UNIVERSITY OF THE AEGEAN
SCHOOL OF ENGINEERING

DEPARTMENT OF INFORMATION AND COMMINICATIONS SYSTEMS ENGINEERING

POSTGRADUATE MASTER'S PROGRAMME

Information and Communication Systems Security

Trust Modelling Methodology for Distributed Environments

MASTER'S THESIS

by

KROUSARLIS THOMAS

Supervisor:
Kokolakis Spyros

Members of the examination committee:
Diamantopoulou Vasiliki
Stergiopoulos George

Samos, January 2024

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΑΣΦΑΛΕΙΑ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Μεθοδολογία μοντελοποίησης εμπιστοσύνης σε κατανεμημένα περιβάλλοντα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΚΡΟΥΣΑΡΛΗ ΘΩΜΑ

Επιβλέπων:
Κοκολάκης Σπύρος

Μέλη εξεταστικής επιτροπής:
Γεώργιος Στεργιόπουλος
Βασιλική Διαμαντοπούλου

Σάμος, Ιανουάριος 2024

This page is intentionally left blank.
Η σελίδα αυτή είναι σκόπιμα λευκή.

Table of Contents

List of Figures

List of Tables

Acronyms

CPU	Central Processing Unit
CVE	Common Vulnerabilities and Exposures
EDR	Endpoint Detection and Response
FW	Firmware
HMAC	Hash-based Message Authentication Code
HSM	Hardware Security Module
HVAC	Heating, Ventilation, and Air Conditioning
ID(P)S	Intrusion Prevention (and Detection) System
IoT	Internet of Things
IT	Information Technology
ITU	International Telecommunication Union
NIST	National Institute of Standards and Technology
OS	Operating System
OT	Operational technology
PSK	Pre-Shared Key
PUF	Physical Unclonable Functions
QoS	Quality of Service
SIEM	Security Information and Event Management
SoS	System of Systems
SW	Software
TLS/SSL	Transport Layer Security / Secure Sockets Layer
TPM	Trusted Platform Module
UML	Unified Modelling Language

Abstract

Devices and autonomous systems have become an integral part of our daily lives, undergoing significant transformations to become increasingly intelligent and able to connect to other systems and process complex information. Particularly in the case of autonomous systems, they are capable of learning and adapting themselves based on the environment and their interactions with other systems. This raises a significant challenge of building trust relationships between them. Still, there is a research question in how to demonstrate the trustworthy state of a system and how this state could be quantified in a way that systems and devices, which lack the natural ability to assess relationships and risks like humans, can understand. This thesis reviews the fundamentals of existing trust modelling and analyses current practices. A major contribution is the development of a trust modelling methodology, including specific steps for quantifying trust, identifying trust entities, assessing trust relationships within a system or network, and defining trust criteria and properties. It also provides a formula for quantifying trustworthiness. The study identifies potential trust properties that can be measured in each system, taking into account various dimensions such as knowledge, integrity, behaviour, and experience of other entities in the network. It also examines trust provisioning techniques to measure trust properties and explores three trust management techniques tailored to specific system requirements for more efficient management and trustworthiness quantification. This involves considering how trust properties are collected and the decentralization aspects of the quantification process. Additionally, the thesis assesses the applicability of the trust assessment solution through two specific scenarios in the railway domain. It also discusses the limitations of its applicability and concludes with suggestions for potential research directions and future work.

Keywords: *trust modelling, trust properties, trustworthiness quantification and trust models*

Περίληψη

Οι συσκευές και τα αυτόνομα συστήματα έχουν γίνει αναπόσπαστο μέρος της καθημερινότητας, ενώ τα τελευταία χρόνια παρατηρούνται σημαντικές αλλαγές ώστε να γίνονται όλο και πιο ευφυή και ικανά να συνδέονται με άλλα συστήματα καθώς και να επεξεργάζονται σύνθετες πληροφορίες. Ιδιαίτερα στην περίπτωση των αυτόνομων συστημάτων, είναι ικανά να μαθαίνουν και να προσαρμόζονται με βάση το περιβάλλον και τις αλληλεπιδράσεις τους με άλλα. Αυτό δημιουργεί μια σημαντική πρόκληση για την οικοδόμηση σχέσεων εμπιστοσύνης μεταξύ αυτών. Ακόμα, υπάρχει το ερώτημα για το πώς θα καταδειχθεί η αξιόπιστη κατάσταση ενός συστήματος και πώς αυτή η κατάσταση θα μπορούσε να ποσοτικοποιηθεί με τρόπο που να μπορούν να κατανοήσουν οι συσκευές, οι οποίες δεν έχουν τη φυσική και νοητική ικανότητα να αξιολογούν σχέσεις και κινδύνους όπως οι άνθρωποι. Η παρούσα διπλωματική εργασία εξετάζει τις βασικές αρχές της μοντελοποίησης εμπιστοσύνης και αναλύει τις τρέχουσες πρακτικές στο πεδίο. Μια σημαντική συνεισφορά είναι η ανάπτυξη μιας μεθοδολογίας μοντελοποίησης εμπιστοσύνης, η οποία περιλαμβάνει συγκεκριμένα βήματα για τον ποσοτικό προσδιορισμό της εμπιστοσύνης, τον εντοπισμό οντοτήτων εμπιστοσύνης, την αξιολόγηση των σχέσεων εμπιστοσύνης εντός ενός συστήματος ή ενός δικτύου συσκευών και τον ορισμό κριτηρίων εμπιστοσύνης. Παρέχει επίσης έναν γενικευμένο τύπο για την ποσοτικοποίηση της αξιοπιστίας των συστημάτων. Ακόμη, η εργασία προσδιορίζει πιθανές ιδιότητες εμπιστοσύνης που μπορούν να μετρηθούν σε κάθε σύστημα, λαμβάνοντας υπόψη διάφορες πτυχές όπως η γνώση, η ακεραιότητα, η συμπεριφορά και η εμπειρία άλλων οντοτήτων στο δίκτυο. Η εργασία εξετάζει επίσης τεχνικές παροχής εμπιστοσύνης για τη μέτρηση των ιδιοτήτων αυτών και διερευνά τρεις τεχνικές διαχείρισης εμπιστοσύνης προσαρμοσμένες σε συγκεκριμένες δυνατότητες και απαιτήσεις των συστημάτων για την αποτελεσματικότερη διαχείριση και ποσοτικοποίηση της αξιοπιστίας. Αυτό περιλαμβάνει την εξέταση του τρόπου συλλογής των ιδιοτήτων καθώς και τη διαδικασίας ποσοτικοποίησης σε αποκεντρωμένα δίκτυα και περιβάλλοντα. Επιπλέον, η διπλωματική εργασία αξιολογεί την πρακτική εφαρμογή της λύσης μέσω δύο συγκεκριμένων σεναρίων στον τομέα των σιδηροδρόμων. Συζητά επίσης τους περιορισμούς της προτεινόμενης λύση και καταλήγει με προτάσεις για πιθανές ερευνητικές κατευθύνσεις και μελλοντικές εργασίες.

Λέξεις Κλειδιά: *μοντελοποίηση εμπιστοσύνης, ιδιότητες εμπιστοσύνης, μοντέλα εμπιστοσύνης και ποσοτικοποίηση*

1

Introduction

1.1 Background

In recent years, the technological landscape has undergone a significant (and rapid) transformation, reshaping the design and capabilities of systems and their underlying components. These advancements are mostly characterized by the integration of processing power and network interfaces into even the most basic system elements as there is a continuous shift towards edge computing [1], where data processing and service delivery are handled and offloaded to edge devices [2]. In the telecommunications domain, for instance, modern infrastructure supports the deployment of microservices directly at the network edge. At the same time, there are several examples in various industrial systems where real-time data processing and decision-making take place directly within the system, reducing latency and bandwidth use [3].

This evolution covers everything from simple sensors and actuators to more advanced computing units. Previously, devices primarily served as data collectors, relying mainly on external servers for processing and decision-making. Modern devices now integrate compact and energy-efficient processors capable of heavy computational tasks (e.g., ARM Cortex-M, Qualcomm Snapdragon). These processors are accompanied by embedded operating systems (and real-time operating systems in the case of safety-critical systems) to manage limited hardware resources effectively and enable the timely processing of critical tasks in applications requiring immediate response, such as industrial automation, autonomous vehicles and smart grids. On top of that, the vast majority of these devices are equipped with a broad set of network interfaces, such as Bluetooth and Wi-Fi, which enable direct communication among various system components.

This digital transformation is reshaping industries, introducing high-end products that enable bi-directional communication across a wide range of devices, equipment, hardware, and software

services. However, while promising, this evolution comes with its challenges since the increased intelligence and connectivity of these devices introduce new security concerns and challenges [4]. This ongoing digitalization and incorporation of embedded processing power and network interfaces further expand the attack surface: malicious actors can exploit vulnerabilities on the system level or, leveraging interconnectivity, can launch cascading attacks across multiple sub-systems. Such attacks were mostly impractical in the past due to the limited connectivity interfaces and the reliance on microcontrollers without processing units or embedded operating systems.

Moreover, most of these systems comprise several hardware modules supplied by multiple global vendors, creating a mixed environment of technologies within a single system. Even when dealing with reputable manufacturers, they typically lack specific knowledge about the final applications or the intended environment where their submodules will be integrated. Once the components leave the manufacturing floor, they are not aware of their actual security status. Thus, they can only design protective measures based on anticipated, rather than exact, usage scenarios and environments. This is also the case when entire products are delivered to clients without knowing or considering the target environment (e.g., network topology, other devices that will be connected, etc.). This means that runtime vulnerabilities cannot be detected efficiently by the manufacturer so that remediation measures can be applied directly during the post-deployment. This leads to a mixed-trust environment within larger systems, a situation that gets worse when the exact designers and manufacturers of the components remain unknown, as in several cases, large vendors design modules that are outsourced for production in third countries. This process typically involves multiple suppliers and other unknown manufacturers, resulting in a complex scenario where visibility into each step of the manufacturing process is not visible.

Adding to this fact that today's infrastructures consist of several heterogeneous subsystems and components, each with unique characteristics, including IT (devices and software for information storage and processing), Operational Technology (OT) infrastructure (devices and software for cyber-physical operations, mainly in the industry), network and communication systems, and their interdependencies with supply chains and other legacy systems. This synthesis leads to the creation of large Systems-of-Systems (SoS) offering services that must work together to execute critical functionalities cooperatively.

For instance, in the case of railways, the infrastructure's heterogeneity spans from heavy equipment with obsolete connectivity interfaces (i.e., interlocking systems or other auxiliary systems like heating, ventilation, and air conditioning - HVACs) and low capability embedded devices to high-end equipment and cloud services (i.e., autonomous driving and remote monitoring) [5]. A critical challenge arises from the fact that a wide range of manufacturers and vendors supply this equipment, complicating the task of ensuring trust between these components. Large equipment, often seen as SoS, comprises hardware parts and modules produced by numerous vendors, where diverse technologies coexist, thereby creating a complex environment. Additionally, differentiating between IT and OT is challenging in most cases as well. Often, identifying the assets needing protection is a complex process, categorising a device as IT or OT, and deciding on the appropriate security measures and standards to apply is also difficult.

Until recently, the conventional threat and trust models assumed that the hardware and software vulnerabilities are entirely independent, and existing security measures have been designed

separately (around software and networks rather than securing the system itself). Due to the hardware's nature, there is a misleading assumption that the underlying hardware is always trusted, particularly in OT systems. It is also considered that hardware vulnerabilities could be exploited only in situ with direct physical access to the system. However, the rich set of new and available connectivity interfaces and possible bypasses through running software makes it possible for an adversary to attack hardware modules entirely remotely. There are several documented attacks [6] that demonstrate how malicious actors exploit connectivity interfaces of secondary systems to compromise primary ones or take advantage of software vulnerabilities through supply chain attacks [7].

In the previous railway scenario, multiple on-board units are required to communicate and exchange information about various conditions. Specifically, the train control management system needs to interact with the automatic train protection systems while also gathering data from on-board sensors. In emergency situations, this system should promptly activate the braking machine interface and other protective systems. Additionally, communication with external systems, such as trackside units responsible for controlling signals and traffic management (e.g., interlocking), is required. A compromise in the signalling or automatic train control system could lead to severe train accidents. Such an incident took place in Poland in 2008, where an attacker successfully hacked a tram system [8]. However, it is essential to note that not all systems carry the same level of criticality. While important, systems like on-board heating, ventilation, and water management are considered less critical. This leads to a mixed-criticality environment of devices and services running on top of them that should be prioritized accordingly to achieve an overall trustworthy state.

Also, we need to consider that in most cases, these systems are characterized by their extended operational lifespans, often remaining active for several decades (i.e., track vacancy detector and wayside equipment). This durability poses unique security challenges. Systems deployed today must be designed considering future technological developments and emerging threats. Given their prolonged use, these systems are increasingly susceptible to evolving cybersecurity threats. As technology progresses, parts of older systems may become outdated, making the search for compatible replacements (hardware parts) or upgrades challenging. The complexity increases due to the interconnectivity within the infrastructure and large systems. Alterations to one system can lead to significant implications across others, while the integration and interoperability between legacy and modern systems should also be considered.

When it comes to the protection of these systems, they are mostly equipped with standard cybersecurity measures like firewalls, which serve as protective barriers between trusted and untrusted networks, along with Intrusion Detection and Prevention Systems (IDPS) and Security Information and Event Management (SIEM) systems for real-time security analysis. While these tools are generally reliable, they fall short of providing deep insights into system operations or detecting sophisticated vulnerabilities. The current security techniques and methodologies are not able to fully protect systems and can be bypassed by newer attack methods. This leaves systems exposed to a range of threats, from traditional malware like Stuxnet to advanced techniques like return-oriented programming, data-oriented programming attacks, and side-channel attacks that exploit information leakage [9].

1.2 Research problem

Establishing trustworthiness and trust relationships is critical, especially in complex environments such as industrial SoS, railways, automotive and aerospace systems, where there is an increasing demand on real-time constraints and efficient use of shared resources due to safety critical scenarios. Moreover, autonomous systems have been increasingly penetrating our daily lives in several cases such as unmanned vehicles, manufacturing, and robotics. These systems are capable of operating without human intervention, while their most important characteristic is their ability to learn throughout their entire lifetime using machine learning techniques and adapt themselves to new circumstances, such as changes in the environment and network.

The core question then becomes: how can one system (or a component within a system) establish trust in another? Typically, trust revolves around assurance and confidence that people and devices will behave in expected ways. Yet, ensuring trust in the context of an "artificial community" of systems and devices presents a greater challenge, given the fact that devices lack an inherent judgmental ability to assess risks and other factors in the way humans do. Therefore, there is a need to quantify the concept of trust in a manner understandable by systems. This will enable the systems to rely on each other's data based on evidence-based trust relationships. However, this leads to another critical research question: How can we effectively model and quantify the trustworthiness of an entity? The failure to accurately measure trust can expose systems to vulnerabilities, making them susceptible to malicious attacks. Additionally, trust is a variable factor, as it can change over time based on various factors. Therefore, another question arises: how can we enable systems to make decisions based on real-time evidence regarding the reliability of other systems, considering their actual characteristics and runtime status of the system?

A significant challenge arises from the diversity of systems and varying hardware characteristics, such as network interfaces, computational capabilities, and supported operating systems. This diversity complicates the monitoring process. Identifying all system interdependencies, including both external and internal ones, is particularly challenging. This is especially true for the complex interplay between IT and OT, where boundaries are frequently blurred. On top of that, the SoS often operate in environments that mix old and new technologies, using a variety of protocols (cryptographic and communication) integrated into their communication, remote control centres, and infrastructure over many years.

In industrial environments, trust relationships are necessary to ensure security as SoS are inherently distributed and collaborative, relying on the accurate delivery of services from other system entities. For instance, in the previous example, an automatic train protection system should rely on the data and services coming from other on-board devices to take correct and informed decisions and to actuate the proper controls (e.g., the braking system). This interdependency forms trust relationships where one entity (the trustor) depends on another (the trustee) to provide essential services or information. Under the zero-trust paradigm, this trust should not be taken for granted but should instead be grounded in concrete evidence and analysis of trustworthiness. Incorporating systems that can model and evaluate such trustworthiness is crucial for creating reliable and resilient systems. These systems should be able to make informed decisions about trust, assessing the credibility of other entities continuously and dynamically. Failure to accurately identify and manage these trust assumptions can leave systems vulnerable to failures or targeted attacks.

The goal is to develop trust models that enable the assessment of trust relationships, particularly in the execution of security and safety critical functions. Such models should be capable of determining the level of trustworthiness required for each device and service within the network. This requires trust assessment models that can pinpoint the necessary trust properties and sources of information to quantify trust for each specific device and function. In essence, this approach shifts from a static, assumption-based trust model to a more evidence-based one, as trust is not a fixed attribute but a dynamic one, changing with context, time, and the evolving threat landscape. This approach can enhance the security and resilience of SoS by enabling them to adapt to new threats and vulnerabilities, maintaining operational integrity even under attack circumstances. The aim of this study is to create a model of quantifying trust while identifying indicative properties and sources of trust that can be used for this quantification.

1.3 Structure of the thesis

The thesis is structured as follows:

- Section 2 discusses a review of trust modelling fundamentals, examining current state-of-the-art practices in the domain. This section provides a state-of-the-art analysis of trust evaluation models, emphasizing methods for quantifying trust and decision-making processes.
- Section 3 presents the proposed methodology of trust modelling, including the identification of essential trust properties necessary for assessing system's trustworthiness. This section also describes several approaches for trust provisioning as well as strategies for trust management and outlines various trust management architectures. The chapter concludes by discussing challenges and future research directions in the field.
- Section 4 focuses on the examination of two specific scenarios in which the trust model can be effectively applied within the context of railway networks. This section also evaluates the model's limitations and potential areas for future research and development.
- Section 5 presents the conclusions of the study, including the key findings and insights from the performed research work.

2

Overview of state-of-the-art practices of trust modelling

This chapter elaborates on the fundamentals of trust modelling, along with an analysis of the current state-of-the-art practices in the field. It delves into a detailed examination of existing trust evaluation models, focusing on trust quantification techniques and decision logics.

2.1 Fundamentals of Trust Modelling

The notion of trust is abstract and multi-dimensional, characterized by its dependency on the involved participants, specific scenarios, and a combination of both quantifiable and non-quantifiable factors that may be influenced by subjective opinions. This diversity in trust's nature poses a challenge in formulating a widely used (and acceptable) model. Broadly speaking, trust can be conceptualized as a qualitative or quantitative score that a trustor (the entity placing trust) assigns to a trustee (the entity being trusted) for executing a specific task within a defined context and timeframe. Trust is a critical factor that influences an entity's confidence to depend on another's services. This concept could be found in daily life, where trust is derived from personal experiences, such as the reputation of a specific person or even a product can be influenced by external feedback (reviews, social media, etc.), or from other trusted people (e.g., family, friends).

While trust's significance is paramount in both physical and cyber worlds, establishing it in cyber environments poses unique challenges due to the absence of physical interactions and tangible experiences. For instance, in the physical domain, one might build trust with a company by visiting its premises and observing security processes used and protocols (e.g., for treating personal data). However, such direct experiences are lacking in the digital world, complicating the trust-building process. Moreover, quantifying trustworthiness in digital settings is particularly challenging due to varying observations of what constitutes "trustworthiness" among different entities. Therefore, various users may perceive the same service differently based on their individual experiences. This subjectivity underscores the complexity of developing a standardized and objective measure of quantifying trust.

These considerations highlight the necessity for alternative trust models that can effectively address and integrate diverse perspectives and criteria. Such models must account for the dynamic and

complex nature of trust, including both empirical evidence (like historical data or previous behaviour) and subjective elements (such as reputation and experience from other entities). In the SoS context, where multiple interdependent systems interact, these challenges become even more demanding as several interactions and dependencies should be considered to assign a trust level to a system or a device.

The following subchapters provide a summary of key terms and notions relevant to the domain. Trust is a broad and extensively studied concept, leading to a plethora of definitions in the literature, some of which even present conflicting notions. Given this diversity and complexity, the definition is provided by the NIST glossary [10] and ITU standards, specifically document Y.307 [11], known for their reliability and broad acceptance in the field.

2.1.1 Trust

Trust is the confidence or belief that one element (the trustor) places in another (the trustee), grounded in the trustor's expectation that the trustee will act reliably and with integrity in fulfilling a specific task. This concept builds the following relationship: the trustor who places trust, the trustee who is the recipient of that trust, and the specific task or responsibility. The nature of trust is highly subjective and contextual, shaped by the trustor's unique perceptions and past experiences. Based on this, trust is not a static property but a dynamic one which evolves through interactions, and it is continuously formed by the newly collected information and experiences. This reflects the relational nature of trust, which may include several other elements and actors, making it even more complex (see Chapter 3.4 about trust management networks).

2.1.2 Trustworthiness

Trustworthiness is a set of attributes that makes an entity considered reliable and worthy of trust. It represents an assessment of an entity's (the trustee's) capability and reliability in fulfilling specific tasks or obligations in response to the trust placed by another party (the trustor). This concept revolves around the demonstrable alignment of the entity's behaviour with its specified responsibilities and expected behaviour, ensuring that it acts predictably and as designed (i.e., as prescribed by the manufacturers in the case of a device). It is more objectively measurable, as it is derived from the entity's characteristics (e.g., device capabilities) and its ability to execute specific tasks effectively. This introduces confidence in others regarding the entity's ability to deliver on commitments. Unlike trust, which is dynamic and context-dependent, trustworthiness tends to be more stable and consistent based on the inherent and proven characteristics of the entity.

2.1.3 Trust opinions

Trust opinions are logical statements that express specific characteristics or behaviours of an element (i.e., a device). The main purpose of a trust opinion is to assess the trustworthiness of a particular element required for forming trust. Trust opinions are categorized into two distinct types:

- Atomic trust opinions: These are foundational statements whose veracity is directly assessed from one element (device). The assessment of atomic trust opinions is grounded in evidence, which may originate from one or multiple trust properties (see Section 3.2).

- Composite trust opinions: Composite opinions are more complex, being constructed from multiple atomic trust opinions originating from different trust entities (devices within the network). They represent an aggregation of simpler statements, and their trustworthiness is derived from the combined assessment of these underlying atomic opinions. As such, composite opinions offer a more multi-perspective on the properties being assessed, reflecting the connection of interrelated trust elements. These opinions may be originated from several trust elements, creating an even more complex (and accurate) network of trust.

2.1.4 Trust relationships and network

A trust relationship is a directional link between two entities, identified as the trustor and the trustee, with respect to a specific property or set of attributes. More specifically:

- The trustor is the entity that extends trust. It is the assessing entity, evaluating and placing trust in another entity. The trustor is an entity capable of making trust assessments, typically characterized by its decision-making and judgment capabilities.
- The trustee is the recipient of trust. The trustee is evaluated based on its trustworthiness in relation to the specific attributes or tasks for which trust is being extended.

This conceptualization indicates that trust relationships are always contextual and related to particular properties or attributes of each element. The trustor assesses the trustee based on specific criteria, which could include reliability, capability, integrity, or other relevant characteristics in a given scenario. The trust relationship is thus not just a belief or sentiment but a directed, evaluative link between an assessing entity and the entity being assessed. Therefore, there are several types of trust relationships that create trust networks, such as direct and referral.

A trust network is a complex graph where various trust entities are interconnected through trust relationships. In this network, entities have the ability to change roles dynamically. For instance, an entity serving as a trustor in one interaction might act as a trustee in another. For example, in the following Figure 1, the node A plays the role of the trustor in the relationship $A \rightarrow C$, while it plays the role of the trustee in the relationship $B \rightarrow A$. This dynamic role-switching provides a multi-dimensional perspective on the network's trust dynamics. As can be seen in this figure as well, trust can be established either through direct observation of the trustee by the trustor (for example, A directly observes C) or via an indirect, *referral-based relationship* where the trustor has a connection with an intermediary node that directly observes the trustee. To clarify, consider B as the trustor and D as the trustee. B lacks direct observation of D; however, B is connected to C, who has a direct relationship with D, creating a referral trust pathway.

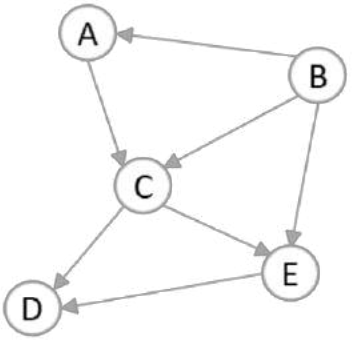

Figure 1: Indicative trust network

2.1.5 Trust modelling

As illustrated in the previous figure (Figure 1), the trust model employs a graph-based framework, including all necessary components and data for executing a specific function. For example, in a railway system's safety-critical function of the previous example, let the node B to be the automatic train protection system that might interact with a rail-side unit (node *A*) responsible for transmitting information. Node *C*, the train's internal gateway, collects data from onboard sensors *D* and *E*, acting as an intermediary between the automatic train protection system and these sensors. It's important to mention that node *B* has the potential to connect directly with node *E*, bypassing the gateway. In this context, the trust opinions eventually pertain to the trustworthiness of these sensors (*E* and *D*).

In this model, components are responsible for creating, transmitting, processing, relaying, and receiving data, which then serves as input for various functions. Within this trust model, vertices symbolize entities known as trust objects, and edges represent the trust relationships between pairs of trust objects. The model also integrates a list of trust properties responsible for forming and quantifying these relationships. A characteristic feature of this model is the directional nature of trust, which is always associated with specific properties. As a result, multiple trust relationships may coexist between the same pair of trust objects, varying based on several properties or the context of the relationship.

Moreover, trust opinions are linked to specific data or node trustworthy status and properties. Thus, we differentiate two types of direct trust relationships: data-centric and device-centric. In a data-centric trust relationship, the trustor's opinion is about a particular data piece, whereas in a device-centric trust relationship, it concerns the attributes of a specific device. A thorough analysis of the data and device properties that are considered in the trust model is presented in Section 3.2.

More specifically:

- Functional trust, commonly referred to as direct trust, is characterized by a direct and unmediated trust relationship, where the trustor possesses firsthand knowledge or experience of the trustee, leading to a trust that is grounded in direct observations and interactions. For example, *A* directly observes *C* in Figure 1.

- Referral trust is determined in a referral trust context, which features an indirect trust relationship. In this scenario, the trustor lacks a direct link with the trustee and instead depends on intermediary nodes that have direct knowledge of the trustee. For instance, in Figure 1, node *B cannot* observe directly node *D*, but only through node *C*. Trust, in this case, is established based on the trustor's confidence in the reliability and accuracy of these intermediaries judgments and observations. This is very common in the case where gateways are responsible for connecting to underlying devices in the network.
- Data-centric trust is evaluated in a context where the focus is on data. Here, the trustor evaluates the trustworthiness of exchanged data and acts as the trustee. This type of trust emphasizes the trustor's dependence on the data's integrity and reliability.
- Device-centric trust, on the other hand, is evaluated based on inherited device characteristics that cannot be changed and affected by other factors.

2.1.6 Quantification

In essence, trust is the subjective belief or confidence that a trustor (the trusting party) places in a trustee (the trusted party), while trustworthiness represents the objective attributes of the trustee that declares this belief. Trustworthiness can be broadly viewed as an evaluation of the trustee's capability to fulfil a specific task within a trust relationship. It is essentially the probability of the trustee meeting the trustor's expectations in a given scenario. These expectations may pertain to various aspects, such as the accuracy of data. Additionally, expectations might involve the trustee's behaviour, particularly in scenarios where the trustee is a node, focusing on the functionality of that node.

To effectively link trustworthiness with the anticipated behaviour, especially when it comes to data sources, it is crucial to recognize that trustworthiness in data doesn't inherently guarantee consistent behaviour. Different trustors may have varying criteria for interpreting trustworthiness into expected behaviour. This bridging can be achieved through decision-making strategies, like implementing rules or policies that infer expected behaviours based on data collected from various sources. For instance, a specific device in the previous example may be believed trustworthy if it meets certain security expectations, like securing that the data coming from the on-board train sensors are reliable and trustworthy.

2.2 Current state-of-the-art

This section is focused on the analysis of the current state of the art in the trust modelling domain. There are several approaches and frameworks that have been proposed to calculate trust, mainly in the IoT domain, which has characteristics similar to the SoS concept. Most of them follow diverse architectures due to the multidimensional nature of trust. It is evident that trust is not a one-size-fits-all concept since it varies greatly depending on the entities involved and their dependencies.

As in the case of SoS, in the dynamic and heterogeneous environment of IoT, understanding the true intentions of devices is crucial to ensure the security posture of the overall network. Køien [12] research focuses on analysing and identifying various aspects of trust as they relate to software, hardware, devices, and services. The focus is on human trust for a variety of IoT elements, including the trustworthiness of devices. This aspect is particularly important as it addresses a core question

of whether humans and devices can rely on others that constitute an integral part of the IoT ecosystem. The author proposed a *subjective logic system* for modelling trust interactions, considering properties like transitivity, integrity, and benevolence. Kurdi et al. [13] introduce a lightweight trust management algorithm also based on *Subjective Logic*. The paper addresses the challenge of establishing trust among multiple service providers by enhancing security and reliability in complex cloud computing environments. It is a common practice that cloud service providers collaborate and exchange resources and services amongst themselves in the event of resource shortage. Therefore, during such shortages, a service may be transferred to another provider, which is very efficient, but it raises several security concerns: if the provider is unreliable, this will cause substantial response delays or even non-responsiveness, which can directly impact the quality of service. To mitigate this risk, the authors proposed a trust management framework to evaluate the reliability of the providers, where the decisions regarding the redirection of requests are taken based on this assessment. This process involves the generation of subjective trust opinions, which are derived from a system rooted in both reputation metrics and service level agreements. Similar to this, Selvaraj and Sundararajan, S. [14] proposed a dynamic *evidence-based trust model* to ascertain the trustworthiness of services in cloud environments. The model uses *fuzzy logic* to derive trust values, improving the reliability of services in the cloud by dynamically adjusting trust assessments based on evidence, thereby enhancing the adaptability and accuracy of trust evaluations. Sanjay and Guru [15] discuss formulating trust relationships and scoring the reputation of parties or entities. Their approach employs *subjective trust measurement* to analyse trust between parties and offers insights into evaluating trust on networks, especially relevant in the context of federated clouds, which is crucial for maintaining service level agreements and ensuring reliable inter-cloud interactions. Jalal et al. [16] propose a *subjective logic-based trust model* specifically for fog computing. It aims to detect rogue fog nodes to prevent processing IoT user requests at these nodes, offering a novel application of subjective logic in trust models. It enhances the security and integrity of fog computing networks by identifying and mitigating the risks posed by malicious or compromised fog nodes, thereby protecting IoT user requests. Gao et al. [17] introduce a new trust model to address trust issues arising from the openness, dynamics, anonymity, and uncertainty of systems. It enhances *subjective logic* with a new trust quantification formula, incorporating negative-event and time effects, and offers a fresh perspective on evaluating trust relationships between peers. It offers a more comprehensive approach to evaluating trust relationships in open, dynamic, and uncertain systems, which is crucial for establishing reliable peer-to-peer interactions in various contexts. Fortino et al. [18] defined a “Cloud-of-things” concept by virtualizing physical devices in the cloud and integrating them with software agents for responsibility management. Their algorithm, which considers mutual trust, effectively handles untrusted agents and computational overhead.

In a different study [19], the concept of trust evaluation is applied to continuous authorization in IoT systems. This research adopts the *zero-trust model*. As a result, each action must be initially authorized and then consistently revalidated. The procedure involves assessing the trust level of a device before granting it access to any resource. This assessment incorporates data from various sources, which is then analysed using methods derived from subjective logic. The resulting *trust score* determines if an entity is permitted to access certain resources. Another research in the IoT field [20], particularly in crowd-sensing scenarios, examines how reliable the data provided by

sensor nodes is. In these IoT networks, the reliability of nodes is often uncertain, necessitating the evaluation of the trustworthiness of their data. This data quality is then used to gauge the reliability of conclusions drawn from it, employing subjective logic to express the level of trust. These trust levels are considered in making decisions based on the outputs of the processing node. Ali et al. [21] developed a trust scheme for Wireless Sensor Networks, focusing on data aggregation with a distribution cluster-based routing algorithm to mitigate trust manipulation. Mendoza et al. [22] proposed an IoT trust management framework with mechanisms for direct interactions and indirect observations, like trust table exchanges. The trustworthiness of each node is calculated by evaluating the quality of service it provides and the endorsements it receives from neighbouring nodes. This trust evaluation is conducted locally by each node for its neighbours, eliminating the need for a centralized authority. Pal et al. [23] focused on access control in IoT, proposing a trust management framework that considers direct, recommended, and derived trust. The novelty of this scheme lies in its direct integration with attribute-based identity systems. This allows for the distinct and concrete identification of entities, enabling their evaluation using specific trust assessment criteria. Bernabe et al. [24] introduced a trust-aware access control mechanism using fuzzy logic based on historical trust evidence and, more specifically, quality of service, reputation, security aspects and social relationships. Bica et al. [25] proposed a multi-layer security framework for trust evaluation in IoT based on reputation scores, while Ruan et al. [26] developed a trust management framework based on measurement theory, considering interactions between various IoT entities and employing reputation to calculate trust levels. Sharma et al. [27] presented a generic trust management framework for IoT that considers both qualitative and quantitative parameters. Bahutair et al. [28] developed an adaptive trust model for IoT services, using an algorithm to quantify trust factors through four stages of per-indicator assessment and trustworthiness prediction. Various attributes have been considered as trust factors, for instance, in computing services, attributes like processing power and memory capacity, while for sensors, crucial factors include their mobility, efficiency, and the speed of communication.

Moreover, Battah et al. [29] proposed a general trust framework for IoT using reputation systems and blockchain technology, implementing smart contracts for decentralized trust scoring storage. This enables devices within the network to access ledger-stored data regarding the trustworthiness ratings of their neighbouring devices. Feraris et al. [30] moved forward with a framework to support designers and developers in considering trust in IoT, focusing on privacy and identity requirements, such as anonymity, pseudonymity, unlikability, confidentiality, and others. Dwarakanath et al. [31] proposed a trust-based approach for distributed event processing in IoT, leveraging trust based on past interactions and user recommendations. The innovation lies in evaluating the system's resilience against adversarial impacts, considering factors such as historical exchanges of recommendation messages and employing a similarity-based method to ascertain the reliability of received recommendations. Hussain et al. [32] proposed a context-aware trust evaluation model for IoT, emphasizing the importance of trust and reputation in all types of IoT interactions. A context determines the specific role or function of a device within a network. This means that if the node attempts to access any type of information beyond its designated purpose, it is considered to have violated its exclusive function, and consequently, its reputation score. Ursino et al. [33] explored trust and reputation in multiple IoT scenarios, suggesting that the diversity of IoT entities requires different trust models. Lastly, Bao and Chen [34] focus on addressing the security and reliability

challenges posed by the integration of numerous devices in heterogeneous network environments. Their study underscores the importance of trust management in IoT, particularly in the context of service composition and in ensuring the security and reliability of interconnected devices in diverse network environments. The authors propose a scalable trust management protocol for IoT, emphasizing the role of social relationships. The paper demonstrates that trust-based service composition in IoT environments significantly outperforms non-trust-based (random) service composition. Overall, the paper emphasizes the importance of trust management in IoT, particularly in the context of service composition and in ensuring the security and reliability of interconnected devices in diverse network environments.

Summarizing the previous state-of-the-art analysis, the following table presents the key contributions in each field:

Domain	Author	Key Contributions
Trust management in cloud computing	Kurdi et al. [13]	Lightweight trust management algorithm based on Subjective Logic
	Selvaraj and Sundararajan, S. [14]	Dynamic evidence-based trust model using fuzzy logic
	Sanjay and Guru [15]	Subjective trust measurement for evaluating trust
	Jalal et al. [16]	Subjective logic-based model to detect rogue fog nodes
	Gao et al. [17]	Enhances subjective logic with new trust quantification formula
	Fortino et al. [18]	Virtualizing physical devices in the cloud
Trust management in IoT domain	Køien [12]	Subjective logic system for modelling trust
	Mendoza et al. [22]	Trust management with mechanisms for direct and indirect observations
	Bica et al. [25]	Reputation scores-based security framework for trust evaluation in IoT
	Ruan et al. [26]	Trust management using reputation scores
	Sharma et al. [27]	Considers both qualitative and quantitative parameters for trust management framework in IoT
	Bahutair et al. [28]	Adaptive trust model, quantifies trust factors through per-indicator assessment
	Battah et al. [29]	Reputation systems and blockchain for decentralized trust scoring
	Feraris et al. [30]	Trust in IoT focusing on privacy
	Dwarakanath et al. [31]	Trust based on POL
	Hussain et al. [34]	Context-aware trust model for IoT, evaluates trust and reputation in various contexts

	Ursino et al. [33]	Different trust and reputation models for diverse IoT entities
	Bao and Chen [34]	Scalable protocol emphasizing social relationships
Trust management for Wireless Sensor Networks	Sohail et al. [20]	Reliability of data in crowd-sensing. Evaluates trustworthiness of sensor node data
	Ali et al. [21]	Data aggregation with distribution cluster-based routing
Trust Management for Access control in IoT	Bernabe et al. [24]	Trust-aware mechanism based on historical trust evidence using fuzzy logic
	Pal et al. [23]	Trust management framework integrated with identity systems
	Dimitrakos et al. [19]	Zero-trust model for continuous authorization

This analysis clearly indicates the need to establish a concrete methodology that is able to combine trust properties collected from several sources. This methodology should consider various aspects, starting from the evaluation model to the quantification and the overall management and enforcement of trust within the network. Hence, the following subchapters present a review of the existing models in each of the dimensions that need to be considered.

Trust evaluation models	Policy-based Trust Evaluation Models (see §2.3.1)
	Reputation-based trust models (see §2.3.2)
	Knowledge-based trust models (see §2.3.3)
	Other models (see §2.3.4)
Trust quantification techniques and decision-logics	Probabilistic logic (see §2.4.1)
	Bayesian probabilistic logic (see §2.4.2)
	Fuzzy logic (see §2.4.3)
	Belief theory and subjective logic (see §2.4.5)
	Machine-learning based models (see §2.4.5)

2.3 Trust evaluation models

Trust evaluation models are designed to assess the reliability and trustworthiness of different entities within a network. They employ a range of methods to measure trust, including past behaviour, evaluating reputation through feedback or ratings, and assessing compliance with established policies or rules. Trust evaluation models can be dynamic, adapting their assessments based on ongoing interactions and behaviours, or static, relying on predefined criteria.

2.3.1 Policy-based trust models

Policy-based trust evaluation models provide a structured approach for assessing trust in various contexts, where the trustworthiness of a device is assessed against predefined policies or rules. These policies may include various criteria, from compliance with security standards and protocols to adherence to specific behavioural profiles. An important feature that makes these models suitable in many different settings is their dynamic nature: the trust level associated with an entity can be adjusted in real-time, reflecting ongoing compliance (or violation) of policies (i.e., based on the criticality of the service that the device should execute the policy can adapt the required level of trust). Also, the policies are context-sensitive, allowing for significant variation depending on the specific environment in which the devices are deployed and operated. For example, the level of trust can be readjusted if the device is placed in a more critical environment and its malfunction may cause significant harm.

Refsdal [35] proposes a UML-based method for modelling and analysing trust, identifying risks and opportunities, and developing trust policies to optimize decisions in dynamic, open, and distributed systems. To formulate these policies, security administrators can use several tools. Drools [36] is one such tool, which is an open-source business rule management system with a forward-chaining, inference-based rules engine. While Drools is not exclusively a policy-based trust evaluation model, it can effectively implement such models. Its core functionality lies in defining, managing, and processing complex business rules and decision logic, which makes it well-suited for scenarios where dynamic rules or policies govern decision-making. For example, an organization could use Drools to automate the process of evaluating device compliance with its trust policies, such as verifying if systems use a specific version of an operating system or are older than a certain benchmark. Therefore, trust evaluation using such models could be automated, enhancing decision-making efficiency and making them particularly valuable in industrial and IoT environments, where strict adherence to security policies is timely and crucial, and human intervention is minimal.

2.3.2 Reputation-based trust models

Reputation-based models evaluate trust based on a combination of feedback, experiences, and ratings provided by other entities within a network. These models are commonly used in IoT scenarios and peer-to-peer networks. They aggregate multiple sources of trust information gathered by several entities, reflecting the network's collective assessment of an entity's trustworthiness. A significant advantage of reputation-based trust models is their reliability and resistance to false feedback, as the trust opinion is collected from diverse sources over time. This approach ensures that even if a malicious device behaves normally towards a specific device or only during certain periods, its overall historical behaviour can be detected by collecting its overall footprint.

However, there are several challenges to consider, such as handling the subjectivity of opinions and the possibility of bias. Despite these challenges, reputation-based trust models offer numerous advantages, mainly due to their dynamic nature. An entity’s reputation is not static but evolves with each new interaction or piece of feedback, meaning that trust assessments are continuously updated to provide a current view of an entity’s trustworthiness. Additionally, these models rely on the collective input and judgment of a large number of devices, making them robust against centralized failures and capable of handling large-scale interactions. Another critical aspect is context-

dependence, as the factors influencing reputation can vary greatly across different platforms or environments. For example, the criteria for assessing a device's reputation in an industrial setting may differ significantly from those used in a smart city environment.

Tajeddine et al. [37] proposed reputation-based trust model for distributed systems effectively incorporates reputation values, direct experiences, trust in host credibility, information decay, and cooperation among host systems. Gong et al. [38] combine reputation and credential, significantly improving the accuracy of trust value computation in situations with little transaction history between peers.

2.3.3 Knowledge-based trust models

Knowledge-based trust models focus on acquiring and analysing knowledge and evidence about an entity's properties (or behaviour) to determine its trustworthiness. Unlike reputation-based models, which rely heavily on subjective feedback or opinions from other peers in the network, knowledge-based models are objective since they perform evidence-driven assessments. More specifically, trust evaluations are grounded in concrete properties, evidence or historical data regarding the entity's past actions, performance, or inherent characteristics (e.g., length of keys used in communication). Therefore, these models are particularly effective in environments where direct verification and factual proof of trustworthiness are paramount, such as in security-sensitive environments. This reliance on direct experience and observation not only provides these models with a degree of control but also protects them from the external biases that often influence reputation-based systems, as described above. The objective nature of knowledge-based trust models is one of their core strengths, allowing trust to be established on verifiable and measurable facts rather than perceptions and beliefs. As in the case of the previous models, knowledge-based trust models are also dynamic, as the trust levels can be continuously updated as new information or evidence is received. This ensures that trust assessments remain relevant and accurate over time. They are also context-specific, tailoring trust evaluations to the particular context based on the available evidence.

Often employed in automated systems, these models make trust decisions based on predefined criteria or measurable attributes, making them a critical component in systems requiring automated, reliable trust assessment. Knowledge-based trust models provide a robust, evidence-driven framework for establishing and managing trust in various applications, from secure communications and system reliability assessments to data integrity verification. Soto et al. [39] proposed a multi-agent architecture and trust model to foster knowledge reuse in communities of practices, with artificial agents using the model to recommend the most trustworthy knowledge to members. Marshall et al. [40] introduced a dynamic model of trust development and knowledge sharing that explains how trust relationships are formed, developed, and dissolved based on levels of trust and knowledge sharing between partners.

2.3.4 Other trust models

Beyond the above-mentioned trust evaluation models, several other approaches have been proposed, each tailored to specific contexts and requirements. Behaviour-based trust models, for instance, focus on assessing trust through an entity's observable actions and interactions with other entities over time, making them ideal in settings where past behaviour is indicative of future actions. In

contrast, certification-based trust models derive trust from certifications or accreditations issued by authoritative bodies during the design of the device, as in the case of medical devices. Role-based trust models assign trust levels according to the roles or positions entities hold within a system or organization. This approach is common in hierarchical structures where certain roles are inherently trusted more due to their responsibilities. On the other hand, context-based trust models consider the specific situation or environment to evaluate trust, recognizing that trust levels may change depending on the context. Each of these models offers a different perspective through which trust can be evaluated and managed while they can be efficiently combined to serve different needs.

2.4 Trust quantification techniques and decision-logics

Trust quantification techniques and decision logics transform qualitative trust assessments into quantitative metrics using several methods, such as statistical analysis, probabilistic and fuzzy logic. Ries et al. [41] presented a new model for evaluating propositional logic terms under uncertainty, which aligns with standard probabilistic approaches and subjective logic. Their paper introduces a novel approach to assessing the trustworthiness of complex systems, emphasizing the integration of probabilistic logic with traditional trust evaluation methods. Kumar [42] focuses on advancing probabilistic logic programming by combining probability theory with logic programming. The study introduces a context-specific likelihood weighting and the extension to support discrete and continuous random variables is a significant advancement, especially for applications dealing with numeric data. The paper of Aldini et al. [43] demonstrates how classical modal languages can be used to define a logical framework for building and analysing context-aware trust relations based on evidence. It offers a novel approach to understanding and managing trust in various contexts, emphasizing the importance of evidence in forming trust relationships. Nafi et al. [44] extend the Certain Trust Model, focusing on its representational aspect using both probabilistic and fuzzy logic. The proposed model is applicable in areas like cloud computing, aiming to ensure the trustworthiness of these platforms, thus addressing the critical need for trust in service-oriented systems. Yuan et al. [45] introduce a framework that integrates Deep Neural Network Sequential Recommendation models with logical reasoning called Sequential Recommendation with Probabilistic Logical Reasoning. Their work enhances recommendation systems by applying probabilistic logic, addressing the challenges of combining deep learning with logical reasoning in a sequential recommendation context. The following subchapters provides an analysis of the most commonly used decision logics in the domain.

2.4.1 Probabilistic Logic

Probabilistic logic [46], merging probability theory and logical reasoning, is a branch of logic designed to handle the uncertainties found in logical expressions. This logic extends beyond the binary boundaries of traditional logic, where statements are classified as true or false (1 or 0), to accommodate better real-world scenarios where such absolute certainty is almost impossible. By assigning probabilities to logical statements, probabilistic logic quantifies belief or confidence levels required in the form of trust opinion, offering a more concrete approach to understanding and reasoning. To this end, central to probabilistic logic is its ability to handle ambiguity and (possible) incomplete information. It does so by blending the probabilistic mathematical framework with the

structure of formal logic, thereby enabling logical propositions to be evaluated in terms of probabilities rather than absolute truths. This integration is particularly useful in the field of trust modelling. Fierens et al. [47] present efficient algorithms for inference and learning in probabilistic logic programs, addressing classical tasks like marginal computation and learning from interpretations in the machine learning domain.

2.4.2 Bayesian Logic

Another dimension of probabilistic logic is the use of Bayesian methods [48]. Central to Bayesian probability is Bayes theorem, which allows to continually update the probability of a hypothesis as more evidence becomes available. It integrates prior knowledge or experience (prior probability) with new, incoming data to revise and update beliefs (posterior probability). This aspect underscores the dynamic nature of reasoning in probabilistic logic, adapting and evolving with additional data, which makes it suitable for the trust modelling domain as the perception of trust is not a static property. Jin et al. [49] propose an uncertain trust model that combines cloud and Bayesian networks for context-aware trust evaluation and dynamic updates in online services. The interplay between Bayesian probability and trust modelling is particularly important in reputation-based trust models (see 2.3.2), where Bayesian methods are useful in dynamically updating trust assessments based on new evidence received from other peers. This gathered information triggers the recalculation of the trust level assigned to an entity. This process ensures trust assessments remain dynamic and reflect the most current incoming trust data.

Moreover, Bayesian probability facilitates the incorporation of prior knowledge into trust evaluations. This feature is especially useful when dealing with newly encountered devices with limited behavioural and historical data. This is a critical issue that characterises highly dynamic networks and is related to onboarding new devices in the domains (called “cold-start issue”). In such scenarios that are characterized by uncertainty, Bayesian approaches can effectively manage these aspects and quantify the initial value of trust based on a probabilistic measure. Another key aspect of Bayesian probability in trust modelling is its capacity for eliminating biases and subjectivity in trust opinions. Trust assessments can be tailored based on individual user preferences and past experiences, acknowledging that different users may interpret and weigh evidence differently. This personalization leads to trust scores that are more closely aligned with individual perspectives and histories.

2.4.3 Fuzzy Logic

Fuzzy logic is a form of many-valued logic that deals with approximate rather than fixed and exact reasoning. This approach is particularly useful in dealing with problems that are too complex for a binary logic due to inherent ambiguities and uncertainties. Unlike conventional logic, where variables are true or false, fuzzy logic works on a range of truth values ranging between 0 and 1. For instance, consider the temperature of a room. Instead of defining a specific cutoff point where a room becomes “hot”, fuzzy logic can represent the temperature as partly belonging to multiple categories. A room at 22°C might be considered 0.4 warm and 0.6 hot, acknowledging that the perception of temperature can be subjective and varies between individuals. This allows for a more

relaxed understanding of properties, making it well-suited for trust modelling, where trust opinions may be vague and other imprecise information may be considered to quantify trustworthiness.

Trust modelling, inherently subjective and context-dependent, benefits from fuzzy logic's ability to include degrees of trust rather than a fixed binary classification. It allows trust models to reflect the dynamic nature of trust, accommodating the variability in perceptions and expectations among different systems. Fuzzy logic has been widely applied in trust modelling across various domains, providing a framework for handling the uncertainties and ambiguities inherent in trust assessments. Prabha and Latha [50] presented a fuzzy logic-based multi-attribute trust model that effectively detects malicious nodes in wireless sensor networks, outperforming weighted summation-based multi-attribute trust evaluation. Also, Liao et al. [51] introduced a fuzzy logic-based trust model in grid environments that improves security by enabling direct trust, derivation and combination of trust, and removing malicious recommendations. Another application in smart grid networks involves using a fuzzy logic-based trust model to identify untrustworthy nodes [52].

2.4.4 Belief theory and subjective logic

Belief theory, also known as evidence theory or Dempster-Shafer theory [53], is a robust methodology for reasoning under uncertainty. It is linked to several other theoretical approaches, including probability, possibility, and imprecise probability theories. This theory stands out for its ability to incorporate and manage varying degrees of knowledge and uncertainty, distinguishing it from classical probability. In the context of trust computational models, belief theory is important in evaluating the trustworthiness of devices within dynamic IoT networks. It does this by quantifying trust in terms of belief, disbelief, and uncertainty regarding one element's reliability towards other elements. This approach allows for a more flexible assessment of trust, accommodating the complex and often dynamic nature of trust relationships. By leveraging belief theory, the trust evaluation models can effectively address the challenges of decision-making in environments where information is incomplete or ambiguous, ensuring more reliable and informed trust evaluations.

Subjective logic [54] is another branch of probabilistic logic in combination with belief theory, which is specifically tailored to handle subjective beliefs influenced by uncertainty and a lack of complete evidence. This form of logic extends beyond the fixed models of conventional probabilistic methods by incorporating subjective probabilities to represent beliefs. Again, this approach is especially pertinent in scenarios where information is incomplete or the reliability of trust opinion and of the gathered information is in question.

In the trust modelling domain, the role of subjective logic is very powerful in assessing and quantifying trust. One of the most important features of subjective logic is that not only considers the incoming evidence but also evaluates the degrees of belief, disbelief, and the inherent uncertainty associated with each entity being assessed. This is crucial in trust modelling, where trust is not a static metric but a dynamic one. Hence, subjective logic can facilitate the combination and updating of beliefs, adapting trust assessments as new evidence becomes available. Lifen [55] presents a trust model based on subjective logic and collaborative filtering, which effectively derivates and transitively trusts in e-commerce and online environments. Haydar [56] proposed global trust model based on subjective logic outperforms the existing local trust model in a question-answering social

network, due to its precise interpretation of trust context and ability to satisfy new users. Moreover, subjective logic is useful in decision-making processes within trust-based systems, particularly when navigating through incomplete information. It enables the evaluation of the reliability of information sources, ensuring that decisions are made on a more informed and sound basis.

2.4.5 Machine learning-based models

Machine learning has penetrated every aspect of our daily lives, including the trust modelling domain. It provides advanced models for assessing and predicting trust in various contexts. These models primarily focus on predicting factors that influence interactions and decisions, as well as identifying patterns of normal behaviour in a device or a cluster of devices. Identifying deviations in such patterns is a strong indication of abnormality.

Recent research in this domain includes studies like the one conducted by Jayasinghe et al. [57] They developed an intelligent trust computation model capable of generating accurate and intuitive trust values for potential actors. Their model quantifies individual trust attributes numerically and introduces a novel algorithm, rooted in machine learning principles, to classify and integrate these trust features. This results in a comprehensive trust value that enhances decision-making, particularly in the context of IoT services and cyber-physical systems. Similarly, Ma et al. [58] introduced a machine learning-empowered trust evaluation method for IoT devices. This approach aggregates trust properties, such as network quality of service, and quantifies trust as continuous numerical values. This method effectively indicates the trust status of a device, thereby facilitating more informed decision-making.

2.5 Potential research directions

As analysed in this chapter, several research studies and approaches in the field focus on deriving trust, primarily concentrating on trust calculation within the IoT and cloud domains. These studies mostly explore how observations from other network entities contribute to trust calculation using a reputation-based approach. Most of the contributions in trust management focus on innovative methods such as the Subjective Logic and the use of reputation scores, and blockchain for decentralized trust management and recording. However, there is an evident need for designing a holistic approach and methodology capable of capturing various trust properties, including those inherent to the device and opinions from other devices in the same network. Current methods often overlook the actual security status of devices and their direct assessment, relying mostly on observations from others. Therefore, it is essential to study the types of trust properties that can be captured and how this information can be retrieved from the device in a trustworthy way. Also, it is necessary to consolidate opinions from multiple sources and quantify the trust score in a consolidated yet device-understandable manner. This will enable devices to rely on each other's data based on evidence-based trust relationships without any arbitrary assumptions about their state. The question remains: How can this information be modelled based on specific criteria, and how can the relationship between devices affect this trust scoring? Another important consideration is the prioritization of devices and processes based on their role in the overall setting, particularly when executing highly critical functions, and how this impacts overall trust quantification. The next chapter will focus on a reference trust modelling methodology that addresses these needs, presenting

a thorough analysis of potentially considerable trust properties and trust provisioning methods to secure such information.

3

Trust modelling methodology and trust properties

This chapter elaborates on the trust modelling methodology along with the identification of the required trust properties. These properties are necessary for quantifying the trustworthiness level of a system. Additionally, the chapter provides insights into trust provisioning methods and management methodologies, concluding with an overview of the trust management architectures.

3.1 Trust modelling methodology

This subchapter outlines the trust modelling methodology with a detailed definition and description of the steps to be followed. Also, it presents a flow-diagram of the derived methodology.

a) **Definition of trust objectives and their context**: The first step of trust modelling involves the definition of the trust objectives within the system, meaning the context in which trust should be applied (i.e., the systems, the network and the topology as well as the specific needs and the criticality of the systems). This includes the definition of the system's properties that contribute to its trustworthy state, depending on its placement, role and criticality within the overall setting (or SoS), such as the reliability or the integrity of specific systems. For instance, in the railway example presented before, the integrity of the data transmitted by the on-board sensors to the braking system is vital for the safety of the train and its passengers, in contrast with other auxiliary and secondary systems that are less critical. At this stage, it is essential to clarify those trust properties as well as other factors that can affect the trust level of the system at this step. Again, in the case of railways, specific legislation and regulatory requirements, such as IEC 62443 [59], which provides a framework to address and mitigate security vulnerabilities in industrial control systems, should be respected throughout the entire life cycle of the systems and during railway operation. Therefore, during this phase, the specific requirements are extracted based on the standards and the legislation forming the specific security and trust requirements. The following subchapter elaborates on the possible trust properties that could be considered for assessing the trust in the domain (see 3.2). However, it should be noted that not all the

properties equally affect the system's trustworthiness as it depends on several factors (e.g., an outdated version of the cryptographic protocol heavily affects the communication between the devices, while other factors may not). Additionally, the system's required levels of trustworthiness depend on the criticality of this specific system and the running service and function within it. For instance, a sensor that controls the breaks in the railway is more vital than a humidity sensor that controls the air conditioning units in the passenger area, indicating that the former holds a higher (more critical) position in the trust hierarchy. Also, the process of transmitting information from this sensor to the automatic driving systems is highly critical, where the integrity of the transmitted information is of paramount importance. Similarly, when a firmware update is transmitted to a critical component, the overall process should be secured both from the communication standpoint (secure and authenticated communication channel) as well as the integrity of the process itself. In summary, the first step includes the overall purpose of the trust model and its connection to the specific system to be assessed as well as the specific aspects to be measured and managed.

b) **Identification of Trust Entities**: The second step is related to the identification of the trust entities involved in a particular system or in the interconnection between systems (in the case of SoS). This includes identifying all critical parts, such as devices, networks, applications, systems, sensors, and network equipment. This step is connected to the threat analysis, which is crucial for examining all deployed components, their characteristics, and running services on them. With this knowledge, identifying potential threats to these services is important, often by comparing system vulnerabilities with existing databases (e.g., CVE). This examination should also consider the specific characteristics and capabilities of each device and system (i.e., processing power and capabilities as well as available network interfaces). For example, a system that exposes wireless network interfaces should be treated in a totally different manner in contrast with another one that is equipped with physical (wired) ports. This means that, in the second case, we assume that the malicious actor is on site and his physical presence is required. Additionally, the processing unit is important and the hardware capabilities since the differentiate the types of the attacks to be considered (e.g., in a more powerful system advanced attacks should be considered that can be exploited in the level of the operating system). This step also involves constructing the Bill of Materials and Asset Inventory, which are critical for industrial systems, providing security administrators with a holistic view of the deployed devices, their applications, and services, enabling effective maintenance.

c) **Trust relationships within a system:** This step is critical for identifying interrelations and dependencies among a system's components and its subsystems. To this end, it is essential to understand how relationships are established between components and how risks and particular threats can propagate (possibly hopping from one device to another). This is important in interconnected infrastructures where threats should be considered not only on the device level but also in the context of devices connected to each other. For example, imagine that device A cannot directly observe device C but only through device B. This means device A must trust device B before assessing the trustworthiness of information received from node C. This also pertains to the threats and risks associated within device B.

If device B is vulnerable, the information it transmits from C to A could be incorrect, even if the original information from C was accurate. This step also involves establishing a hierarchy and categorizing services based on their criticality. In this context, factors such as latency and downtime of specific services should also be considered, especially in highly dynamic environments where the context and criticality of a device or service can change, requiring that the trust model prioritize certain assessments due to their importance in the overall setting.

d) **Trust criteria and properties:** Based on the previous examination of the trust relationship, the specific trust criteria that should be determined. These criteria are the backbone of the trust model, as they define the specific properties to be assessed during the system's runtime and when evaluating the trustworthiness of a component. As mentioned above, the selection of these properties is closely related to the context and the scope of the trust model (Step a), the nature of the component, meaning the specific characteristics (Step b) and, of course, the relationship between the entities (Step c). The three-tier categorization of the trust properties is outlined as follows:
    i. **Knowledge:** It includes firsthand observations of a specific entity, and involves both data-centric aspects, such as those related to the behaviour of the device in the network or the quality of the provided data, and device-centric, which pertain to system integrity (e.g., the running operating system and services). Therefore, this category is connected to the direct trust derivation, and includes aspects related to compliance with standards and policies as well as other official certifications (from recognized authorities and institutions).
    ii. **Experience:** This category is related to the historical behaviour and actions of the entity, including aspects such as previous network behaviour, transaction history, fulfilment of obligations, past interactions, respect of QoS criteria. This experience is a combination of both direct and indirect trust aspects as the observations could be collected by other trust entities or other modules in the system, such as an intrusion detection system. In case of indirect trust, specific attention should be given to the trust relationship between the trustor and the intermediary.
    iii. **Reputation:** This category pertains to the trust inferred from a device's social behaviour and the opinions and recommendations shared from other entities. This allows the more holistic view and quantification of the device's trustworthiness, enabling reliance not just on specific direct assessments (occurred in a specific timing). This is particularly useful in cases where a device exhibits abnormal behaviour during specific timeframes (or only in interactions with certain devices), deviating from its normal, benign behaviour.

e) **Trust metrics**: After identifying the critical properties of a specific element, which is connected to the level of criticality for a particular function, the required trust level can be defined. This is directly linked with the quantification of trust metrics. For instance, a relatively low trust level may be adequate for a device's low-critical functionality. On the contrary, in a highly critical situation, the highest level of trust assurance is necessary. This raises the question of how to quantify the level of trust. Such quantification should consider all aspects as described in the previous step. As mentioned in subchapter 2.4, there are

several metrics available for calculating and quantifying trust properties and the overall trust score (level) within the context of a trust network. Quantitative criteria typically involve calculating a trust score on a numerical scale, usually ranging from 1 to 100, to represent trust levels. Qualitative criteria, on the other hand, assess whether the trust score falls above or below a specified threshold, incorporating various logics. Beyond the sole calculation of trust properties, there are two crucial aspects to consider. The first is the assignment of weights to different trust factors. The second important aspect is the aggregation process. This involves combining individual scores to form an overall trust rating, ensuring a thorough evaluation of trustworthiness. Both weight assignment and score aggregation are essential for a holistic understanding of the trust network.

i. Weights: The process of assigning weights to various trust criteria should be linked to the criticality of the component and its operational service, while also considering the specific context of the device. Additionally, when it comes to indirect trust, such as reputation-based scoring, assigning appropriate weights is crucial. This is because trust opinions collected from others need careful evaluation and should not be accepted as an as-is value.

ii. Aggregation: To derive the overall trust level of a specific element in the trust network, it is essential to combine individual trust metrics and scores. This integrated approach ensures the holistic assessment of trustworthiness. For this purpose, a generalised formula is given below:

$$T = (w_K \times K \times C) + (w_E \times E \times C) + (w_R \times R \times C)$$

- *T is the Trust Score.*
- *w_x are the weights for Knowledge (K), Experience (E), and Reputation (R), respectively.*
- *K, E, and R represent the scores for Knowledge, Experience, and Reputation.*
- *C is the criticality score.*

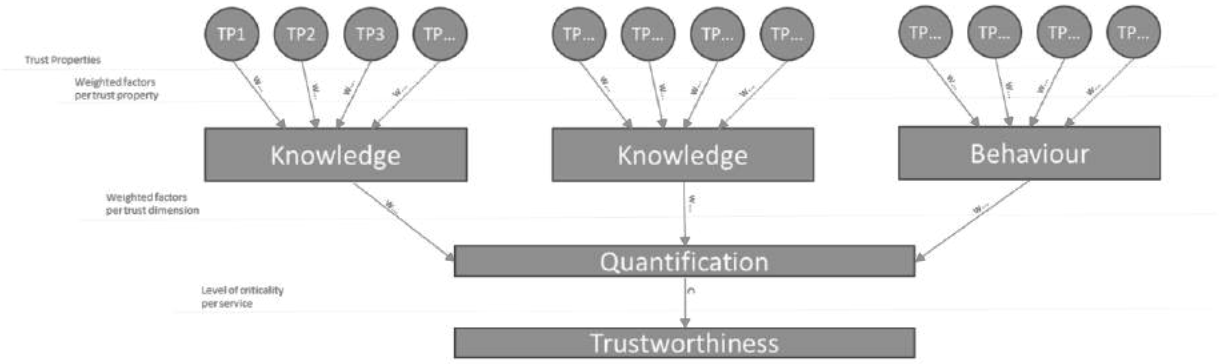

Figure 2: Trust quantification

f) **Trust policies and implementation of the model**: Apart from the quantification of the trust level, formulating trust policies related to trust governance is crucial. This involves how trust is established for each element, maintained, re-evaluated and revoked if required. Also, technical measures should be put in place to support the trust model, such as the trust provisioning (e.g., trust and monitoring agents to collect the necessary information as well as the assessment logic). Especially in highly critical and complex settings, trust-related decisions should be taken in real-time and be capable of capturing all changes and modifications in the trust model. It is also crucial to continually update the trust model with "fresh" information. Therefore, reassessment should be an iterative process throughout the system's entire lifecycle, considering changes in the trust network and the interdependencies among various trust properties. Specific policies are needed to monitor the model's correct execution and the specific actions that need to be taken in case of potential malfunctions, as indicated by a low trust score. This might include updating the system or rolling back to a previous version, taking into account critical security factors like the service's criticality and the maximum expected downtime of the system. Additionally, during execution, other aspects should also be considered, such as the cryptographic methods and communication protocols in place to secure the exchange of information between trust elements (to ensure confidentiality).

g) **Monitor and update the model:** The overall performance of the model should be continuously monitored to ensure its accurate operation and effective assessment of trust. This process involves more than just recalculating trust levels based on changes in the environment and network topology. It also includes adjusting the trust criteria, metrics, quantification formulas, and algorithms themselves. Such adjustments necessitate ongoing testing and validation of the trust model. This ensures that it accurately reflects the trust levels, based on predefined tests, and meets the defined objectives. These objectives can be verified through simulations and actual use case testing.

The diagram below illustrates the entire process and steps of the designed trust model:

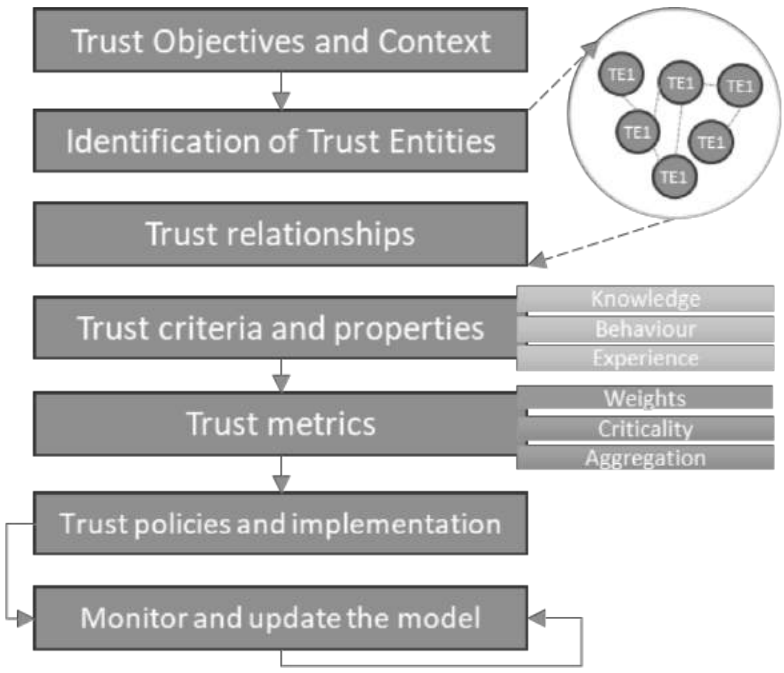

Figure 3: High-level trust model

3.2 Trust properties

This subchapter delves into the specific trust properties that play a crucial role in the trust assessment process. Recall that trust relationships describe the connection between a trustor and a trustee, focusing on specific properties of interest, known as trust properties. These relationships are assessed by measuring various trust factors, determining their assurance levels, and validating if they meet predefined criteria. The evaluation of these criteria should be adaptable, depending on the criticality of the service being executed. In high-criticality scenarios, where the consequences of failure are severe, such as in automated braking systems, the required trust levels for involved components must be exceptionally high. Conversely, in less critical situations, like those involving auxiliary systems such as air conditioning, the trust requirements can be more relaxed. This adaptability is also crucial in prioritizing the execution of trust assessments and the collection of necessary trust properties.

The assessment process is typically executed by different devices (or central entities, depending on the trust architecture). These entities rely on evidence provided by the trustee as sources of trust. This evidence informs about the status of relevant properties. For instance, in the railway scenario discussed in the previous chapter, the automated braking system depends on data from both onboard sensors and trackside units, such as track vacancy detectors, for controlling signals and traffic management. Therefore, it is evident that there is an increased need for integrity and high accuracy of the data collected for decision-making about activating the braking system.

Moreover, trust evaluations consider multiple factors and are always context-specific. Consequently, different trust sources are chosen based on the required trust properties, forming an atomic trust opinion. This opinion solely relies on evidence from trust sources without combining various opinions (e.g., those coming from neighbouring devices). In the railway example, evidence might include data from onboard sensors confirming their correct configuration and as-designed operation, ensuring that data processing is uncompromised. This evidence may be provided by specific agents and tracers deployed directly on these units. At this point, it is important to note that hardware-based trusted computing components play a vital role in securely isolating the execution of security-sensitive functions. This isolation is particularly crucial for the effective tracing of these functions, ensuring that their operation remains secure and uncompromised.

Given the wide range of potential trust sources, which vary based on the nature of the device and the properties of the trust relationship, a tentative list of trust sources is provided in the following section. This list is not exhaustive but aims to illustrate possible sources, categorized into several categories, capturing the different levels of trust. As mentioned earlier, it is also important to factor in the service's criticality when considering trust sources.

3.2.1 Knowledge-related trust properties

The knowledge-based trust properties are identified based on their capability to provide proof and evidence of the system's integrity. This includes elements such as the running operating system, operating firmware, and running applications. Additionally, important aspects include the communication technologies employed, such as the communication interfaces and the cryptographic libraries used. These properties play a critical role in validating the reliability and robustness of the system's core components, thereby preventing harm and ensuring its continued secure operation. The table below provides a detailed description of some indicative properties:

Table 1: Knowledge-related trust properties

Manufacturer reputation	The manufacturer's reputation is crucial when establishing trust between two entities, or within a trust network. More specifically, the credibility and track record of the manufacturer in producing reliable and secure devices are significant factors of trustworthiness. This is a qualitative metric; however, it cannot solely provide assurances about the trustworthiness of a device since the manufacturer's visibility is very limited when the device is deployed in the domain. Still, it is an essential indication that the device is certified, and supports security measures while it has been designed based on standards, in contrast with devices coming from unknown vendors. Also, most manufacturers have very recently started using the MUD [60] profiles that contain crucial information about the security status of the device.
Compliance with standards and certifications	Certifications from well-known certification bodies or governmental organizations are a sound indication of trustworthiness. This means the device has been thoroughly examined and has passed successful

	audits conducted by independent auditors. Additionally, adherence to industry standards and protocols, especially those related to security and data protection (such as ISO/IEC standards or specific industry compliance requirements), is very important. Moreover, compliance with governmental and regulatory standards, particularly in areas like privacy and data security (e.g., GDPR), can indicate a possibly trusted device. The importance of these certifications and compliances depends on the device's domain; however, certain domains, such as medical devices, have very strict requirements.
Security Features	The presence of built-in security features in the device, such as data encryption capabilities, hardware security mechanisms, monitoring agents, and support for a secure boot process, significantly impacts the device's trustworthiness.
Software and Firmware Integrity	The integrity of the running software and firmware is vital for ensuring the security posture of a device [61]. This can be achieved by providing evidence by verifying the integrity and authenticity of the OS and the entire software stack of the device during its operation. Such verification is essential for detecting any attacks or compromises that might occur after the boot-up. A common method is checksum verification and code integrity checks [62], where the system periodically recalculates and compares the checksums of critical files against known "gold" configurations to detect unauthorized changes (this could be performed on the FW level as well [63]). The digital signature verification is another technique, confirming both the data integrity and its authentic source. Additionally, behavioural monitoring is also employed to track unusual applications or OS activities that might indicate a breach. This can be achieved using specialized agents such as eBPF [82] monitoring hooks, which allow in-depth monitoring of specific software or SW parameters in real-time. Tools such as Wazuh agents [64] are also useful, as they gather various parameters and trigger alerts based on predefined rules. Memory scanning is also critical for identifying malware, especially rootkits that might reside in memory without altering disk files. Finally, monitoring changes in critical configuration files and system binaries is another key strategy in monitoring the integrity and security of a device.
Up-to-date operating system and firmware	The release of a new OS (or firmware) version often indicates the presence of existing vulnerabilities that could compromise the system's integrity. Therefore, this strongly suggests that systems operating on outdated OS/firmware versions should be considered less trustworthy as they might be vulnerable to known security risks. In practical terms, once the version of the OS has been identified, it

	is compared against vulnerability databases (i.e., CVE databases of MITRE [65]) where risk scores are included (indicating the severity of the identified vulnerabilities). Hence, mapping these risk values with the running OS version can calculate the trust score. For this specific trust property, it is important to consider that apart from having an up-to-date version of the OS, a secure update should be in place so as to verify if the manufacturer authentically provides the specific version.
Application vulnerabilities	Similar to Operating Systems, vulnerabilities at the application level are crucial to consider. The trustworthiness of an application can be assessed by comparing its version against vulnerability databases. These databases list known vulnerabilities and provide indications of risk, including metrics and the potential harm that could arise if the vulnerability were exploited by malicious actors. Based on these metrics, the overall trustworthiness of the system can be evaluated. Typically, an outdated version of an application is considered less trustworthy, as it may be more vulnerable to known security threats.
Data handling and privacy practices	Another key trust factor is the manner in which the device handles and stores sensitive data. To this end, data encryption protocols and mechanisms, particularly cryptographic libraries, are employed to protect sensitive data both in transit and at rest. Additionally, data integrity mechanisms, such as checksums and digital signatures, are crucial in ensuring the integrity of stored or transmitted data. Regarding privacy, devices should implement specific mechanisms tailored to the context and regulations of each domain (e.g., data minimization and retention policies in line with GDPR). To assess these properties, specific cryptographic inventory tools can be utilized (like Cryptosense [66]). These tools are capable of scanning the systems and identifying the cryptosystems and protocols in use, including detailed information on running processes, the cryptographic algorithms employed, key sizes, and the total number of key pairs utilized over the system's lifetime.
Communication interface	Communication technologies include a wide range of protocols and hardware interfaces that devices use to communicate with others or central and backend systems. These technologies range from Wi-Fi and short-range Bluetooth to long-range technologies like LoRa [67] or other cellular networks (4G/5G, etc.), and wired connections (primarily used in industrial applications). Moreover, most devices use multiple connection interfaces for various functionalities. For instance, a gateway might use its wired connection to connect to backend servers for data transmission, while its connection to deployed sensors could be wireless-based. Communication

	technologies are critical as several vulnerabilities are known for each of them, some critical and others not. For example, there are several obsolete Wi-Fi security protocols like WEP, known for their vulnerabilities and susceptibility to attacks and hacking [68]. Likewise, older versions of Bluetooth have known security flaws, and even some newer versions can be vulnerable if not properly configured [69]. Even Zigbee [70], often considered highly secure, can sometimes have flaws leading to critical vulnerabilities [71]. Therefore, the difficulty in compromising data transmitted over these network interfaces depends on the communication technology used, and, of course, this affects the level of trust. Direct wired links are generally considered much more secure than Bluetooth interfaces, while the presence of multiple communication interfaces in a device increases its attack surface, thereby reducing the overall trust level of the device.
Protection of communications	The robustness of authentication protocols is another important aspect of trust. For instance, TLS/SSL [72] and OAuth2[73] are widely used protocols that provide end-to-end authenticated and encrypted communication channels. These protocols are essential when transmitting data from one device to another or to backend services. However, there are several insecure protocols still in use, such as pre-shared authentication keys (PSK). PSK involves a shared secret key between two entities and can be highly insecure if the keys are not managed correctly, especially in scenarios with multiple devices where the keys are assumed to be securely disseminated in a previous step. Additionally, several obsolete methods, such as the conventional scheme of basic authentication using a username and password, are still commonly used in the IoT domain. This method is vulnerable if the passwords are weak, reused, or transmitted over unencrypted channels. Moreover, several legacy protocols, such as older versions of TLS and SSL (e.g., SSL v2.0, SSL v3.0, TLS 1.0, TLS 1.1) [74], are considered insecure due to well-known vulnerabilities. In the context of communication between two entities, several other techniques should be implemented to provide protection against attacks such as spoofing, replaying, and on-the-fly modification attacks. These protections could be integrated into the transmitted messages and checked on the receiving end of the channel, like HMAC codes. Lastly, a common misuse in IoT devices is the use of factory default settings and passwords, making them susceptible to adversaries who can easily find these passwords in public repositories and manuals. Hence, the use of strong authenticated and encrypted channels in system communication plays a critical role in assessing the trust level of a system.

Hardware security mechanisms	Hardware security mechanisms found in most modern devices are isolated computing units within the processing unit (or the memory). They are mostly responsible for managing the storage of cryptographic material as well as for performing security-sensitive functionalities, such as encryption/decryption, hashing, digital signatures, etc. Additionally, they can be used for software isolation. These mechanisms could be Trusted Platform Modules (TPMs) [75], Physical Unclonable Functions (PUFs) [76] or secure enclaves [77] and play a crucial role in maintaining the confidentiality and integrity of information; thus, their presence is critical in assessing the trustworthiness of a device.
Management and storage of cryptographic material	This trust property is related to the hardware security mechanisms described above. Secure cryptographic key management and storage are vital in assessing the trustworthiness of a device. This involves focusing on the cryptographic key, its generation, storage, and resistance to attacks. The generation of keys must be sufficiently random, which depends on the seed used and the type of algorithm. The type and length of the key are also crucial. After generation, the distribution of keys is critical to minimize interception by malicious actors. Also, implementing key usage policies is important to ensure that the keys are used and employed correctly throughout their lifecycle, in accordance with predefined and intended uses. Equally important is the secure storage of cryptographic material, using technologies like HSMs and TPMs, or secure enclaves within processors. These aim to physically and logically isolate the cryptographic keys from potential threats. Additionally, employing encrypted storage and strict access control schemes further enhances key security. The lifecycle of the key should involve regular refreshment to reduce risks associated with long-term exposure. This includes the revocation and secure destruction of keys when they are no longer needed or if they become compromised. However, several deployed systems struggle with these issues due to factors like complexity, scalability, and limitations of legacy systems that cannot support such techniques. To this end, implementing a secure key management policy is critical in assessing the trustworthiness of a device. This also impacts compliance with regulatory and industrial standards, and is fundamental in protecting sensitive data and maintaining the confidentiality and integrity of information.
Secure Boot	The secure boot process prevents unauthorized software from running during device boot-up [78]. It performs that by verifying the digital signatures of the firmware and sometimes critical OS files, against a list of trusted sources released by the manufacturer. This verification process ensures that each component of the boot

	sequence is trustworthy by verifying its integrity and authenticity and that it has not been tampered. This allows the device to effectively block malicious software like malware or bootkits from compromising the boot process [79].
Isolation of security critical functions	This aspect also serves as a trust factor, relating to hardware security mechanisms and secure key management and storage, as previously mentioned. The isolated execution of security-critical (and safety-critical, depending on the application) functions refers to a technique where sensitive operations, such as cryptographic processes, authentication, and authorization, are carried out in a secure, segregated environment. This approach ensures that these critical functionalities are isolated and safeguarded against threats and vulnerabilities present in the broader system. As noted earlier, this typically involves the use of TPMs. In addition to hardware-based solutions, this isolation also encompasses software-based techniques, such as virtualization and containerization. These methods create isolated execution environments that are separate from the main operating system. Such isolation is instrumental in mitigating risks associated with unauthorized access or tampering. Therefore, executing security-critical functions in isolation (including tasks like key generation and updates) enhances the overall security posture of the systems and, consequently, their level of trustworthiness.
Cryptographic libraries separation	This trust property, also related to hardware security mechanisms as previously described, emphasizes the importance of cryptographic separation and the integrity of crypto libraries. This is particularly crucial in multi-user or multi-tenant environments, where different users access the same system under varying authorization and access policies. This technique not only segments sensitive data but also protects cryptographic materials, such as keys. It is essential for maintaining separation and restricting access solely to authorized entities. Cryptographic separation ensures that data and keys belonging to one user or group cannot be accessed by others, even if other security layers are breached. Additionally, the integrity of cryptographic libraries, encompassing algorithms and protocols, is paramount. A flaw or vulnerability within these libraries can compromise the security of dependent applications. This necessitates that these libraries are not only well-implemented but also regularly updated. When evaluating the trustworthiness of a device, the security measures in place to protect these libraries from unauthorized modifications or tampering must be assessed. Such intrusions could introduce backdoors or diminish their security effectiveness. Thus, cryptographic separation and the integrity of cryptographic implementations are key to safeguarding sensitive

	information from unauthorized access and maintaining the trustworthiness of the system.
Secure SW/FW updates	When updating the software and firmware of a system, implementing secure mechanisms is crucial to ensure the trustworthiness of the updates and patches being installed. This includes verifying the authenticated source and using an encrypted communication channel. Encryption prevents unauthorized parties from viewing or tampering with the source code of the updates, which could potentially introduce backdoors. Additionally, the source of the updates should be authenticated through digital signatures to ensure that the patches originate from a legitimate and trusted source, such as the manufacturer. Secure and authenticated communication channels are also used to deliver these updates to the relevant systems. This further safeguards against interception or manipulation by malicious actors during the transmission of the updates. Therefore, the presence of these appropriate mechanisms is a critical indicator of trust. They protect not only the updates themselves but also preserve the overall security posture of the systems. If an attacker were to compromise an update, they could potentially introduce malware or backdoors, leading to more significant security breaches. Implementing this approach is essential in building and maintaining trust in a system.
Run-time integrity	When considering the runtime of a device, there are several attacks that can be executed which may not be easily detected by the host device and its implemented security measures. These attacks can compromise applications and, through them, gain access to sensitive data or the device's sensitive memory areas. Such attacks are often used to recover secrets from recently used spaces in memory. Detecting these attacks is very difficult due to zero-day vulnerabilities, which are currently unknown in the community, and due to backdoors potentially present in third-party libraries used in the application's source code. This latter method is known as a supply chain attack. There are several techniques to protect against such attacks. For example, monitoring and tracing low-level properties of the device, such as memory interactions and other attributes, can indicate malicious behaviour. Tracing specific sensitive areas of memory enables the detection of abnormal levels of memory interactions. These measurements can be compared against behavioural (probabilistic) profiles that are constructed during the device's design phase. These profiles describe the device's nominal behaviour, including the anticipated execution of tasks and the expected use of resources. Additionally, tracking specific functions of an application that accesses sensitive and secret data and configuring an overflow threshold in the memory controller can be

	effective. To this end, it is possible to protect the system during runtime from advanced attacks and even zero-day exploits by identifying abnormalities in system operation. Employing such monitoring agents and segregating memory areas by predefining highly sensitive functions and memory locations are crucial strategies in enhancing runtime security.

3.2.2 Experience-related trust properties

Unlike knowledge-related properties, which are primarily based on the device's integrity without considering its behaviour during runtime, these properties focus mostly on reliability over time and abnormalities at the device or network level (e.g., unjustified spikes in processing or memory consumption).

Table 2: Experience-related trust properties

Historical reliability	In assessing the reliability of a device, it is crucial to consider its historical performance record. This includes key statistical data such as failure rates, uptime statistics, and consistency in performance under varying conditions. Additionally, the consistency of the provided data is another important dimension to be considered as any discrepancies over time may indicate potential malicious malfunctions. This evidence data is an important empirical indicator, providing insights into the potential compromise (or robustness) of the device.
Device-centric abnormalities	During the design phase of a device, its nominal (expected) behaviour can be projected. This involves outlining the anticipated execution of tasks and the expected utilization of resources in the intended environment and application. This projection can be further expanded to include memory interactions and other attributes that may signify malicious behaviour. Monitoring specific sensitive areas of memory, for instance, can reveal abnormal levels of interaction, potentially indicating a security breach. Therefore, behavioural profiles should be constructed during the device's design phase. These probabilistic profiles will include operating parameters, such as task execution and resource usage patterns. By comparing real-time measurements against these established profiles, any discrepancies could indicate trust violations, leading to a lower level of trustworthiness. To effectively monitor these parameters, it is essential to deploy monitoring agents and tracers at the device level (e.g., Prometheus [80], New Relic [81], eBPF [82], Telegraf [83]). For example, in a gateway that collects data from sensors every 10 minutes, agents can be installed to track the system's data processing activities and memory usage. These agents could observe any unusual spikes in memory consumption or

	unexpected data transmissions, which could signal a potential security breach or system malfunction.
Network-based abnormalities	Similar to device-level properties, network-based data can serve as a reliable indicator of trust. This means that through a network-based Intrusion Detection System (IDS) [84], network traffic data can be collected. This collection can take place both during the design and the operational stages of the device. During the design phase, patterns of data transmission can be gathered. However, the most effective profiling is performed during operation, when interactions with other devices occur, allowing for the development of a complete network profile. This profile models the overall behaviour, such as the interactions between a gateway and other devices deployed in the infrastructure. Any anomalies in behaviour or suspicious activities will lead to reduced trust levels.

3.2.3 Behaviour-related properties

This last category pertains to the experience gathered from other entities (devices) in the network, in contrast to the two previous categories that examine the behaviour of the device solely from its own parameters and perspective. This category involves collecting data from several devices within the network, which can then be used in the previously presented aggregation formula to calculate an overall trust score at the network level.

Evidence-based network feedback	The feedback-based reputation approach evaluates a device's behaviour, incorporating feedback from other devices within the network. This feedback might include direct reviews from devices that have previously interacted with the assessed one, providing insights into its real-world reliability, performance, and any encountered issues. Therefore, the reputation is established based on observations of its behaviour, considering referrals or ratings from other network nodes. Moreover, other factors, such as network experience, consistency, and redundancy checks, are also considered. For instance, the quality and consistency of data provided to various devices are compared, verifying their accuracy. The type of data provides a trust indication and plays a critical role in determining its reliability and its trustworthiness level respectively. Inputs from different devices can be cross-referenced to identify any discrepancies, indicating potential malicious activity. This reputation-based scoring then informs the rest of the devices in order to recalculate their trustworthiness.

3.3 Trust provisioning

To effectively retrieve information related to device properties and behaviour, a variety of tracers or tools can be utilized. These tools operate on multiple levels, from external network evaluation to in-depth analysis of software characteristics and internal system functionalities.

For network monitoring, various applications can analyse network traffic, identifying anomalous patterns and potential security breaches, such as Wireshark [85]. Additionally, Intrusion Detection Systems (IDS), like Snort [86] and Suricata [87], play an important role in detecting malicious activities. The information from the network level is particularly important for making decisions related to the behaviour of a device within a network.

On the device level, evaluating trustworthiness is more complicated, as the existing tools can capture different levels of the device and software stack. For this purpose, several tools are used such as tracers and monitoring tools (e.g., Prometheus [88], New Relic [81], eBPF [82], Telegraf [83], Wazuh agents [64]). Also, log analysis software, like Graylog [89], can examine log files generated by devices, focusing on detecting unusual events or trends that may raise security concerns. Furthermore, Security Information and Event Management (SIEM) systems, such as IBM's QRadar [90], play a vital role in analysing data from diverse sources, including individual devices. Complementary to SIEMs are Endpoint Detection and Response (EDR) solutions that provide another layer of defence by providing continuous monitoring of endpoints through agents deployed directly on devices, such as IoT devices. These are essential for identifying and responding to threats in real-time, thereby reinforcing endpoint security. Performance Monitoring Tools complement these efforts by tracking key performance metrics like CPU usage, memory utilization, and response times, which is crucial for maintaining optimal device performance and quickly identifying operational anomalies.

Vulnerability Scanners, such as OpenVAS [91], are essential for the proactive identification of known vulnerabilities within devices. By assessing potential risks and the overall security posture, they are fundamental in maintaining a robust defence against evolving cyber threats. Energy Consumption Monitoring Tools offer a different perspective by tracking the power usage of devices, providing insights that can be instrumental in detecting issues related to efficiency and potential hardware malfunctions. Lastly, Software and Firmware Analysis Tools, like Binwalk [92], are crucial for delving into the core of device security. They scrutinize device software and firmware for hidden vulnerabilities, backdoors, or malicious code, ensuring the foundational security of the device. These tools gather critical data and insights, enabling an understanding of how devices function, interact, and adapt to diverse scenarios. This information is then used to retrieve the required information to assess the trustworthiness of the device, as will be presented in the next chapter that analyses the critical trust factors and properties that need to be assessed.

3.4 Trust management

After gathering information from relevant trust provisioning tools (as described in the previous chapter), the consolidated trust quantification and overall assessment should be performed. This process involves various techniques, largely dependent on the setup environment. For instance, whether devices are permanently deployed in the domain or if new devices continuously join, and

more importantly, on the characteristics of the device and their computational capacity, which determines if the trust assessment logic and models can be deployed directly on the device.

The simplest method for trust assessment, without any direct device intervention, is the centralized approach. In this approach, the trustworthiness claims (information collected by the devices – see 3.2 about trust properties) are consolidated at a central entity responsible for assessing the trustworthiness level of each entity, regardless of the chosen evaluation model. Figure 4 presents a simplified illustration of this scheme, where the trust assessment is a centralized entity responsible for assessing the level of trust. The grey arrows represent interactions among the devices, while the orange ones represent interactions between the devices and the trust assessment. This includes collecting necessary trustworthiness claims and sharing the required information back with the devices.

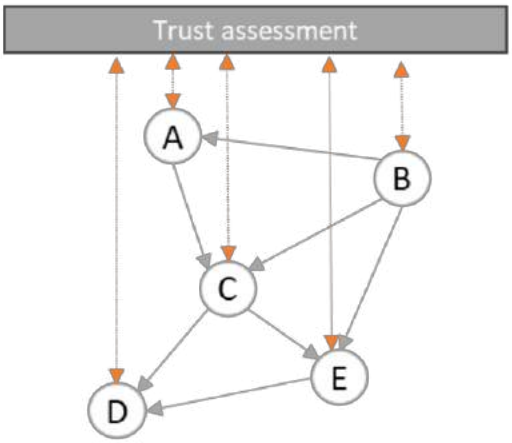

Figure 4: Centralized scheme

A semi-decentralized scheme can complement this approach, where transmitted data is recorded in a distributed ledger, making it accessible to the other entities of the network. In this setup, the centralized trust assessment logic retrieves information directly from the ledger, calculates trust scores, and stores the output back on the ledger. This open accessibility is crucial, especially when a device aims to interact with another without any prior knowledge. By accessing the ledger any device can obtain already assessed trust values (that characterize the current trust level of each device) and can decide whether interaction is safe based on this score and the criticality level of the service to executed or data to be transmitted. This method is particularly beneficial for reputation-based trust modelling methods, where trust opinions from multiple entities are aggregated to establish an overall trust level. The role of the ledger is vital in such settings as it ensure the integrity and auditability of the recorded trust information, establishing transparent trust relationships across the network. Figure 5 presents an illustration of this scheme, where trustworthiness claims are recorded in the distributed ledger (indicated by orange arrows). Through this ledger, the trust assessment can access the required information to assess the level of trust. Instead of sharing the outcomes directly with the devices, this information is stored back in the ledger and can be retrieved on demand by any device.

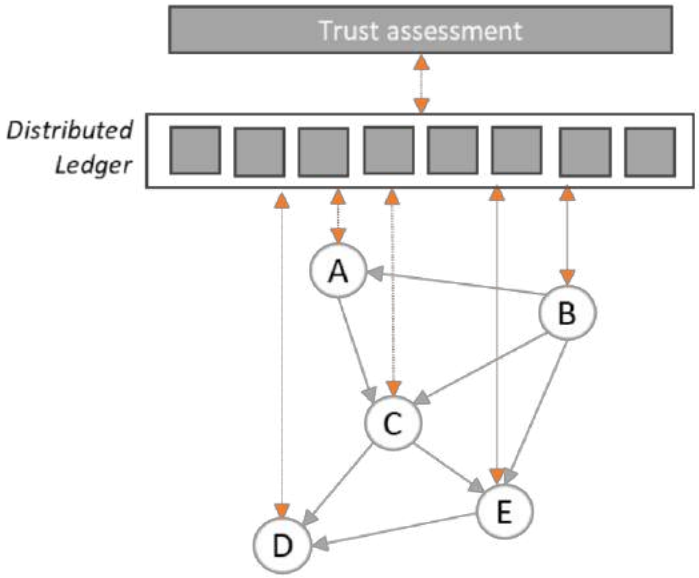

Figure 5: Semi-decentralized with the use of a distributed ledger

An alternative is the completely decentralized scheme, where devices independently assess the trust of others. This requires substantial processing power and computational capacity at the device level to execute trust algorithms. Additionally, devices must have hardware-based security mechanisms like TPMs to securely perform trust assessments and security-sensitive functions, such as encryption/decryption and outcome signing. The direct assessment of the trustworthiness levels can also facilitate further attestation of devices based on challenges requested by peers. This is very useful in scenarios requiring an advanced trust level for sensitive and critical processes, where additional checks and proofs of trustworthiness are necessary. Moreover, this approach aligns well with using a distributed ledger as an intermediary to share and disseminate trust information across the network. While all these approaches are useful, selecting the most appropriate one requires careful consideration of factors like scalability, performance, and latency in quantifying trust levels. Figure 6 presents this scheme, where the trust assessment is deployed at the device level in the form of decentralized trust agents (represented by orange squares). The devices are capable of directly assessing the trust level of others (indicated by orange arrows) and can record/share the outcomes through the distributed ledger.

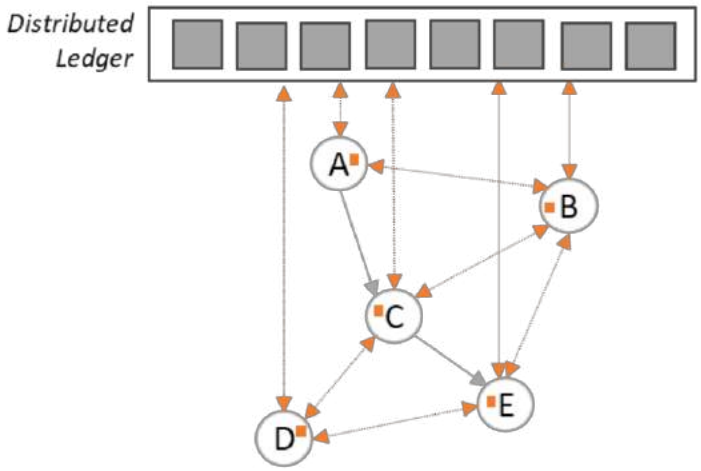

Figure 6: Decentralized scheme

3.5 Outcomes on Trust Modelling in Distributed Systems

This chapter presented the trust modelling in the domain of distributed systems, highlighting mostly their applicability as well as their limitations. The main contribution of this chapter, which is the core part of this study, is the identification of trust properties, focusing on aspects such as knowledge, experience, and behaviour of the systems to derive to an evidence-based trust score. The relationship between these trust properties and their impact on trust decisions in distributed environments is also analysed.

Also, this chapter discussed the need to establish a hierarchy and categorization of the devices and running services based on their criticality, considering factors like latency and downtime, especially in dynamic environments. This enables the accurate and on-time assessment of the trustworthy state of the device, considering the criticality and prioritizing the assessment of highly critical functionalities. Furthermore, the chapter addresses the issue of trust provisioning, discussing how trust is established, maintained, and updated within a network. Key challenges and potential solutions in trust management are also explored.

The chapter concluded with a focus on trust management, particularly emphasizing the importance of consolidating trust quantification and overall assessment after gathering information from relevant trust provisioning tools. This process is highly dependent on the setup environment, such as whether devices are permanently deployed in the domain or if new devices continuously join in dynamic environments. A crucial factor in this process is the characteristics of the devices, especially their computational capacity, which determines if the trust assessment logic and models can be deployed directly on the device level. To this end, three approaches are presented based on the setup of the trust assessment model and the placement of the logic in the network.

4

Applicability of the trust model and its limitations

This chapter examines two specific scenarios in which the trust model presented in the previous chapter can be effectively applied within the context of railway networks. Additionally, it discusses the limitations of its applicability. Still, the primary objective of this chapter is to showcase some practical applications of trust assessment. It is important to note that once the proposed model is fully developed, it will require further experimentation and evaluation, and these criteria are elaborated in the final subchapter.

4.1 Trust scenarios in the case of railways

As an indicative trust scenario, the presented trust model can be implemented in the railway's domain, thus providing a realistic assessment of it. Railways are very complex systems, consisting of several heterogeneous systems and various technologies to ensure efficient and safe travel for passengers [5]. At their core are the *Signalling Systems* used to control the movement of railway traffic, which include electronic interlocking and level crossing systems, crucial for directing traffic and ensuring safety on the tracks. That is coupled with several track-side units that share information between the trains and the Command-and-Control centre. The *Command-and-Control* includes systems such as Automatic Train Control and Automatic Train Supervision, which manage the movement of trains alongside the Energy Traction system that powers the trains. The on-board *Auxiliary Systems* support the primary functions of the railway, such as energy systems, HVAC, and lighting systems, mainly ensuring comfort but also being very critical in case of emergencies. Also, there are several other systems dedicated to the passenger comfort like Passenger Announcement and Information Systems. Lastly, *Telecom Systems*, including dedicated radio and wired on-board networks, as well as voice communication and timekeeping systems, are essential for effective communication across the railway infrastructure, safeguarding that all operations are well-coordinated and efficient.

Together, all of these systems create a very complex network of distributed SoS, coming from various vendors and manufacturers, which is further expanded by the presence of legacy systems with obsolete interfaces and outdated processes. Another important characteristic is the distributed nature of the deployment of these systems: some are installed on-board trains, others are centralized in command-and-control centres, and additional systems are positioned along the railway tracks,

making the overall monitoring and control of the infrastructure very challenging. In the following subchapters, two hypothetical attack scenarios will be explored to showcase the trust management's potential applicability within the railways domain.

4.1.1 Signalling and automatic train control system scenario

In this attack scenario, the attacker targets the railway's signalling system and automatic train control system, aiming to cause a train accident. Although the probability of such an attack is relatively low, its high impact potential increases its significance and the potential value of using the proposed trust modelling methodology.

There are several critical components of the railway network involved in this scenario. For instance:

- Command and Control Systems, including Automatic Train Control.
- Trackside Units responsible for signal control and traffic management.
- Interlocking Systems to arrange the movement of tracks.
- On-board Automatic Train Control mechanisms.

The following schematic (Figure 7) provides a simplistic representation of the diverse components involved in this scenario. It presents the deployment and interaction of various systems within the railway infrastructure, including those installed on-board the trains, situated in command-and-control centres, and positioned along the railway tracks, which interact with the interlocking systems exchanging critical signals.

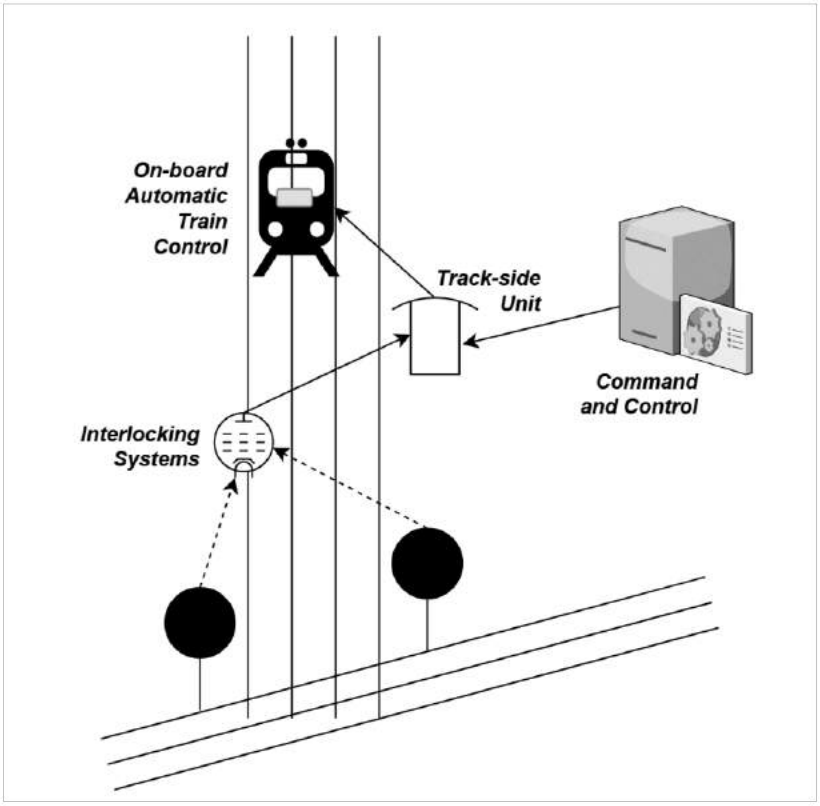

Figure 7: Signaling and automatic train control system scenario.

In this attack scenario, the attacker first gathers essential information to manipulate the railway systems. This may include details like the types of requests made by the traffic management system,

IP addresses, and other specific technical data related to the systems deployed on the tracks, such as data from the interlocking systems. It should be noted that while the means of obtaining this information are important, they are not the primary focus of this analysis. Based on this knowledge, the attacker develops malicious software to infiltrate the command-and-control systems, aiming to remotely take control of railway junctions and train movement. The goal is to inject false data into the track-side unit responsible for traffic management and for sending information to the train's onboard systems. Using the infiltrated software, the attacker can initiate an attack, gaining control over critical junctions and trains within the railway network.

In this scenario, deploying a trust model can help preventing such an attack. The primary objective is to evaluate the trustworthiness of all network entities, especially during information exchange between critical components. This includes an in-depth assessment of each involved component (e.g., trackside unit). A centralized monitoring and trust assessment system should be established to achieve this. The trust assessment could be deployed in a centralized server, for instance, in the command-and-control center, that aggregates all monitoring data from the network's elements and devices and combines this with other behavioural and historical characteristics of the devices (e.g., network patterns). It also consolidates information from all network entities, analysing and cross-referencing it to identify potential vulnerabilities and threats. By monitoring behavioural patterns and communication between these entities, the system can detect anomalies that might indicate an attempted attack. Moreover, the trust assessment could incorporate real-time analysis capabilities by deploying monitoring agents to the most critical devices in the network. Of course, there is an assumption that the devices have been characterized before deployment and their expected behaviour has been documented as a reference model, which can be compared to actual measurements.

Once the data is collected, the next step is to quantify the trust level of each device, which could vary based on the level of criticality of the service executed. For example, if a train is requested to be redirected to other tracks, this level could be set to high, meaning that the communication between the trackside unit and train, as well as the device sending the information, should be thoroughly assessed.

Nevertheless, another critical case is the following: the operator should collect information about the current traffic from several track-side units to safely redirect a train to another track. This involves collecting information from several systems. Before trusting them, these systems should be assessed at the highest possible level of trustworthiness. If the level of trustworthiness falls below a specific predefined threshold, then the gathered information should not be considered trustworthy, nor should the device be trusted. Additionally, in the case of a software update on the onboard Automatic Train Control mechanisms, this could be set to "extremely high" to ensure that the overall process is secure and reliable, also verifying that the sender of the update is authentic. If the sender is compromised, there is a high risk of deploying malicious software to the onboard units.

4.1.2 On-board systems scenario

Another possible scenario could involve a malware targeting a train's on-board systems, causing a false emergency stop. In this scenario, several on-board systems come into play, such as:

- On-board Automatic Train Control mechanisms.

- Train Brake Systems.
- Other sensors and onboard gateways that collect critical information.

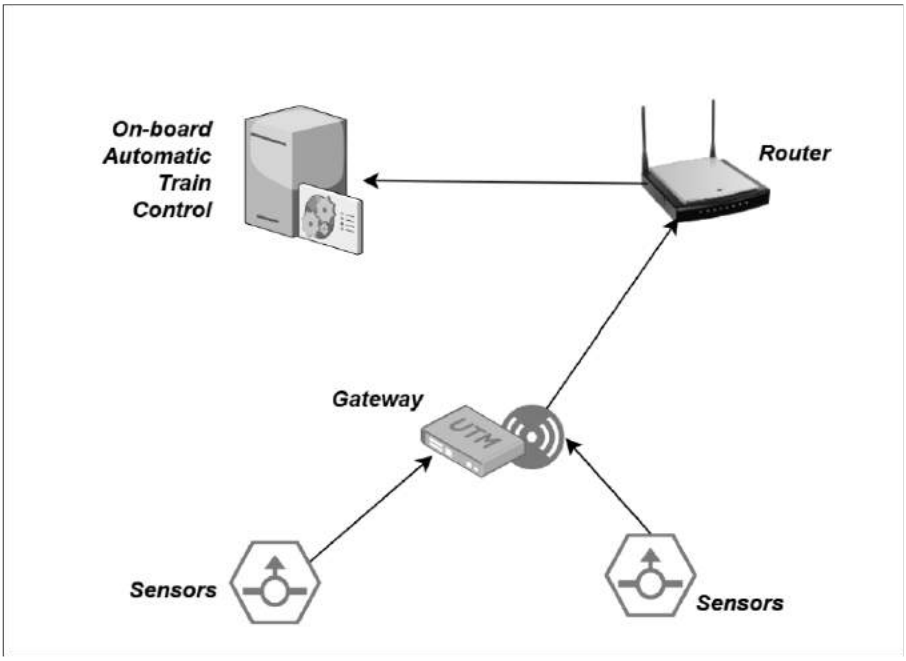

Figure 8: On-board systems scenario

In this scenario, the attacker can inject a malware into the on-board systems of the train either through physical access or by using removable devices employed in OT systems for various purposes, such as firmware updates. Additionally, the attacker can inject malware into auxiliary and secondary systems (not used for train control), like an onboard gateway, through a wireless connection interface that is not updated to the latest version. The goal is to propagate this malware to the rest of the onboard systems, allowing the attacker to feed false data to the train's automatic control system.

Similar to the previous case, the malware could be deployed to the centralized systems of the remote-control system of the trains in the command-and-control centre, enabling the attacker to take over the systems and gain remote access. This malware can facilitate interference with the communication with traffic supervising systems and enable remote manipulation by injecting a payload that disrupts these systems. This could lead to a false stop of train traffic, which is extremely dangerous apart from the disruption of the traffic.

As mentioned in the previous example, the role of trust management is critical in such scenarios. It enables the assessment of the trustworthy state of the intermediate components under strict security and timing requirements. This is crucial in scenarios like a critical braking event, where the on-board systems of the train must trust other systems for sharing critical information to enable real-time cooperation. The objective here is to safeguard communication in real-time and safety-critical conditions during operations, while preventing the propagation of vulnerabilities from one system to another. This is essential to protect passenger safety, data integrity, and service availability from being compromised by malicious actors and adversaries. In scenarios where decisions are time-critical and latency is a vital factor, it is also important to adjust the criticality of the assessed systems

and prioritize the execution of assessments to ensure trust-related information is delivered promptly. The possibility of systems impersonation should also be considered and assessed in the context of distributed systems deployed in uncontrolled environments, where external entities, may have physical access (e.g., track-side units), can potentially compromise the systems.

To further evaluate the trust model's applicability, a range of performance indicators and criteria must be considered, with a primary focus on its suitability, performance, and scalability within in large networks. These criteria are further explained in Chapter 4.3.

4.2 Limitations of the proposed methodology

There are several limitations and challenges to the widespread adoption of the trust modelling methodology as described in the previous chapter. Trust modelling (and trust management) are particularly challenging in highly dynamic environments, where devices continually appear, move between domains, and disappear. This can easily occur in less controlled environments, like smart cities, where unknown devices continuously join networks. Given this volatile structure, the "Zero-Trust" paradigm should be considered, which requires finely controlled access to data and resources, avoiding implicit trust assumptions. Instead, all interactions should depend on the trust level of the involved devices. Therefore, interactions must be monitored using contextual metrics beyond the conventional ones, and the trustworthiness of all elements should be continuously re-evaluated based on their behaviour.

A significant issue in this context is the onboarding of devices into the network, which forms the cornerstone of trust. However, a notable challenge is the so-called "cold start issue" [93], which occurs when a device joins a network for the first time without any prior interactions. Hence, it is crucial to assess the device initially while maintaining network transparency. The issue is twofold: on one hand, it is vital to enable smooth device onboarding by focusing on device-centric aspects (e.g., manufacturer reputation, updated firmware). On the other hand, assigning a low trust value could be too risky, potentially blocking the device from interacting with others. Likewise, allowing the device to start interacting without sufficient assessment may lead to network corruption.

Another aspect is related to the capabilities and requirements of monitoring agents and tracers. The devices are often lightweight in terms of processing power, and their computational capacity may not be sufficient to integrate power-consuming tracers capable of tracking every action of the device. Additionally, there are restrictions, as several IoT devices do not even host an operating system and are primarily based on bare-metal firmware. Deploying a tracer should also consider the specific needs of the device, especially in highly critical situations aimed at enhancing assurance without compromising applicability and performance. Consequently, there is a need to develop tracers that can effectively balance performance with security.

A different important topic concerns the aggregation of trust scoring, where devices potentially need to operate under completely decentralized conditions without a central authority managing trust and calculating trust scores. This demands more robust computational and processing capabilities at the device level to allow for the deployment of trust agents capable of performing such calculations and deriving trust scores independently. On top of that, a large number of existing devices lack hardware-supported roots of trust, making it even more challenging to execute trust-related operations directly on the device. These measures, along with the necessary information, are

essential for efficiently and reliably managing the lifecycle of devices, which is crucial for maintaining high Quality of Service (QoS). In this regard, metrics and event collection will primarily be processed on the devices, enabling them to utilize services and data from others. However, this raises another question of how to disseminate trust information to other devices, especially when managing trust-related incidents (such as trust revocation, security policy violations, etc.). As a result, reliable means of communication and trustworthy data sharing should be considered, particularly when a device needs to consult others and assess the trust level of a target device before initiating interaction.

Lastly, an important dimension to consider is privacy and the provision of sensitive information by devices. To address this, the self-sovereign identity paradigm can be adopted. Under this paradigm, devices themselves control the responsibility of trust assurance by constructing verifiable presentations, which are done without exposing any sensitive information to external parties. These verifiable presentations are created using verifiable credentials that are issued and maintained at the user level. This ensures that each user discloses data based on specific attributes and the level of access they are willing to provide. Such attributes could represent various types of information, including device-related information required to quantify trust (e.g., the version of the operating system).

4.3 Further evaluation criteria

To further evaluate the applicability of the trust assessment methodology, several other factors should be considered in real-world scenarios, such as the performance of the trust quantification, which is very important (mostly) in time-sensitive scenarios (like the one presented previously). Therefore, in scenarios where trust assessment must evaluate the trustworthiness of a system within specific time constraints set by the security operator or the user (e.g., maximum latency in critical braking), it should be fast enough to avoid unacceptable delays in the decision-making process. A “late” decision regarding the low trustworthiness of a system could lead to safety issues, especially when trusting potentially malicious systems.

Another aspect that must be assessed relates to the scalability of the solution, specifically how it could scale in large networks, and how its performance is affected in an evolving network in terms of the number of devices. Also, it should be considered that in a large network, the trust properties to be taken into account in the quantification process may be numerous, thus requiring substantial processing power. In such cases, the assessment should specify the upper acceptable boundaries of latency in terms of time and the number of assessed nodes and devices in the network. This not only affects the quantification but also the process of evidence generation and the time required for further analysis so that it can be consumed efficiently by the trust model. To analyse the performance of the trust, detailed benchmarking is required once the algorithms have been developed, considering the varying time constraints and the complexity of the environment (i.e., how long it takes to compute all required trust properties so that the model can calculate its output and whether this still meets the time requirements).

4.4 Future Work

The analysis presented in this thesis primarily focuses on providing a novel approach to modelling trust in the domain of distributed Systems-of-Systems, taking into account the peculiarities and diverse nature of the systems involved in such settings.

The future research plan relates to the actual design of algorithms that will realize the trust assessment methodology. This involves creating a proof-of-concept for the proposed trust quantification methodology and the overall trust assessment architecture, detailing how these should be realized. Additionally, there are several trust decision-making algorithms that can be utilized for quantifying each of the trust evaluation models, as analysed in Chapter 2.3, while considering the trust quantification techniques in specific contexts (Chapter 2.4), for instance, the use of belief theory in large networks of devices.

To support autonomous decision-making and decentralized trust quantification, the integration of machine learning algorithms should be analysed. However, the real-time and ultra-low delay requirements of sensitive applications must be respected. Machine learning applications sometimes require interaction between cloud servers and devices (depending on the training of the model). If the model needs to be trained directly on the device, then suitable machine learning techniques, such as TinyML or federated learning approaches, could be investigated for the trust calculation. Furthermore, the use of machine learning techniques in the quantification and assessment of trust introduces a need for explainability. Until recently, only optimization and efficiency metrics have been primarily assessed for such models, with less focus on specific quality and security requirements. Additionally, the trustworthiness of these models, particularly concerning the origin and correctness of the training data, should be assessed, as inaccuracies may lead to misleading decisions. This is crucial in the case of autonomous systems, which adapt and evolve over time based on their interactions within deployed environment, meaning that trust assessment models should also adapt accordingly. This requires a machine learning-based model capable of evolving with changes in functionality, the environment, and the evolution of trust between devices.

Another aspect concerns human-machine interaction, as it expands the notion of trust to include aspects and requirements about trustworthiness and safety due to interaction with humans. In this regard, it is also critical for the systems to be able to trust humans, as they are part of the environment. This leads to questions about how we can detect, explain, and assess human behaviour, model this behaviour, and specify trust from a social perspective as if recommendations from the users are malicious, this could affect the trust assessment, propagating false information to the network. Even if the intermediary is honest and does not modify the input received from others, it can still disseminate false information.

Moreover, the adoption of the developed model faces several limitations and challenges, as presented in the previous chapter. These challenges, which will be further studied, include the onboarding of devices into networks, particularly the “cold start issue,” which presents a significant challenge in balancing initial assessment with network integrity. Another aspect to be analysed is related to the characteristics of the tracers, considering the miniaturization of systems and their evolving presence in lightweight devices. This requires a careful balance between performance and security, as decentralized trust scoring poses complexities, requiring devices to independently

calculate trust scores without a central authority. Privacy considerations should also be explored, following the self-sovereign identity paradigm that enables devices to control trust assurance without revealing sensitive information via verifiable presentations and credentials. This is particularly important in cases where devices record and hold sensitive information, such as connected medical devices. Another important aspect to consider is the operational assurance of the trust model itself, meaning how the model is protected against external threats and attacks that may attempt to alter its configuration when quantifying the trustworthy state of a device, or even trying to change its behaviour or the already generated outcomes. Therefore, it is critical to study how to safeguard the trustworthy operation of the model.

5

Conclusions

Building trust relationships between systems in the era of autonomous systems and complex industrial Systems-of-Systems presents a significant challenge. This challenge affects not only the security posture of these systems but also poses security and safety challenges that may harm other systems as well as humans. Therefore, it is crucial to define how trustworthiness can be quantified in terms understandable to systems and devices, considering they lack the natural ability to evaluate relationships and other risk factors like humans.

In this direction, this thesis reviews the fundamentals of existing trust modelling, examining current state-of-the-art practices. It is worth highlighting that different research disciplines approach this problem in various ways and define trust differently as well. Additionally, a detailed analysis of existing trust evaluation models is presented, focusing on methods for quantifying trust and decision-making processes crucial for deriving trust metrics.

The main contribution of this thesis includes defining a trust modelling methodology, providing detailed steps for defining trust objectives, identifying trust entities, assessing trust relationships within a system or a network, and, most importantly, defining trust criteria and properties, as well as a formula for quantifying trustworthiness. A key aspect is the identification of potential trust properties that can be measured in each system, considering various dimensions such as knowledge, integrity, behaviour, and experience of other entities in the network. Another significant contribution, is the study of trust provisioning techniques, exploring how trust properties can be measured, and through which mechanisms and applications. Lastly, the study explores three trust management techniques based on specific system requirements for an even more efficient quantification of the trustworthiness, considering how trust properties could be collected and the decentralization aspects of the quantification process itself.

Moreover, the thesis examines two specific, realistic scenarios in the railway domain to assess the applicability of the trust assessment solution. It also explores the limitations of its applicability and concludes with potential research directions and future work.

References

[1] Moore, G. (1998). Cramming More Components Onto Integrated Circuits. Proceedings of the IEEE, 86, 82-85. https://doi.org/10.1109/JPROC.1998.658762.

[2] Wang, F., Xu, J., Wang, X., & Cui, S. (2017). Joint offloading and computing optimization in wireless powered mobile-edge computing systems. 2017 IEEE International Conference on Communications (ICC), 1-6. https://doi.org/10.1109/TWC.2017.2785305.

[3] Zhao, Z., Min, G., Gao, W., Wu, Y., Duan, H., & Ni, Q. (2018). Deploying Edge Computing Nodes for Large-Scale IoT: A Diversity Aware Approach. IEEE Internet of Things Journal, 5, 3606-3614. https://doi.org/10.1109/JIOT.2018.2823498.

[4] Liao, B., Ali, Y., Nazir, S., He, L., & Khan, H. U. (2020). Security analysis of IoT devices by using mobile computing: a systematic literature review. IEEE Access, 8, 120331-120350.

[5] "Railway Cybersecurity", European Union Agency for Cybersecurity (ENISA), 2020, ISBN: 978-92-9204-412-1, DOI: 10.2824/235164

[6] Potteiger, B., Zhang, Z., & Koutsoukos, X. (2020). Integrated moving target defense and control reconfiguration for securing Cyber-Physical systems. Microprocess. Microsystems, 73, 102954. https://doi.org/10.1016/j.micpro.2019.102954.

[7] "ENISA Threat landscape for supply chain attacks", European Union Agency for Cybersecurity (ENISA), 2021, ISBN: 978-92-9204-509-8 – DOI: 10.2824/168593

[8] Chuck Squatriglia, WIRED, "Polish Teen Hacks His City's Trams, Chaos Ensues"; https://www.wired.com/2008/01/polish-teen-hac/, accessed on 28.12.2023.

[9] Virvilis, N., Gritzalis, D., & Apostolopoulos, T. (2013). Trusted Computing vs. Advanced Persistent Threats: Can a Defender Win This Game?. 2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing, 396-403. https://doi.org/10.1109/UIC-ATC.2013.80.

[10] NIST's Computer Security Resource Center, Glossary: https://csrc.nist.gov/glossary, accessed 30.12.2023

[11] Y.3057: A trust index model for information and communication technology infrastructures and services.

[12] Køien GM (2011) Reflections on trust in devices: an informal survey of human trust in an internet- of-things context. Wirel Pers Commun 61(3):495–510

[13] Kurdi, H., Alfaries, A., Al-Anazi, A. et al. A lightweight trust management algorithm based on subjective logic for interconnected cloud computing environments. J Supercomput 75, 3534–3554 (2019). https://doi.org/10.1007/s11227-018-2669-y

[14] Selvaraj, A., & Sundararajan, S. (2017). Evidence-based trust evaluation system for cloud services using fuzzy logic. International Journal of Fuzzy Systems, 19, 329-337.

[15] Hm, Sanjay and Prakash, Guru, Trust Modeling and Service Level Agreement Monitoring in Federated Cloud (2021). JETIR July 2021, Volume 8, Issue 7, Available at SSRN: https://ssrn.com/abstract=3887385

[16] Muhtadi, Jalal & Alamri, Rawan & Khan, Farrukh & Saleem, Kashif. (2021). Subjective logic-based trust model for fog computing. Computer Communications. 178. 10.1016/j.comcom.2021.05.016.

[17] W. Gao, G. Zhang, W. Chen and Y. Li, "A Trust Model Based on Subjective Logic," 2009 Fourth International Conference on Internet Computing for Science and Engineering, Harbin, China, 2009, pp. 272-276, doi: 10.1109/ICICSE.2009.70.

[18] Fortino G, Messina F, Rosaci D, Sarné GM (2018) Using trust and local reputation for group formation in the cloud of things. Futur Gener Comput Syst 89:804–815

[19] Theo Dimitrakos, Tezcan Dilshener, Alexander Kravtsov, Antonio La Marra, Fabio Martinelli, Athanasios Rizos, Alessandro Rosetti, and Andrea Saracino. Trust aware continuous authorization for zero trust in consumer internet of things. In 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pages 1801–1812, 2020.

[20] Muhammad Sohail, Liangmin Wang, Shunrong Jiang, Samar Zaineldeen, and Rana Umair Ashraf. Multi-hop interpersonal trust assessment in vehicular ad hoc networks using threevalued subjective logic. IET Information Security, 13, 05 2019.

[21] Ali BA, Abdulsalam HM, AlGhemlas A (2018) Trust based scheme for iot enabled wireless sensor networks. Wireless Pers Commun 99(2):1061–1080

[22] Mendoza CV, Kleinschmidt JH (2018) A distributed trust management mechanism for the internet of things using a multi-service approach. Wirel Person Commun 103(3):2501–2513

[23] Pal S, Hitchens M, Varadharajan V (2019) Towards the design of a trust management framework for the internet of things. In: 13th International Conference on Sensing Technology (ICST), pp 1–7

[24] Bernabe JB, JlH Ramos, Gomez AFS (2016) TACIoT: Multidimensional Trust-aware Access Control system for the Internet of Things. Soft Comput 20(5):1763–1779

[25] Bica, I., Chifor, BC., Arseni, ŞC., Matei, I. (2020). Reputation-Based Security Framework for Internet of Things. In: Simion, E., Géraud-Stewart, R. (eds) Innovative Security Solutions for Information Technology and Communications. SecITC 2019. Lecture Notes in Computer Science(), vol 12001. Springer, Cham. https://doi.org/10.1007/978-3-030-41025-4_14

[26] Ruan Y, Durresi A, Alfantoukh L (2016) Trust management framework for internet of things. In: IEEE International Conference Advanced Information Networking and Applications (AINA), pp 1013–1019

[27] Sharma A, Pilli ES, Mazumdar AP, Govil M (2016) A framework to manage trust in internet of things. In: International Conference Emerging Trends in Communication Technologies (ETCT). IEEE, pp 1–5

[28] Bahutair M, Bougeuttaya A, Neiat AG (2019) Adaptive trust: usage-based trust in crowdsourced iot services. In: IEEE International Conference on Web Services (ICWS), pp 172–17

[29] Battah AA, Iraqi Y, Damiani E (2022) A trust and reputation system for iot service interactions. IEEE Trans Netw Serv Manage 19(3):2987–300

[30] Ferraris D, Fernandez-Gago C (2020) Trustapis: a trust requirements elicitation method for iot. Int J Inf Secur 19(1):111–127

[31] Dwarakanath R, Koldehofe B, Bharadwaj Y, Nguyen TAB, Eyers D, Steinmetz R (2017) Trustcep: adopting a trust-based approach for distributed complex event processing. In: 18th IEEE International Conference on Mobile Data Management (MDM), pp 30–39

[32] Hussain Y, Zhiqiu H, Akbar MA, Alsanad A, Alsanad AA-A, Nawaz A, Khan IA, Khan ZU (2020) Context-aware trust and reputation model for fog-based iot. IEEE Access 8:31622–31632

[33] Ursino D, Virgili L (2020) An approach to evaluate trust and reputation of things in a multi-iot scenario. Computing 102(10):2257–2298

[34] Bao, Fenye & Chen, Ing-Ray. (2012). Trust management for the internet of things and its application to service composition. 1-6. 10.1109/WoWMoM.2012.6263792.

[35] Refsdal, A., Solhaug, B., & Stølen, K. (2008). A UML-based Method for the Development of Policies to Support Trust Management., 33-49. https://doi.org/10.1007/978-0-387-09428-1_3.

[36] Drools: https://www.drools.org/, accessed on 30.12.2023.

[37] A., Kayssi, A., Chehab, A., & Artail, H. (2005). A comprehensive reputation-based trust model for distributed systems. Workshop of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks, 2005., 116-125. https://doi.org/10.1109/SECCMW.2005.1588304.

[38] J., Chen, J., Deng, H., & Wang, J. (2009). A Trust Model Combining Reputation and Credential. 2009 WASE International Conference on Information Engineering, 1, 635-638. https://doi.org/10.1109/ICIE.2009.159.

[39] Soto, J., Vizcaíno, A., Portillo-Rodríguez, J., & Piattini, M. (2009). An Agent System to Manage Knowledge in CoPs. Int. J. Cogn. Informatics Nat. Intell., 3, 75-94. https://doi.org/10.4018/jcini.2009010105.

[40] Marshall, R., Nguyen, T., & Bryant, S. (2005). A dynamic model of trust development and knowledge sharing in strategic alliances. Journal of General Management, 31, 41 - 57. https://doi.org/10.1177/030630700503100103.

[41] Ries, S., Habib, S.M., Mühlhäuser, M., Varadharajan, V. (2011). CertainLogic: A Logic for Modeling Trust and Uncertainty. In: McCune, J.M., Balacheff, B., Perrig, A., Sadeghi, AR., Sasse, A., Beres, Y. (eds) Trust and Trustworthy Computing. Trust 2011. Lecture Notes in Computer Science, vol 6740. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-21599-5_19

[42] Kumar, N. (2023). Hybrid Probabilistic Logic Programming: Inference and Learning. arXiv preprint arXiv:2302.00496.

[43] Aldini, A., Curzi, G., Graziani, P., Tagliaferri, M. (2021). Trust Evidence Logic. In: Vejnarová, J., Wilson, N. (eds) Symbolic and Quantitative Approaches to Reasoning with Uncertainty. ECSQARU 2021. Lecture Notes in Computer Science, vol 12897. Springer, Cham. https://doi.org/10.1007/978-3-030-86772-0_41

[44] Nafi, K.W., Kar, T.S., Hossain, A., & Hashem, M. (2013). An Advanced Certain Trust Model Using Fuzzy Logic and Probabilistic Logic theory. ArXiv, abs/1303.0459.

[45] Yuan, H., Zhao, P., Xian, X., Liu, G., Liu, Y., Sheng, V. S., & Zhao, L. (2023). Sequential Recommendation with Probabilistic Logical Reasoning. arXiv preprint arXiv:2304.11383.

[46] Haenni, R., Romeijn, J., Wheeler, G., & Williamson, J. (2010). Probabilistic Logics and Probabilistic Networks. https://doi.org/10.1007/978-94-007-0008-6.

[47] Fierens, D., Broeck, G., Renkens, J., Shterionov, D., Gutmann, B., Thon, I., Janssens, G., & Raedt, L. (2013). Inference and learning in probabilistic logic programs using weighted Boolean formulas. Theory and Practice of Logic Programming, 15, 358 - 401. https://doi.org/10.1017/S1471068414000076

[48] Ngo, L., & Haddawy, P. (1995). Probabilistic Logic Programming and Bayesian Networks, 286-300. https://doi.org/10.1007/3-540-60688-2_51.

[49] Jin, B., Wang, Y., Liu, Z., & Xue, J. (2011). A Trust Model Based on Cloud Model and Bayesian Networks. Procedia environmental sciences, 11, 452-459. https://doi.org/10.1016/J.PROENV.2011.12.072.

[50] Prabha, V., & Latha, P. (2017). Fuzzy Trust Protocol for Malicious Node Detection in Wireless Sensor Networks. Wireless Personal Communications, 94, 2549-2559. https://doi.org/10.1007/s11277-016-3666-1.

[51] Liao, H., Wang, Q., & Li, G. (2009). A Fuzzy Logic-Based Trust Model in Grid. 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing, 1, 608-614. https://doi.org/10.1109/NSWCTC.2009.218.

[52] Aljawharah Alnasser and Hongjian Sun. A fuzzy logic trust model for secure routing in smart

grid networks. IEEE access, 5:17896–17903, 2017.

[53] Shafer, Glenn. "Dempster-shafer theory." Encyclopedia of artificial intelligence 1 (1992): 330-331.

[54] Jøsang, Audun. Subjective logic. Vol. 3. Cham: Springer, 2016.

[55] Liang, Lifen (2008). Trust Derivation and Transitivity in a Recommendation Trust Model. 2008 International Conference on Computer Science and Software Engineering, 3, 770-773. https://doi.org/10.1109/CSSE.2008.484.

[56] Haydar, C., Roussanaly, A., & Boyer, A. (2013). Local Trust Versus Global Trust Networks in Subjective Logic. 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 1, 29-36. https://doi.org/10.1109/WI-IAT.2013.5.

[57] Jayasinghe, U., Lee, G., Um, T., & Shi, Q. (2019). Machine Learning Based Trust Computational Model for IoT Services. IEEE Transactions on Sustainable Computing, 4, 39-52. https://doi.org/10.1109/TSUSC.2018.2839623.

[58] Ma, W., Wang, X., Hu, M., & Zhou, Q. (2021). Machine Learning Empowered Trust Evaluation Method for IoT Devices. IEEE Access, 9, 65066-65077. https://doi.org/10.1109/ACCESS.2021.3076118.

[59] IEC 62443-1-1, Industrial communication networks – Network and system security – Part 1-1: Terminology, concepts and models (https://webstore.iec.ch/preview/info_iec62443-1-1%7Bed1.0%7Den.pdf) retrieved on 30.12.2023

[60] RFC 8520, Manufacturer Usage Description Specification, https://www.nccoe.nist.gov/mud-related-resources (accessed on 30.12.2023); https://datatracker.ietf.org/doc/html/rfc8520 (accessed on 30.12.2023).

[61] Cui, Ang, Michael Costello, and Salvatore Stolfo. "When firmware modifications attack: A case study of embedded exploitation." (2013).

[62] Wang, T., Wei, T., Gu, G., & Zou, W. (2010, May). TaintScope: A checksum-aware directed fuzzing tool for automatic software vulnerability detection. In 2010 IEEE Symposium on Security and Privacy (pp. 497-512). IEEE.

[63] Basnight, Zachry, et al. "Firmware modification attacks on programmable logic controllers." International Journal of Critical Infrastructure Protection 6.2 (2013): 76-84.

[64] Wazuh: https://wazuh.com/, accessed on 30.12.2023.

[65] Common Vulnerabilities and Exposures – MITRE: https://cve.mitre.org/, accessed on 30.12.2023.

[66] Cryptosense: https://github.com/cryptosense, accessed on 30.12.2023.

[67] LoRaWAN: https://lora-alliance.org/, accessed on 30.12.2023.

[68] Lashkari, Arash Habibi, Mir Mohammad Seyed Danesh, and Behrang Samadi. "A survey on wireless security protocols (WEP, WPA and WPA2/802.11 i)." 2009 2nd IEEE international conference on computer science and information technology. IEEE, 2009.

[69] Garbelini, M. E., Bedi, V., Chattopadhyay, S., Sun, S., & Kurniawan, E. (2022). {BrakTooth}: Causing Havoc on Bluetooth Link Manager via Directed Fuzzing. In 31st USENIX Security Symposium (USENIX Security 22) (pp. 1025-1042).

[70] Zigbee: https://csa-iot.org/all-solutions/zigbee/, accessed 30.12.2023

[71] Vidgren, N., Haataja, K., Patino-Andres, J. L., Ramirez-Sanchis, J. J., & Toivanen, P. (2013, January). Security threats in ZigBee-enabled systems: Vulnerability evaluation, practical experiments, countermeasures, and lessons learned. In 2013 46th Hawaii International Conference on System Sciences (pp. 5132-5138). IEEE.

[72] Oppliger, Rolf. SSL and TLS: Theory and Practice. Artech House, 2023.

[73] Siriwardena, P. (2014). OAuth 2.0. , 91-132. https://doi.org/10.1007/978-1-4302-6817-8_7.

[74] Satapathy, Ashutosh, and Jenila Livingston. "A Comprehensive Survey on SSL/TLS and their Vulnerabilities." International Journal of Computer Applications 153.5 (2016): 31-38.

[75] Kinney, Steven L. Trusted platform module basics: using TPM in embedded systems. Elsevier, 2006.

[76] Gao, Yansong, Said F. Al-Sarawi, and Derek Abbott. "Physical unclonable functions." Nature Electronics 3.2 (2020): 81-91.

[77] Costan, Victor, Ilia Lebedev, and Srinivas Devadas. "Secure processors part I: background, taxonomy for secure enclaves and Intel SGX architecture." Foundations and Trends® in Electronic Design Automation 11.1-2 (2017): 1-248.

[78] Zhen Ling, Huaiyu Yan, Xinhui Shao, Junzhou Luo, Yiling Xu, Bryan Pearson, Xinwen Fu, Secure boot, trusted boot and remote attestation for ARM TrustZone-based IoT Nodes, Journal of Systems Architecture, Volume 119, 2021, 102240, ISSN 1383-7621, https://doi.org/10.1016/j.sysarc.2021.102240.

[79] Yuriy Bulygin, Andrew Furtak and Oleksandr Bazhaniuk."A Tale of One Software Bypass of Windows 8 Secure Boot." Black Hat 2013

[80] Prometheus: https://prometheus.io/ accessed on 30.12.2023.

[81] New Relic: https://newrelic.com/ accessed on 30.12.2023.

[82] eBPF: https://ebpf.io/ accessed on 30.12.2023.

[83] Telegraf: https://www.influxdata.com/time-series-platform/telegraf/ accessed on 30.12.2023.

[84] Liao, Hung-Jen, et al. "Intrusion detection system: A comprehensive review." Journal of Network and Computer Applications 36.1 (2013): 16-24.

[85] Wireshark: https://www.wireshark.org/, accessed on 30.12.2023

[86] Snort: https://www.snort.org/, accessed on 30.12.2023.

[87] Suricata: https://suricata.io/, accessed on 30.12.2023.

[88] Prometheus: https://prometheus.io/ accessed on 30.12.2023.

[89] Graylog: https://graylog.org/, accessed on 30.12.2023.

[90] QRadar: https://www.ibm.com/qradar, accessed on 30.12.2023.

[91] OpenVAS: https://www.openvas.org/, accessed on 30.12.2023.

[92] Binwalk: https://github.com/ReFirmLabs/binwalk, accessed on 30.12.2023.

[93] Michail Bampatsikos, Ilias Politis, Christos Xenakis, and Stelios C. A. Thomopoulos. 2021. Solving the cold start problem in Trust Management in IoT. In Proceedings of the 16th International Conference on Availability, Reliability and Security (ARES '21). Association for Computing Machinery, New York, NY, USA, Article 128, 1–9. https://doi.org/10.1145/3465481.3469208