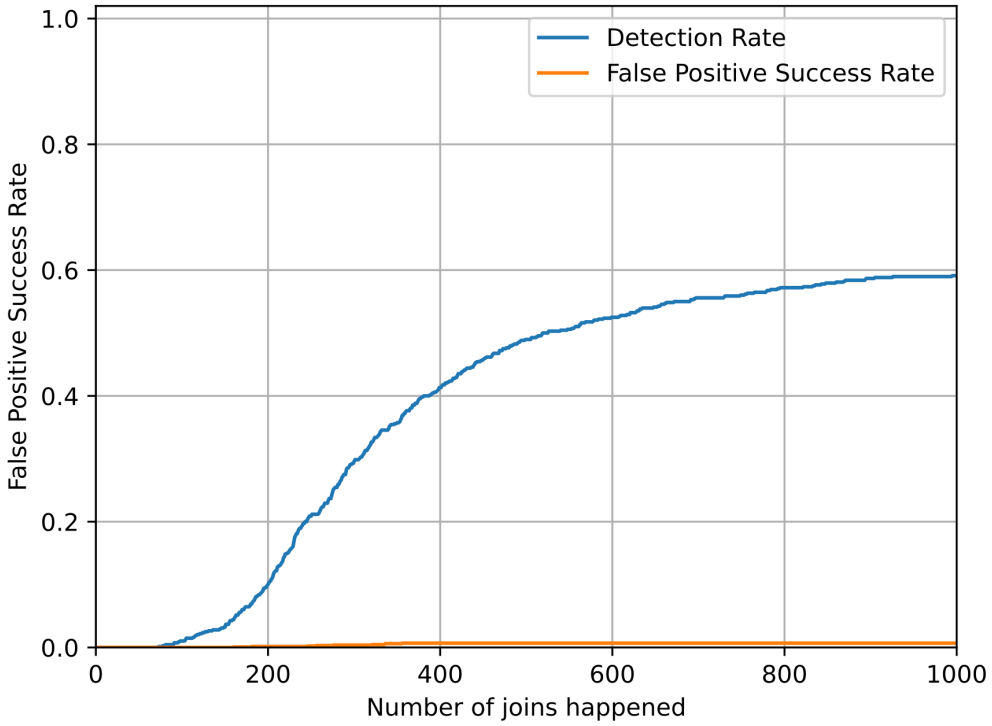


(b) $T2 = 0.6$

Figure 6.10: Variation of the detection rate and false positive success for $I_{FP} = 33\%$ and $I_{FN} = 30\%$ with different values of $T2$.

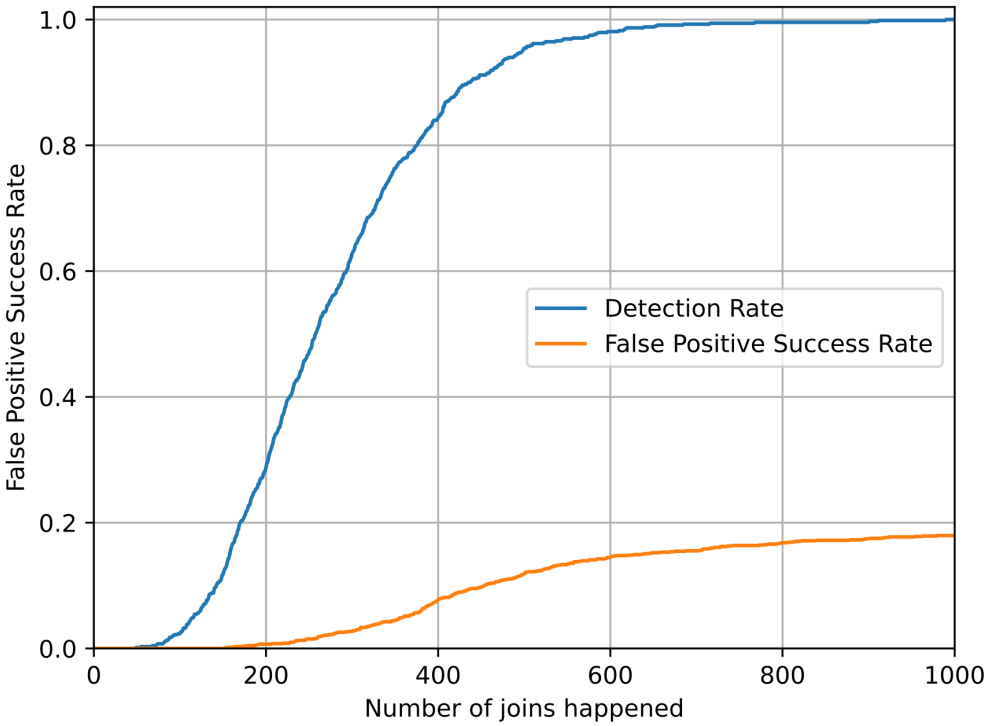


(a) $T2 = 0.9$

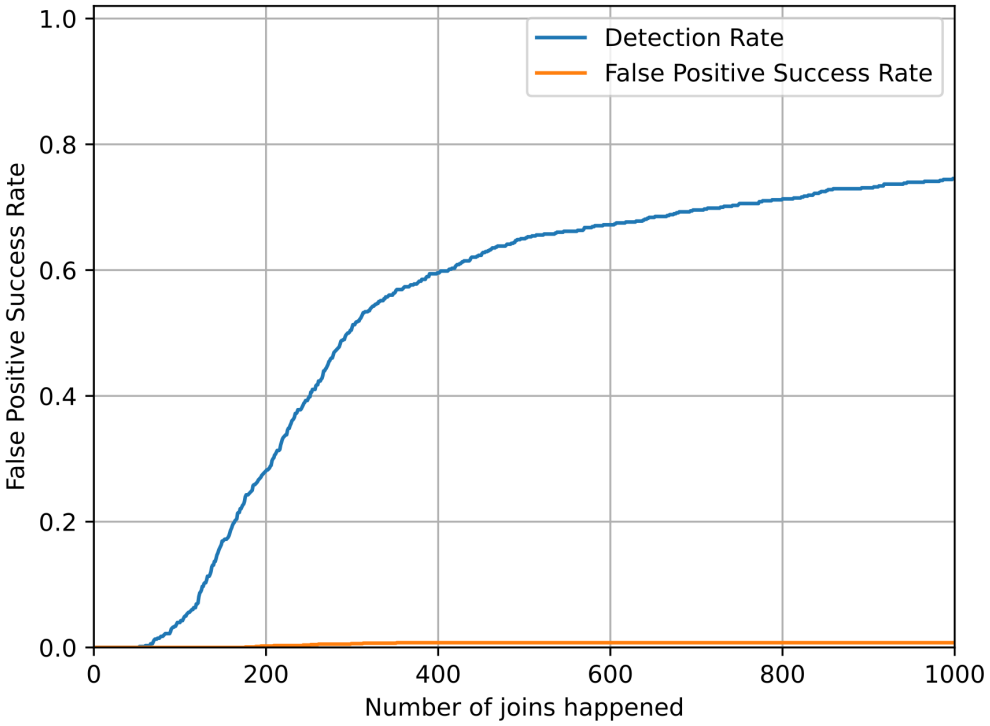


(b) $T2 = 0.7$

Figure 6.11: Variation of the detection rate and false positive success for $I_{FP} = 22\%$ and $I_{OF} = 20\%$ with different values of $T2$.



(a) $T2 = 0.8$



(b) $T2 = 0.6$

Figure 6.12: Variation of the detection rate and false positive success for $I_{FP} = 33\%$ and $I_{OF} = 30\%$ with different values of $T2$.

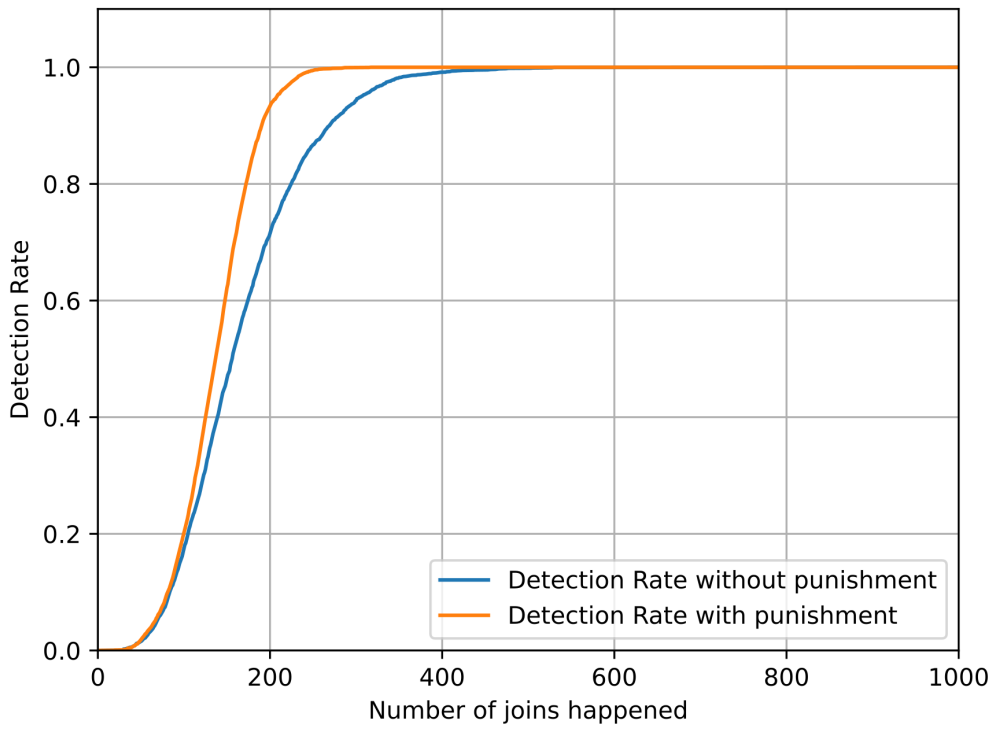


Figure 6.13: Variation of the detection rates while punishing the detected malicious nodes (T1=5 T2=0.5).

in order to ensure that our detection system works properly. $T1$ must be chosen according to the scale of the network. It depends on the number of nodes in the network and the expected number of nodes to join it. Therefore, it depends on the probability that a node is selected to be a Proxy node. $T2$ must be chosen according to the type of attacks that are expected to be conducted against the detection system and their intensities. The theoretical analysis and the experiments showed that for an ON-OFF attack, $T2$ must be the higher possible for a better detection of malicious nodes. The rule $T2 > (1 - (I_{OF} - f))$ described in Section 5.4.2 must be respected for proper functioning of the detection system. Therefore, the intensity of ON-OFF attack I_{OF} can be known according the historical behaviour of the nodes in the network. For a False Positive attack, $T2$ must be the smallest possible for a lower success of the False Positive attack. The rule $T2 < 1 - I_{FP}$ described in Section 5.4.3 must be respected to ensure that the system is robust against False Positive attacks. The intensity of the False Positive attack I_{FP} can be known according to the historical rate of malicious nodes that join the network.

In the case a combination of attacks is expected to be conducted against the detection system, the value of $T2$ must be tuned in way to make the detection system more robust against the riskier type of attack expected. For example, if an ON-OFF attack and a False Positive attack are expected to be conducted together, and if it is more important for the network to be robust against the False Positive attacks in order to keep the maximum of honest nodes running, then the value of $T2$ must be tuned to its lower bound.

Finally, the proposed detection system aims to detect the maximum number of malicious nodes and to reduce to the maximum the False Positive success against the honest nodes. Like any detection system, it can not ensure 100 % of safety in a network, but it aims to punish the malicious nodes in the network to conduct the less possible of attacks in a joining phase.

6.9 Summary

In this chapter, we have conducted experiments within a distributed system to further evaluate the performance of our solution. We have assessed the resilience of our solution by examining its performance in various network conditions and under different detection system parameters. We have considered a range of attack types in our study to ensure comprehensive

testing. The simulation results align with the theoretical analysis presented in chapter 5 and demonstrate the robustness of our detection system against attacks.

7 Conclusion

7.1 Conclusion

This thesis has addressed the critical issue of securing the joining phase in IIoT networks. This phase is particularly vulnerable to various security threats, and our research has focused on mitigating these risks. We have proposed two key contributions to enhance the security of IIoT networks.

In the first part of our research, we introduced a novel mutual authentication and key establishment protocol for IIoT networks. This solution addresses the limitations of pre-shared keys (PSKs) by providing a robust and autonomous approach. Our protocol ensures that the network coordinator and joining nodes can mutually authenticate without the need for PSKs. End device authentication relies on certificates, while coordinator authentication employs a lightweight consensus mechanism based on Shamir Secret Sharing. This innovative approach not only eliminates the requirement for pre-configured keys but also enhances the overall security of IIoT networks.

A comprehensive evaluation of our security protocol was carried out, focusing on its performance within the 6TiSCH framework, particularly with regard to communication costs and cryptographic operations. Our scheme was implemented using the Contiki-NG operating system, renowned for its robust 6TiSCH network stack. To emulate real-world IoT device constraints, we employed the Zolertia Z1 platform. The evaluation took place in a controlled

environment through simulation using the Cooja network simulator. We assessed our solution in both Global and Local mode scenarios, exploring different placements for the network's coordinator, whether at the center of the grid or in a corner. A key point of comparison was the default authentication mechanism in 6TiSCH. In summary, our solution brings substantial added value when compared to CoJP. It provides robust security without compromising communication efficiency, latency, or energy consumption—striking a vital balance for successful IIoT network deployments in dynamic, large-scale environments.

In the second part of our research, we concentrated on securing the joining phase against malicious proxy nodes. These nodes play a critical role in the communication process between new nodes and the network coordinator. To safeguard this phase, we introduced a robust detection system centralized at the coordinator. This system relies on participation logs of nodes serving as proxy nodes, their honest involvement, and the number of reports received from joining nodes concerning these proxy nodes. The solution considers multiple types of attacks and is designed to identify malicious proxy nodes effectively. Our theoretical evaluation, as well as practical experiments, demonstrated the robustness of this system and its ability to resist attacks originating from both proxy nodes and joining nodes.

In conclusion, this thesis has provided significant contributions to enhancing the security of IIoT networks, particularly during the critical joining phase. Our solutions not only address current security challenges but also lay the foundation for future work in securing IoT networks comprehensively. We have achieved autonomous mutual authentication, eliminated the need for pre-shared keys, and introduced an effective detection system for malicious proxy nodes. The practical applications and theoretical analyses have shown that our solutions are resilient, efficient, and adaptable to the dynamic and large-scale nature of IIoT networks.

7.2 Perspectives

Throughout this thesis, we have concentrated our work on securing the joining phase in IIoT. We started by proposing a mutual authentication scheme based on a consensus. The two main vulnerabilities of this solution are the following: the existence of malicious nodes between the proxy nodes, and a malicious behaviour of a compromised joining node after the join phase.

Then, as a continuity for the first proposal, we proposed a malicious proxy node detection in order to make the joining phase more robust and safe. However, this detection system must be more robust in order to cover the detection of malicious joining nodes after their join. In a large scale and dynamic IIoT, verifying the legitimacy of a joining node based on its firmware certificate is vulnerable to the cases where compromised nodes exist between trusted type of nodes. Therefore, we visualise a clear perspective for our future works. First, our objective is to enhance the capabilities of this detection system to address a broader spectrum of malicious activities within a network. While our initial solution focuses on detection during the joining phase, we aspire to introduce a comprehensive solution capable of identifying malicious nodes across various phases and considering diverse types of attacks. Our goal is to design a detection system that establishes a connection between multiple layers of an IoT network, thereby strengthening its robustness. To achieve this robustness, the behavior monitoring component of the detection system must be founded on multiple elements and factors, primarily tied to the specific application and role of each node in the network.
Additionally, in the context of the 6TiSCH protocol's security framework, one significant issue is the lack of a key revocation mechanism. When a node successfully joins a 6TiSCH network, it is furnished with various parameters, notably the link layer key, which grants access to shared network information. However, a critical shortcoming arises when a node leaves the network, as the protocol does not specify a method for revoking this key. The absence of such a mechanism poses a serious challenge in preserving the network's forward secrecy, a vital aspect of security, underscoring the urgent necessity for its development. Moreover, we proposed in this thesis a punishment mechanism adapted for 6TiSCH after the detection of a malicious proxy node. This punishment mechanism consists of excluding malicious nodes from playing the role of a proxy node in further joining phases. In the case where a more general detection system is proposed, the punishment mechanism must therefore consist of revoking the link-layer key from the node to be excluded. Therefore, this second solution is of high importance in maintaining a network safety.

Bibliography

[AAJ20] Zahrah A Almusaylim, Abdulaziz Alhumam, and NZ Jhanjhi. “Proposing a secure RPL based internet of things routing protocol: a review”. In: *Ad Hoc Networks* 101 (2020), p. 102096.

[Abb+21] Sohail Abbas et al. “Blockchain-based authentication in internet of vehicles: A survey”. In: *Sensors* 21.23 (2021), p. 7927.

[Abh+18] Nalam Venkata Abhishek et al. “An intrusion detection system for detecting compromised gateways in clustered IoT networks”. In: *2018 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*. IEEE. 2018, pp. 1–6.

[Agr+22] Shaashwat Agrawal et al. “Federated learning for intrusion detection system: Concepts, challenges and future directions”. In: *Computer Communications* (2022).

[Ala+17a] Mussab Alaa et al. “A review of smart home applications based on Internet of Things”. In: *Journal of network and computer applications* 97 (2017), pp. 48–65.

[Ala+17b] Fadele Ayotunde Alaba et al. “Internet of Things security: A survey”. In: *Journal of Network and Computer Applications* 88 (2017), pp. 10–28.

[Ash+23] Fatma Foad Ashrif et al. “Survey on the authentication and key agreement of 6LoWPAN: Open issues and future direction”. In: *Journal of Network and Computer Applications* (2023), p. 103759.

[ATW19] Mohammed Alshahrani, Issa Traore, and Isaac Woungang. “Anonymous mutual IoT interdevice authentication and key agreement scheme based on the ZigBee technique”. In: *Internet of Things* 7 (2019), p. 100061.

[AZ20] Fatimah Hussain Al-Naji and Rachid Zagrouba. “A survey on continuous authentication methods in Internet of Things environment”. In: *Computer Communications* 163 (2020), pp. 109–133.

[Bar+19] Mario Barbareschi et al. “A PUF-based mutual authentication scheme for cloud-edges IoT systems”. In: *Future Generation Computer Systems* 101 (2019), pp. 246–261.

[Bau+16] Johannes Bauer et al. “ECDSA on things: IoT integrity protection in practise”. In: *International conference on information and communications security*. Springer. 2016, pp. 3–17.

[BEK14] Carsten Bormann, Mehmet Ersue, and Ari Keranen. *Terminology for constrained-node networks*. Tech. rep. 2014.

[BG15] Anna L Buczak and Erhan Guven. “A survey of data mining and machine learning methods for cyber security intrusion detection”. In: *IEEE Communications surveys & tutorials* 18.2 (2015), pp. 1153–1176.

[Bou+20] Yassine Boufenneche et al. “Network formation in 6TiSCH industrial internet of things under misbehaved nodes”. In: *2020 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*. IEEE. 2020, pp. 1–6.

[Boy+18] Hugh Boyes et al. “The industrial internet of things (IIoT): An analysis framework”. In: *Computers in industry* 101 (2018), pp. 1–12.

[Bra18] An Braeken. “PUF based authentication protocol for IoT”. In: *Symmetry* 10.8 (2018), p. 352.

[BWH18] Elhadj Benkhelifa, Thomas Welsh, and Walaa Hamouda. “A critical review of practices and challenges in intrusion detection systems for IoT: Toward universal and resilient systems”. In: *IEEE communications surveys & tutorials* 20.4 (2018), pp. 3496–3509.

[CAS21] Krishna Keerthi Chennam, Rajanikanth Aluvalu, and S Shitharth. “An authentication model with high security for cloud database”. In: *Architectural wireless networks solutions and security issues* (2021), pp. 13–25.

[Cen+21] Marco Centenaro et al. “A survey on technologies, standards and open challenges in satellite IoT”. In: *IEEE Communications Surveys & Tutorials* 23.3 (2021), pp. 1693–1720.

[Cer+15] Christian Cervantes et al. “Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things”. In: *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE. 2015, pp. 606–611.

[Cha+19] Nadia Chaabouni et al. “Network intrusion detection for IoT security based on learning techniques”. In: *IEEE Communications Surveys & Tutorials* 21.3 (2019), pp. 2671–2701.

[Che+19] Songlin Chen et al. “Radio frequency fingerprint-based intelligent mobile edge computing for internet of things authentication”. In: *Sensors* 19.16 (2019), p. 3610.

[CL+99] Miguel Castro, Barbara Liskov, et al. “Practical byzantine fault tolerance”. In: *OsDI*. Vol. 99. 1999. 1999, pp. 173–186.

[CL21] Chien-Ming Chen and Shuangshuang Liu. “Improved secure and lightweight authentication scheme for next-generation IOT infrastructure”. In: *Security and Communication Networks* 2021 (2021), pp. 1–13.

[Col+18] José Francisco Colom et al. “Scheduling framework for distributed intrusion detection systems over heterogeneous network architectures”. In: *Journal of Network and Computer Applications* 108 (2018), pp. 76–86.

[Con+18] Mauro Conti et al. *Internet of Things security and forensics: Challenges and opportunities*. 2018.

[Cui+23] Jie Cui et al. “Multi-factor based session secret key agreement for the Industrial Internet of Things”. In: *Ad Hoc Networks* 138 (2023), p. 102997.

[Da +19] Kelton AP Da Costa et al. “Internet of Things: A survey on machine learning-based intrusion detection approaches”. In: *Computer Networks* 151 (2019), pp. 147–157.

[Das+18] Rajshekhar Das et al. “A deep learning approach to IoT authentication”. In: *2018 IEEE international conference on communications (ICC)*. IEEE. 2018, pp. 1–6.

[DC18] Abebe Abeshu Diro and Naveen Chilamkurti. "Distributed attack detection scheme using deep learning approach for Internet of Things". In: *Future Generation Computer Systems* 82 (2018), pp. 761–768.

[DR21] D Dujovne and M Richardson. "RFC 9032 Encapsulation of 6TiSCH Join and Enrollment Information Elements". In: (2021).

[Dun+07] Adam Dunkels et al. "Software-Based on-Line Energy Estimation for Sensor Nodes". In: *Proceedings of the 4th Workshop on Embedded Networked Sensors*. EmNets '07. Cork, Ireland: Association for Computing Machinery, 2007, pp. 28–32. ISBN: 9781595936943. DOI: 10.1145/1278972.1278979. URL: https://doi.org/10.1145/1278972.1278979.

[Duq18] Simon Duquennoy. *https://docs.contiki-ng.org/en/develop/doc/programming/TSCH-and-6TiSCH.html*. Tech. rep. 2018.

[EC20] Ike C Ehie and Michael A Chilton. "Understanding the influence of IT/OT Convergence on the adoption of Internet of Things (IoT) in manufacturing organizations: An empirical investigation". In: *Computers in Industry* 115 (2020), p. 103166.

[El-+19] Mohammed El-Hajj et al. "A Survey of Internet of Things (IoT) Authentication Schemes". In: *Sensors* 19.5 (2019), p. 1141.

[Esf+17] Alireza Esfahani et al. "A lightweight authentication mechanism for M2M communications in industrial IoT environment". In: *IEEE Internet of Things Journal* 6.1 (2017), pp. 288–296.

[Ge+19] Mengmeng Ge et al. "Deep learning-based intrusion detection for IoT networks". In: *2019 IEEE 24th pacific rim international symposium on dependable computing (PRDC)*. IEEE. 2019, pp. 256–25609.

[Ger+23] Apostolos Gerodimos et al. "IoT: Communication protocols and security threats". In: *Internet of Things and Cyber-Physical Systems* (2023).

[GZC19] Xinghao Guo, Zhen Zhang, and Jie Chang. "Survey of mobile device authentication methods based on RF fingerprint". In: *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 2019, pp. 1–6.

[Haj+19] Somayye Hajiheidari et al. “Intrusion detection systems in the Internet of things: A comprehensive investigation”. In: *Computer Networks* 160 (2019), pp. 165–191.

[Haj+23] Ali Haj-Hassan et al. “Consensus-based mutual authentication scheme for Industrial IoT”. In: *Ad Hoc Networks* 145 (2023), p. 103162.

[Has+19a] Wan Haslina Hassan et al. “Current research on Internet of Things (IoT) security: A survey”. In: *Computer networks* 148 (2019), pp. 283–294.

[Has+19b] Vikas Hassija et al. “A survey on IoT security: application areas, security threats, and solution architectures”. In: *IEEE Access* 7 (2019), pp. 82721–82743.

[HHN20] Ali Haj-Hassan, Carol Habib, and Jad Nassar. “Real-time spatio-temporal based outlier detection framework for wireless body sensor networks”. In: *2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE. 2020, pp. 1–6.

[HQS19] Jianwei Hou, Leilei Qu, and Wenchang Shi. “A survey on internet of things security from data perspectives”. In: *Computer Networks* 148 (2019), pp. 295–306.

[Hum+17] Abdulmalik Humayed et al. “Cyber-physical systems security—A survey”. In: *IEEE Internet of Things Journal* 4.6 (2017), pp. 1802–1831.

[Hus+13] Hassen Redwan Hussen et al. “SAKES: Secure authentication and key establishment scheme for M2M communication in the IP-based wireless sensor network (6L0WPAN)”. In: *2013 Fifth international conference on ubiquitous and future networks (ICUFN)*. IEEE. 2013, pp. 246–251.

[Hus+22] Saddam Hussain et al. “Certificateless signature schemes in Industrial Internet of Things: A comparative survey”. In: *Computer Communications* 181 (2022), pp. 116–131.

[Imt+21] Ahmed Imteaj et al. “A survey on federated learning for resource-constrained IoT devices”. In: *IEEE Internet of Things Journal* 9.1 (2021), pp. 1–24.

[Ist+21] Kazi Istiaque Ahmed et al. “Machine learning for authentication and authorization in iot: Taxonomy, challenges and future research direction”. In: *Sensors* 21.15 (2021), p. 5122.

[Jan+14] Mian Ahmad Jan et al. “A Robust Authentication Scheme for Observing Resources in the Internet of Things Environment”. In: *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE. 2014, pp. 205–211.

[JJK22] Anu Jagannath, Jithin Jagannath, and Prem Sagar Pattanshetty Vasanth Kumar. “A comprehensive survey on radio frequency (rf) fingerprinting: Traditional approaches, deep learning, and open challenges”. In: *Computer Networks* 219 (2022), p. 109455.

[Jou+23] Mohammed Jouhari et al. “A survey on scalable LoRaWAN for massive IoT: Recent advances, potentials, and challenges”. In: *IEEE Communications Surveys & Tutorials* (2023).

[Kal+22] Alakesh Kalita et al. “Effect of DIS attack on 6TiSCH network formation”. In: *IEEE Communications Letters* 26.5 (2022), pp. 1190–1193.

[Kas+13] Prabhakaran Kasinathan et al. “Denial-of-Service detection in 6LoWPAN based Internet of Things”. In: *2013 IEEE 9th international conference on wireless and mobile computing, networking and communications (WiMob)*. IEEE. 2013, pp. 600–607.

[KBL18] Djamel Eddine Kouicem, Abdelmadjid Bouabdallah, and Hicham Lakhlef. “Internet of things security: A top-down survey”. In: *Computer Networks* 141 (2018), pp. 199–221.

[KH17] Zeeshan Ali Khan and Peter Herrmann. “A trust based distributed intrusion detection mechanism for internet of things”. In: *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*. IEEE. 2017, pp. 1169–1176.

[Kha+20] Umair Khalid et al. “A decentralized lightweight blockchain-based authentication mechanism for IoT systems”. In: *Cluster Computing* 23.3 (2020), pp. 2067–2087.

[Kha+22] Alaa O Khadidos et al. “An intelligent security framework based on collaborative mutual authentication model for smart city networks”. In: *IEEE Access* 10 (2022), pp. 85289–85304.

[Kie+21] Peter Kietzmann et al. “A Performance Study of Crypto-Hardware in the Low-End IoT”. In: *Proceedings of the 2021 International Conference on Embedded Wireless Systems and Networks*. EWSN ’21. Delft, The Netherlands: Junction Publishing, 2021, pp. 79–90.

[KJ23] Pooja Kumari and Ankit Kumar Jain. “A comprehensive study of DDoS attacks over IoT network and their countermeasures”. In: *Computers & Security* (2023), p. 103096.

[Kou+20] Djamel Eddine Kouicem et al. “Decentralized blockchain-based trust management protocol for the Internet of Things”. In: *IEEE Transactions on Dependable and Secure Computing* 19.2 (2020), pp. 1292–1306.

[KP17] Apurva S Kittur and Alwyn Roshan Pais. “Batch verification of digital signatures: approaches and challenges”. In: *Journal of information security and applications* 37 (2017), pp. 15–27.

[KPB19] S Karthikeyan, Rizwan Patan, and B Balamurugan. “Enhancement of security in the Internet of Things (IoT) by using X. 509 authentication mechanism”. In: *Recent Trends in Communication, Computing, and Electronics: Select Proceedings of IC3E 2018*. Springer. 2019, pp. 217–225.

[Kri+18] Kosmas Kritsis et al. “A tutorial on performance evaluation and validation methodology for low-power and lossy networks”. In: *IEEE Communications Surveys & Tutorials* 20.3 (2018), pp. 1799–1825.

[KRR15] Athar Ali Khan, Mubashir Husain Rehmani, and Martin Reisslein. “Cognitive radio for smart grids: Survey of architectures, spectrum sensing mechanisms, and networking protocols”. In: *IEEE Communications Surveys & Tutorials* 18.1 (2015), pp. 860–898.

[KS18] Minhaj Ahmad Khan and Khaled Salah. “IoT security: Review, blockchain solutions, and open challenges”. In: *Future generation computer systems* 82 (2018), pp. 395–411.

[Kum+22] Ashish Kumar et al. “A comprehensive survey of authentication methods in Internet-of-Things and its conjunctions”. In: *Journal of Network and Computer Applications* 204 (2022), p. 103414.

[Lag+21] Asif Ali Laghari et al. “A review and state of art of Internet of Things (IoT)”. In: *Archives of Computational Methods in Engineering* (2021), pp. 1–19.

[Lee+14] Tsung-Han Lee et al. “A lightweight intrusion detection scheme based on energy consumption analysis in 6LowPAN”. In: *Advanced Technologies, Embedded and Multimedia for Human-centric Computing: HumanCom and EMC 2013*. Springer. 2014, pp. 1205–1213.

[Li+18] Dongxing Li et al. “A blockchain-based authentication and security mechanism for IoT”. In: *2018 27th International Conference on Computer Communication and Networks (ICCCN)*. IEEE. 2018, pp. 1–6.

[LSP19] Leslie Lamport, Robert Shostak, and Marshall Pease. “The Byzantine generals problem”. In: *Concurrency: the works of leslie lamport*. 2019, pp. 203–226.

[Lu17] Yang Lu. “Industry 4.0: A survey on technologies, applications and open research issues”. In: *Journal of industrial information integration* 6 (2017), pp. 1–10.

[LY21] Wassila Lalouani and Mohamed Younis. “Robust distributed intrusion detection system for edge of things”. In: *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2021, pp. 01–06.

[Mac] Ken MacKay. *micro-ecc GitHub repository*. https://github.com/kmackay/micro-ecc, last accessed December 7th, 2022.

[Mal+22] Priyanka Mall et al. “PUF-based authentication and key agreement protocols for IoT, WSNs and smart grids: a comprehensive survey”. In: *IEEE Internet of Things Journal* (2022).

[Mam+21] Moustafa Mamdouh et al. “Authentication and Identity Management of IoHT Devices: Achievements, Challenges, and Future Directions”. In: *Computers & Security* 111 (2021), p. 102491.

[Min+22] Yang Ming et al. “A Secure One-to-Many Authentication and Key Agreement Scheme for Industrial IoT”. In: *IEEE Systems Journal* (2022).

[MNC20] Reem Melki, Hassan N Noura, and Ali Chehab. “Lightweight multi-factor mutual authentication protocol for IoT devices”. In: *International Journal of Information Security* 19.6 (2020), pp. 679–694.

[Nae+23] Muhammad Naeem et al. “Modelling and Analysis of a Sigfox based IoT Network using UPPAAL SMC”. In: *IEEE Sensors Journal* (2023).

[Naz+21] Rashid Nazir et al. “Survey on wireless network security”. In: *Archives of Computational Methods in Engineering* (2021), pp. 1–20.

[Nes+19] Nataliia Neshenko et al. “Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on Internet-scale IoT exploitations”. In: *IEEE Communications Surveys & Tutorials* 21.3 (2019), pp. 2702–2733.

[Oik+22] George Oikonomou et al. “The Contiki-NG open source operating system for next generation IoT devices”. In: *SoftwareX* 18 (2022), p. 101089.

[OKR14] Doohwan Oh, Deokho Kim, and Won Woo Ro. “A malicious pattern detection engine for embedded security systems in the Internet of Things”. In: *Sensors* 14.12 (2014), pp. 24188–24211.

[Ome+21] Aleksandr Ometov et al. “A survey on wearable technology: History, state-of-the-art and current challenges”. In: *Computer Networks* 193 (2021), p. 108074.

[Ost+06] Fredrik Osterlind et al. “Cross-level sensor network simulation with cooja”. In: *Proceedings. 2006 31st IEEE conference on local computer networks*. IEEE. 2006, pp. 641–648.

[Por+14] Pawani Porambage et al. “Two-phase authentication protocol for wireless sensor networks in distributed IoT applications”. In: *2014 IEEE Wireless Communications and Networking Conference (WCNC)*. Ieee. 2014, pp. 2728–2733.

[Pra+23] Michal Prauzek et al. “IoT Sensor Challenges for Geothermal Energy Installations Monitoring: A Survey”. In: *Sensors* 23.12 (2023), p. 5577.

[QSS21] Xiaoying Qiu, Xuan Sun, and Xiameng Si. “Machine Learning-Based Security Authentication for IoT Networks”. In: *Smart Grid and Internet of Things: 4th EAI International Conference, SGIoT 2020, TaiChung, Taiwan, December 5–6, 2020, Proceedings*. Springer. 2021, pp. 106–115.

[Rah+20] Sawsan Abdul Rahman et al. “Internet of things intrusion detection: Centralized, on-device, or federated learning?” In: *IEEE Network* 34.6 (2020), pp. 310–317.

[RHM19] Rizwan Hamid Randhawa, Abdul Hameed, and Adnan Noor Mian. "Energy efficient cross-layer approach for object security of CoAP for IoT devices". In: *Ad Hoc Networks* 92 (2019), p. 101761.

[RWV13] Shahid Raza, Linus Wallgren, and Thiemo Voigt. "SVELTE: Real-time intrusion detection in the Internet of Things". In: *Ad hoc networks* 11.8 (2013), pp. 2661–2674.

[Sal+18] Ola Salman et al. "IoT survey: An SDN and fog computing perspective". In: *Computer Networks* 143 (2018), pp. 221–246.

[San+18] Ramon Sanchez-Iborra et al. "Enhancing lorawan security through a lightweight and authenticated key management approach". In: *Sensors* 18.6 (2018), p. 1833.

[SBA21] Shivam Saxena, Bharat Bhushan, and Mohd Abdul Ahad. "Blockchain based solutions to secure IoT: Background, integration trends and a way forward". In: *Journal of Network and Computer Applications* 181 (2021), p. 103050.

[SCL20] Rahul Sharma, Chien Aun Chan, and Christopher Leckie. "Evaluation of centralised vs distributed collaborative intrusion detection systems in multi-access edge computing". In: *2020 IFIP Networking Conference (Networking)*. IEEE. 2020, pp. 343–351.

[SCM21] Gustavo A Nunez Segura, Arsenia Chorti, and Cintia Borges Margi. "Centralized and distributed intrusion detection for resource-constrained wireless SDN networks". In: *IEEE Internet of Things Journal* 9.10 (2021), pp. 7746–7758.

[Sel+19a] Göran Selander et al. *Object Security for Constrained RESTful Environments (OSCORE)*. RFC 8613. July 2019. DOI: 10.17487/RFC8613. URL: https://www.rfc-editor.org/info/rfc8613.

[Sel+19b] Göran Selander et al. *Object security for constrained restful environments (oscore)*. Tech. rep. 2019.

[Sha+18] Kewei Sha et al. "On security challenges and open issues in Internet of Things". In: *Future generation computer systems* 83 (2018), pp. 326–337.

[Sha+20a] Akasha Shafiq et al. "An identity-based anonymous three-party authenticated protocol for iot infrastructure". In: *Journal of Sensors* 2020 (2020), pp. 1–17.

[Sha+20b] Alireza Shamsoshoara et al. “A survey on physical unclonable function (PUF)-based security solutions for Internet of Things”. In: *Computer Networks* 183 (2020), p. 107593.

[Sha79] Adi Shamir. “How to Share a Secret”. In: *Commun. ACM* 22.11 (Nov. 1979), pp. 612–613. ISSN: 0001-0782.

[Sis+18] Emiliano Sisinni et al. “Industrial internet of things: Challenges, opportunities, and directions”. In: *IEEE transactions on industrial informatics* 14.11 (2018), pp. 4724–4734.

[Son+20] Yanxing Song et al. “Applications of the Internet of Things (IoT) in smart logistics: A comprehensive survey”. In: *IEEE Internet of Things Journal* 8.6 (2020), pp. 4250–4274.

[Soo+18] Mehdi Sookhak et al. “Security and privacy of smart cities: a survey, research issues and challenges”. In: *IEEE Communications Surveys & Tutorials* 21.2 (2018), pp. 1718–1743.

[SPK+21] Kaumudi Singh, TV Prabhakar, Joy Kuri, et al. “Quick and efficient network access schemes for IoT devices”. In: *Ad Hoc Networks* 115 (2021), p. 102435.

[SS17] Fatih Sakiz and Sevil Sen. “A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV”. In: *Ad Hoc Networks* 61 (2017), pp. 33–50.

[SS98] Joseph H Silverman and Joe Suzuki. “Elliptic curve discrete logarithms and the index calculus”. In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 1998, pp. 110–125.

[Sta18] Alex Stanoev. *https://docs.contiki-ng.org/en/release-v4.9/doc/tutorials/RAM-and-ROM-usage.html*. Tech. rep. 2018.

[SV18] Trusit Shah and Subbarayan Venkatesan. “Authentication of IoT device and IoT server using secure vaults”. In: *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE. 2018, pp. 819–824.

[TAA+23] Rabi Tanash, Mahmoud AlQudah, Salem Al-Agtash, et al. “Enhancing energy efficiency of IEEE 802.15. 4-based industrial wireless sensor networks”. In: *Journal of Industrial Information Integration* 33 (2023), p. 100460.

[Tia+22] Chuang Tian et al. “Reliable PUF-based mutual authentication protocol for UAVs towards multi-domain environment”. In: *Computer Networks* 218 (2022), p. 109421.

[Tsi+21] Konstantinos Tsiknas et al. “Cyber threats to industrial IoT: a survey on attacks and countermeasures”. In: *IoT* 2.1 (2021), pp. 163–186.

[US20] Silvia Liberata Ullo and Ganesh Ram Sinha. “Advances in smart environment monitoring systems using IoT and sensors”. In: *Sensors* 20.11 (2020), p. 3113.

[Vil+19] Xavier Vilajosana et al. “Ietf 6tisch: A tutorial”. In: *IEEE Communications Surveys & Tutorials* 22.1 (2019), pp. 595–615.

[Vog+18] Elvis Vogli et al. “Fast network joining algorithms in industrial IEEE 802.15. 4 deployments”. In: *Ad Hoc Networks* 69 (2018), pp. 65–75.

[VSO17] Eduardo K Viegas, Altair O Santin, and Luiz S Oliveira. “Toward a reliable anomaly-based intrusion detection in real-world environments”. In: *Computer Networks* 127 (2017), pp. 200–216.

[Vuč+21a] Mališa Vučinić et al. *Constrained Join Protocol (CoJP) for 6TiSCH*. RFC 9031. May 2021. DOI: 10.17487/RFC9031.

[Vuč+21b] Mališa Vučinić et al. “RFC9031: Constrained Join Protocol (CoJP) for 6TiSCH”. In: *Internet Engineering Task Force* RFC9031 (2021).

[Wan+19] Xiaoliang Wang et al. “An improved authentication scheme for internet of vehicles based on blockchain technology”. In: *IEEE access* 7 (2019), pp. 45061–45072.

[Wan+21] Weizheng Wang et al. “Blockchain-based reliable and efficient certificateless signature for IIoT devices”. In: *IEEE transactions on industrial informatics* 18.10 (2021), pp. 7059–7067.

[Xia+16] Liang Xiao et al. “PHY-layer spoofing detection with reinforcement learning in wireless networks”. In: *IEEE Transactions on Vehicular Technology* 65.12 (2016), pp. 10037–10047.

[Xu+18] Hansong Xu et al. “A survey on industrial Internet of Things: A cyber-physical systems perspective”. In: *Ieee access* 6 (2018), pp. 78238–78259.

[Yan+17] Yuchen Yang et al. “A Survey on Security and Privacy Issues in Internet-of-Things”. In: *IEEE Internet of Things Journal* 4.5 (2017), pp. 1250–1258.

[Yan+21] Xing Yang et al. “A survey on smart agriculture: Development modes, technologies, and security and privacy challenges”. In: *IEEE/CAA Journal of Automatica Sinica* 8.2 (2021), pp. 273–302.

[Zar+17] Bruno Bogaz Zarpelão et al. “A survey of intrusion detection in Internet of Things”. In: *Journal of Network and Computer Applications* 84 (2017), pp. 25–37.

[Zha+23] Junqing Zhang et al. “Radio Frequency Fingerprint Identification for Device Authentication in the Internet of Things”. In: *IEEE Communications Magazine* (2023).

[Zhe+22] Yue Zheng et al. “PUF-based mutual authentication and key exchange protocol for peer-to-peer IoT applications”. In: *IEEE Transactions on Dependable and Secure Computing* (2022).

[Zho+20] Man Zhou et al. “Distributed collaborative intrusion detection system for vehicular Ad Hoc networks based on invariant”. In: *Computer Networks* 172 (2020), p. 107174.

[Zoh+23] Alireza Zohourian et al. “IoT Zigbee device security: A comprehensive review”. In: *Internet of Things* (2023), p. 100791.

[Zuh+17] Fatima Tul Zuhra et al. “Routing protocols in wireless body sensor networks: A comprehensive survey”. In: *Journal of Network and Computer Applications* 99 (2017), pp. 73–97.