Trust-based model to secure internet of things network

Chaimaa Boudagdigue

THÈSE

Présentée à Avignon Université pour obtenir le diplôme de DOCTORAT

SPÉCIALITÉ : INFORMATIQUE

École Doctorale 536 “Agrosciences & Sciences”
Laboratoire Informatique d’Avignon

CedoC ST2I, ENSIAS
Université Mohammed V de Rabat

Modèles de confiance pour sécuriser les réseaux Internet des Objets

par
Chaimaa Boudagdigue

Soutenance le 15 mars, 2023 devant un jury composé de :

M.	Nicolas MONTAVONT	Professeur, IMT-Atlantique	Rapporteur
M.	Mohammed BOULMALF	Professeur, International University of Rabat, Maroc	Rapporteur
M^{me}	Valeria LOSCRI	Professeur, Inria Lille-Nord Europe, France	Rapportrice
M.	Yezekael HAYEL	Professeur, Laboratoire Informatique d’Avignon, France	Examinateur
M.	Olivier FESTOR	Professeur, Inria Lorraine, France	Examinateur
M.	Abderrahim BENSLIMANE	Professeur, Laboratoire Informatique d’Avignon, France	Directeur de thèse
M.	Abdellatif KOBBANE	Professeur, Université Mohammed V de Rabat, Maroc	Co-Directeur de thèse

Acknowledgements

I would like to express my sincere gratitude to all those who, in their own manner, have helped me in this long adventure.

I would like first of all to express my gratitude and my deep thanks to my thesis director Mr. Abderahim BENSLIMANE, Professor at the University of Avignon, for his support, his precious advice and his scientific contribution along these years of thesis.

I would also like to thank my supervisor, Mr. Abdellatif KOBBANE, Professor at Mohammed V university, for his pragmatic and efficient advice, which allowed me to progress and to achieve this work. I had the honor to have been one of his PhD students.

I would also like to express my gratitude and my deep respect to Prof. Valeria LOSCRI, Prof. Nicolas MONTAVONT and Prof. Mohammed BOULMALF who accepted to judge this work and to be the reporters. I would like to thank all the members of the jury for the great attention they have given to my work.

I can never thank enough my dear parents, my sister, my brother, and all my family for their support throughout this long term work. My dear husband, thank you for being here! No words can ever be enough to tell you all that I have in my heart, far beyond these pages. To all my family, thank you and i love you!

I will not finish without thanking all the staff of the Avignon computer laboratory and of the Mohammed V university for their welcome and their support throughout these years.

Résumé

L'internet des objets (IoT) est un nouveau paradigme dans lequel tout objet de la vie quotidienne peut faire partie de l'internet. L'objet doit simplement être équipé d'un microcontrôleur, d'un émetteur-récepteur et de piles de protocoles appropriées qui le rendent capable de communiquer. L'IoT rend les objets du quotidien intelligents et capables d'interagir de manière collaborative afin de fournir des services intelligents dans différents domaines tels que: l'agriculture, l'industrie, la santé et bien d'autres. Pour atteindre ces objectifs, les objets IoT sont amenés à gérer les données confidentielles et privées de leurs utilisateurs, ce qui les rend très vulnérables aux menaces de sécurité. Cependant, les objets IoT ne disposent pas des ressources nécessaires (énergie, mémoire, puissance de traitement, etc.) pour mettre en œuvre une sécurité robuste ou pour appliquer les mesures de sécurité traditionnelles basées sur les techniques de cryptographie habituellement déployées dans l'Internet traditionnel. En outre, les mesures de sécurité traditionnelles ne peuvent pas garantir la fiabilité des réseaux IoT, notamment en présence d'attaques internes.

Ainsi, notre travail consiste à proposer un modèle de gestion de la confiance dynamique où chaque objet IoT du réseau évalue le niveau de confiance de ses voisins à l'aide d'une métrique de confiance, avant d'interagir avec eux. Cela permet à l'objet de prédire le comportement futur de ses voisins et d'éviter les menaces de sécurité probables. La métrique de confiance d'un objet peut changer d'état en fonction de la coopération, de la réputation et de l'honnêteté de cet objet. Elle peut augmenter, diminuer, rester inchangée ou tendre vers zéro si l'objet est malveillant. Nous modélisons ces changements d'état en utilisant une chaîne de Markov à temps discret, une méthode mathématique efficace qui s'appuie sur l'état précédent d'un processus pour prédire son état futur. Deuxièmement, nous orientons nos recherches vers la gestion de la confiance dans les réseaux IoT industriels (IIoT). En effet, les réseaux IIoT désignent les dispositifs industriels (machines de production, robots, etc.) connectés à des réseaux sans fil, et qui collectent et partagent des données sur leurs environnements. L'IIoT rend les entreprises plus réactives, mais en même temps,

elle ouvre l'infrastructure de l'entreprise à des risques de sécurité.
Afin de remédier à ce problème, nous proposons trois contributions: Pour faciliter le processus de gestion de la confiance, la première contribution consiste à changer l'architecture traditionnelle des réseaux IIoT en une nouvelle architecture hiérarchique en proposant un nouveau concept basé les relations industrielles entre les dispositifs IIoT. La deuxième contribution propose un modèle de gestion de confiance dynamique, adapté aux exigences des environnements industriels. Et enfin, la troisième contribution consiste à évaluer la capacité et le dynamisme de notre modèle de confiance proposé pour détecter les changements de comportement des nœuds non confiants en utilisant le simulateur contiki/cooja.
Nous exploitons ensuite les résultats de ces contributions et la force de la théorie des jeux de signalisation pour proposer un mécanisme de révocation de certificats pour les réseaux IIoT. L'objectif principal de ce mécanisme est d'isoler efficacement et précisément les dispositifs IIoT non fiables pour qu'ils ne contribuent plus aux activités du réseau. Lorsqu'un certificat est révoqué, les autres nœuds du réseau doivent en être informés immédiatement. La question la plus importante est de savoir comment distribuer efficacement les informations de révocation de certificat parmi les dispositifs IIoT. Pour cela, nous proposons dans la dernière partie de cette thèse un nouveau schéma de vérification de certificats efficace, basé sur des certificats à courte durée de vie (SLC), et adapté aux exigences des réseaux IIoT. La période de validité de chaque SLC, dans le schéma proposé, est proportionnelle au niveau de confiance de son propriétaire. Cela permet de trouver un bon compromis entre la durée de vie du certificat et le trafic lié au renouvellement du certificat, tout en conservant un niveau de sécurité élevé. L'évaluation des performances que nous avons réalisée montre l'efficacité de notre schéma de vérification des certificats. En effet, le schéma proposé permet de réduire le temps nécessaire pour l'obtention des informations de révocation ainsi que les frais de stockage et de communication qui en résultent.

Mots-clés: *Confiance, Gestion de la confiance, Internet des objets, Internet industriel des objets, Theorie des jeux, Jeu de signalisation, Chaîne de Markov, Gestion des certificats numériques, Certificats à courte durée de vie.*

Abstract

The Internet of Things (IoT) is a new paradigm where any device of everyday life can become part of the Internet. The device just needs to be equipped with a microcontroller, a transceiver and appropriate protocol stacks that make it able to communicate. IoT makes everyday devices intelligent and able to interact in a collaborative way in order to provide intelligent services in different fields such as: agriculture, industry, healthcare and many others. To achieve these objectives, IoT devices must manage confidential and privacy-related data of their users, which makes them very vulnerable to security threats. However, IoT devices do not have the necessary resources (energy, memory, processing, etc.) to implement strong security or to apply the traditional security measures based on cryptographic techniques usually deployed in traditional Internet. Moreover, the traditional security measures cannot ensure the reliability of the IoT networks, especially in the presence of internal attacks. Hence, our work consists in proposing a dynamic analytical trust management model where each IoT device in the network evaluates the trust level of its neighbors using a trust metric, before interacting with them. This allows the device to predict the future behavior of its neighbors and avoid probable security threats. The trust metric of a device can change state depending on the cooperation, reputation and honesty of that device. It can increase, decrease, remain unchanged or tend to zero if the device is untrusted. We model these state changes by using a discrete-time Markov chain, an effective mathematical method that relies on the previous state of a process to predict its future state. Secondly, we focus our research on trust management in industrial IoT networks (IIoT). Indeed, IIoT networks refer to industrial devices (production machines, robots, etc.) connected to wireless networks, and which collect and share data on their environments. IIoT makes companies more reactive, but at the same time it opens the company's infrastructure to security risks. In order to address this issue, we propose three contributions: To facilitate the trust management process, the first contribution is to change the traditional architecture of IIoT networks into new hierarchical architecture by creating

a new concept called the industrial relationships between IIoT nodes. The second contribution proposes a dynamic trust management model, adapted to the requirements of industrial environments. And lastly, the third contribution is to evaluate the ability and the dynamism of our proposed trust management model to detect behavioral changes of malicious and selfish nodes, by using the contiki/cooja simulator. We subsequently exploit the results of these contributions and the strength of the Signaling game theory to propose a certificate revocation mechanism for IIoT networks. The main purpose of this mechanism is to effectively and accurately isolate malicious IIoT devices from further contributing to network activities. When a certificate is revoked, the other nodes in the network must be informed immediately. The most important issue is how to efficiently distribute certificate revocation information among IIoT devices? Therefore, we propose, in the last part of this thesis, a new efficient certificate verification scheme, based on short-lived certificates (SLC), and suitable for IIoT network requirements. The validity period of each SLC, in the proposed scheme, is proportional to the trust level of its owner. This makes a good trade-off between certificate life and overheads resulting from certificate renewal process, while keeping a high security level. The performance evaluation we conducted proves the effectiveness of our proposed certificate verification scheme to reduce the time needed to obtain the revocation information as well as the resulting storage and communication overhead to achieve this goal.

Keywords: *Trust, Trust Management, Internet of Things, Industrial Internet of Things, Game theory, Signaling game, Markov chain, Digital certificate management, Short-Lived Certificates.*

Contents

Acknowledgements i

Résumé iii

Abstract v

1 Introduction 1

2 Towards the notion of trust in communication and networking 7

2.1 Introduction 8

2.2 Towards the notion of trust in communication and networking . 8

2.2.1 Definition of trust in communication and networking: . . 8

2.2.2 Properties of Trust in communication and networking: . 10

2.2.3 Trust vs Reputation: 11

2.2.4 Trust vs Security: 12

2.3 Trust management in wireless networks 12

2.3.1 Trust Management modules: 13

2.3.2 Direct and indirect Trust management: 13

2.3.3 Trust-related attacks: 15

2.3.4 Applications of Trust management 16

2.3.5 Categories of Trust Management Models: 21

2.4 Conclusion 23

3 Trust Modelling methods 25

3.1 Introduction 26

3.2 Basic Methods 26

3.2.1 Weight function 26

3.2.2 Relational measurement 27

3.2.3 Fuzzy Logic 29

3.2.4 Game Theory 30

3.2.5 Entropy Theory . . . 31
3.3 Bayesian Methods . . . 33
3.3.1 Bayesian interference model . . . 33
3.3.2 Dempster-Shafer (DS) theory . . . 34
3.4 Graph methods . . . 36
3.4.1 Network features . . . 36
3.4.2 Semiring . . . 38
3.5 Machine learning methods . . . 40
3.5.1 Support Vector Machine (SVM) . . . 40
3.5.2 K-Means Clustering . . . 42
3.6 Conclusion . . . 44

4 Trust Management in IoT networks **45**
4.1 Introduction . . . 45
4.2 Requirements and constraints of IoT networks . . . 47
4.3 Related work . . . 48
4.4 The proposed trust management model analysis . . . 53
4.4.1 Trust management Model Overview . . . 53
4.4.2 Trust parameters . . . 54
4.4.3 Pseudo code of the estimation algorithm . . . 58
4.4.4 Probabilities of transition from one state to another . . . 58
4.5 Performance evaluation . . . 60
4.6 Conclusion . . . 64

5 Trust Management in IIoT networks **65**
5.1 Introduction . . . 66
5.2 Related work . . . 67
5.3 Future automotive factory architecture based on IIoT . . . 70
5.4 The proposed trust management model analysis . . . 72
5.4.1 Phase I: Designation process of CL . . . 73
5.4.2 Phase II: The community establishment method . . . 77
5.4.3 Phase III: The proposed trust management model . . . 80
5.5 Performance evaluation . . . 83
5.5.1 Part 1: Comparative energy study . . . 84
5.5.2 Part 2: The sensitivity, the responsiveness and the resiliency of our proposed trust management model to trust-related attacks launched by malicious IIoT nodes. . . . 89

5.6 Conclusion 97

6 Certificate revocation in IIoT networks using Signaling game 99

6.1 Introduction 100

6.2 Related certificate revocation strategies 101

6.3 Stage Certificate Revocation Game 103

6.4 Equilibria of the stage Certificate Revocation Game 106

6.4.1 Pure-strategy BNE of the stage Certificate Revocation Game 107

6.4.2 Mixed-strategy BNE of the stage Certificate Revocation Game 109

6.5 Multi-stage dynamic Certificate Revocation Game 111

6.6 The PBE of the Certificate Revocation Game in mixed-strategy 113

6.7 The proposed certificate revocation mechanism based on the PBE of the multi-stage Game 115

6.8 Performance evaluation of the certificate revocation mechanism 116

6.9 Conclusion 121

7 Trust-based Certificate Management for IIoT networks 123

7.1 Introduction 124

7.2 Related work 125

7.3 Proposed architecture 127

7.4 SLC-based certificate verification scheme 129

7.4.1 SLC as an alternative to the conventional certificate verification schemes 130

7.4.2 Workflow of our Certificate verification scheme 132

7.5 Performance Evaluation 132

7.5.1 Performance evaluation of the SLC management process 133

7.5.2 Performance evaluation of the SLC-based certificate verification scheme 136

7.6 Security Analysis 139

7.7 Conclusion 141

8 Conclusion and perspectives 147

List of Figures 154

List of Tables 155

Bibliography **157**

Personal publications 157

References 157

Chapter 1

Introduction

1.1 Context

In the last decade, Internet of Things (IoT) approached our lives silently and gradually due to the availability of wireless communication systems (e.g., Radio-frequency identification, WiFi, 4G, IEEE 802.15.x), which have been increasingly employed as technology driver for crucial smart monitoring and control applications.

IoT finds application in many different areas, such as home automation, industrial automation, smart cities, healthcare, and many others as shown in Fig. 1.1. For example, in agriculture, IoT allows a permanent monitoring of the state of the soil, the rate of humidity, the rate of mineral salts, etc. This information is transmitted to the farmer in real time to take the necessary measures and ensure proper production. The healthcare sector has also seen a huge number of IoT-based applications that monitor the physical activity of patients and eventually inform their doctors if irregular and abnormal signs are detected. In smart cities, IoT enables the collection of a broad spectrum of demographic data (e.g., carbon footprint, noise level, etc.) to prevent excessive emission of CO_2, better organize available resources (e.g., smart waste management), and better deploy human resources at the disposal of the city.

According to the analysis conducted in 2020 by IoT ANALYTICS [1], a leading provider of strategic information to the IoT market, on the main application areas of IoT, 22 % of the IoT projects of public companies identified concern the Manufacturing/Industrial area. Followed by the transportation area (15 %) and the energy area (14 %). The integration of IoT in the industrial area offers a potential economic impact of $1.2 trillion to $3.7 trillion a year in

[1] https://iot-analytics.com/top-10-iot-applications-in-2020/

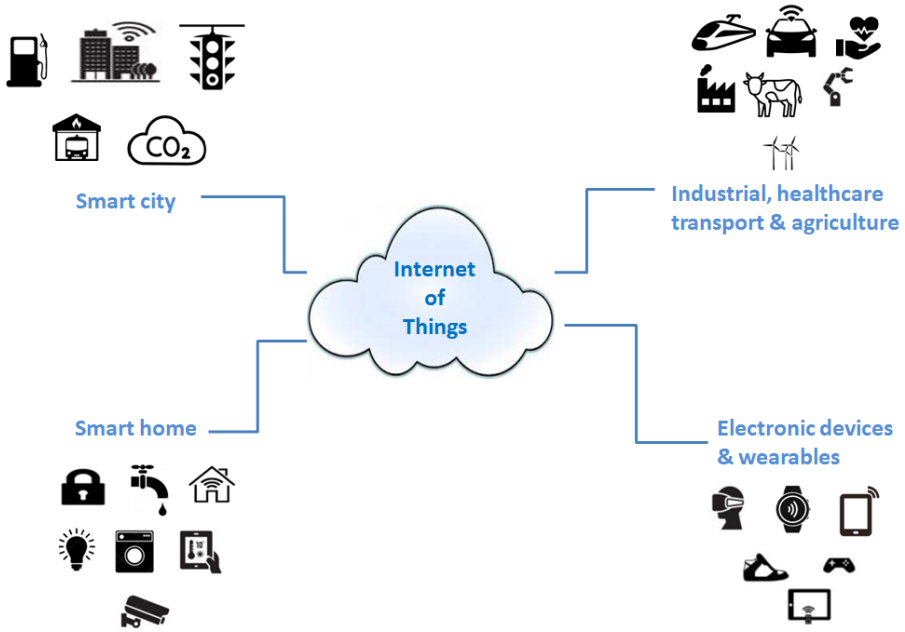


Figure 1.1: Applications of IoT

2025 [2]. Indeed, the application of IoT technology in an industrial setting and its integration into industrial processes is designated by a new concept called Industrial Internet of Things (IIoT). IIoT technology is a fundamental element of the transition to Industry 4.0, it brings many benefits to manufacturers such as smart production, predictive maintenance and remote control.

1.2 Problematic and objectives of the thesis

Since the emergence of the IoT in the late 1990s, security experts have always warned about the potential risks to connect a large number of things, often not sufficiently secured, to the Internet. Several reasons may explain why IoT is highly vulnerable to cyberattacks. First, IoT devices manage sensitive information related to the privacy of their users. Second, many manufacturers of IoT devices have not considered security as a top priority in the conception phase of their products. Third, the high level of heterogeneity, coupled with the large scale of IoT systems and the shared nature of the medium in wireless networks, makes IoT networks highly vulnerable to cyberattacks. Finally, the most important reason is that IoT devices do not have the necessary resources (energy, memory, processing) to implement strong security or to apply the traditional security mechanisms based on cryptographic techniques usually deployed in traditional Internet as explained in [5].

Several researches consider trust management as a potential solution to security issues related to IoT. Indeed, trust management can improve the

[2]https://research.aimultiple.com/iot-applications/

security of IoT networks by using the notion of belief and by continuously analyzing the behavior of devices in the network. Since they do not rely on cryptographic techniques that are very costly in terms of resources, security mechanisms based on trust management are well adapted to the constraints of IoT networks. They detect accurately internal attacks which traditional security mechanisms are unable to detect. A thorough analysis of the literature reveals that a large majority of trust management models proposed for IoT networks are not resilient to trust distortion attacks, which weakens the security of IoT networks instead of strengthening it. Also, these trust management models did not consider the inherent characteristics of IoT networks, such as limited resources, heterogeneity, context awareness, etc. Thus, by highlighting research gaps and open challenges, our work consists in proposing an effective distributed trust management model considering the requirements and constraints of IoT networks. The proposed trust management model uses a Markov Chain to formalize the variation of trust metrics of IoT devices according to their dynamic behaviors in the network.

Currently, many companies rely on IIoT to support the Industrial Revolution 4.0. For this purpose, these companies deploy IIoT devices in all their industrial services and processes, which make these devices increasingly attractive targets for security threats. Indeed, IIoT networks require continuous monitoring to detect unwanted behaviors conducted by compromised and malicious devices. To deal with this issue, our research has been oriented towards the trust management in IIoT networks. In this context, we define a new concept called industrial relationships where IIoT devices are able to establish industrial relations between them. We use these industrial relationships to change the traditional architecture of IIoT networks into new hierarchical architecture constituted of clusters called industrial communities. Each community is monitored by a leader who evaluates the trust of member nodes and supports the trust management process in the IIoT network. We emphasize that the leader is a specific node with high trust and sufficient resources to perform monitoring tasks. The trust is evaluated according to three parameters namely, cooperation, honesty related to the direct interactions and honesty related to the indirect interactions. These parameters are used later to propose a dynamic trust management model suitable for IIoT networks. To the best of our knowledge, at the time we started our research on trust management in IIoT, there was no proposed trust management model specifically for IIoT networks that considers the requirements and constraints of these networks.

According to an analysis provided by the cybersecurity company Barracuda

[3], 94% of industrial organizations have experienced a "security incident" in the past 12 months. The majority of these attacks are conducted by malicious IIoT devices that have successfully passed the authentication phase and are part of the network. We notice that the main purpose of malicious IIoT devices is to disclose data related to the industrial processes and damage the basic functionalities of the network. Hence, in the proposed hierarchical architecture, the leader of each community uses the result of the monitoring process to manage the certificates of the member nodes belonging to its community. The leader renews the certificates of legitimate member nodes after expiration and revokes the certificate of malicious nodes to prevent their owner from launching internal attacks in the network. However, the type of member nodes (legitimate or malicious) is hidden information for the leader. Therefore, the leader lets the member node act first in order to observe its behavior and act accordingly. For this purpose, we model the interactions between a member IIoT node and its leader by using signaling game theory, in which players can use the actions of their adversary to make deductions about hidden information. The leader can obtain its best response strategy based on its belief on the member IIoT node, updated by using the Bayesian rules. We subsequently design a certificate revocation mechanism for IIoT networks that outperforms other existing mechanisms in the literature in terms of revocation rate and revocation time, even in the presence of a large percentage of malicious devices in the network.

When a malicious device is revoked, it is crucial to quickly notify other IIoT devices in the network to avoid any transaction with it. Indeed, the information about revoked certificates allows IIoT devices to verify the status of their peers' certificates before establishing a connection with them. However, the distribution of this information in the IIoT network in a resource efficient manner and without latency is a difficult task. Conventional schemes for distribution of information about revoked certificates including certificate revocation list (CRL), online certificate status protocol (OCSP) and Bloom filter are not appropriate to be applicable in IIoT networks because they are subject to some issues like communication overhead, storage overhead and latency. Hence, as alternative to these conventional schemes, we propose a new efficient certificate verification scheme, based on short-lived certificates (SLCs) and suitable for IIoT network requirements. SLC operates as an ordinary certificate, except that its validity period is a short span of time which can be few hours or few days. With this shortened validity period, the certificates of mali-

[3]https://blog.barracuda.com/2022/07/12/report-the-state-of-industrial-security-in-2022/

cious devices would expire before their status could be verified by conventional schemes, and before a major attack could be properly conducted. The SLCs in our proposal do not have the same validity period. In fact, the validity period of each SLC is proportional to the trust level of its owner. We assign a larger validity period to the legitimate devices and a shorter to the less trusted ones. This makes a good trade-off between certificate life and overheads resulting from certificate renewal process, while keeping a high security level.

1.3 Organization of the manuscript

This manuscript is divided into eight chapters. In chapter 2, we give a detailed introduction of the trust concept in the field of communication and networking, based on definitions from several disciplines including sociology, economics, philosophy and many others. Afterwards, we present the different modules, components, and related attacks that can be considered during the trust management process. We present also the significant applications of trust management identified for wireless networks.

In chapter 3, we provide an extensive analysis of the most widely used trust modeling methods in wireless networks. Indeed, we consider four classes of trust modeling methods including, Basic Methods, Bayesian Methods, Graph methods and Machine learning methods. We define the mathematical theories available for each method and how they can be applied to modeling trust. The main purpose of this chapter is to provide the readers a clear vision towards the design of an effective trust management model.

In chapter 4, we propose a distributed trust management model for IoT networks [1] where each node monitors its neighbor and assigns it a local trust metric according to its behavior in the network. The proposed trust management model considers not only the variation of the trust metric according to the dynamic behavior of the devices, but also the constraints related to the monitoring process in wireless environments.

In chapter 5, we propose three contributions: In the first contribution, we use the industrial relationships between devices to construct a new hierarchical architecture for IIoT networks instead of their traditional centralized architecture [2]. The purpose of the proposed architecture is to strengthen the trust management process in IIoT networks. In the second contribution, we propose a dynamic trust management model, adapted to the requirements of industrial environments. In the third contribution, we use the simulator contiki/cooja to evaluate the performance of the proposed hierarchical architecture and the trust management model proposed for the IIoT network.

In chapter 6, we rely on the properties of signaling game theory to design a certificate revocation mechanism as well as its corresponding algorithm [3]. The proposed revocation mechanism improves the security of the IIoT network, it deals with malicious devices and detects accurately the behavior changes of devices that exhibit honest behavior up to achieving a high level of trust and then behave in an untrusted manner.

In chapter 7, we improve the hierarchical architecture proposed in chapter 5 to support the management of certificates in IIoT networks, by using the blockchain concept. We also propose an efficient certificate verification scheme for IoT network, based on SLC [4]. In this chapter, we study also the time needed to obtain the revocation information as well as the resulting storage and communication overhead to achieve this goal in our proposed scheme. We choose these two parameters because they are very important for the validation of any certificate verification scheme.

In chapter 8, we close this document by evoking a summary of our work as well as the perspectives concerning the various works realized.

Chapter 2

Towards the notion of trust in communication and networking

Contents

2.1 Introduction **8**

2.2 Towards the notion of trust in communication and networking **8**

2.2.1 Definition of trust in communication and networking: . . 8

2.2.2 Properties of Trust in communication and networking: . 10

2.2.3 Trust vs Reputation: 11

2.2.4 Trust vs Security: 12

2.3 Trust management in wireless networks **12**

2.3.1 Trust Management modules: 13

2.3.2 Direct and indirect Trust management: 13

2.3.3 Trust-related attacks: 15

2.3.4 Applications of Trust management 16

2.3.5 Categories of Trust Management Models: 21

2.4 Conclusion **23**

2.1 Introduction

The significant advances in wireless networks including IoT, Mobile Ad-hoc Networks (MANETs), Vehicular ad-hoc networks (VANETs), etc., have a direct and positive impact on our daily lives. But to have a wide acceptance by the large public, these networks must meet high security requirements because they manage sensitive data related to the daily and private lives of their users. Indeed, it is important to ensure that every cooperating entity in the network is honest and does not represent a failure point for the entire network.
Traditional security measures proposed for the Internet do not ensure the reliability of the wireless networks, especially in the presence of internal attacks. These measures must be reinforced by trust management. Indeed, trust management introduces a general perception of the risk and uncertainty related to entities to maintain their legitimacy. It also allows continuous analysis and monitoring of the behavior of entities based on the notion of belief.
This chapter is organized as follows. To understand the proposed trust management approaches in the literature for wireless networks, we start by providing a comprehensive definition of the concept of trust and introduce its main properties in section 2.2. We also provide a clear distinction between trust, reputation and security to differentiate them, as these three concepts are often used interchangeably. In section 2.3, we describe the different trust management modules, trust management components, and related attacks that can be considered during the trust management process. Thereafter, we present the significant applications of trust management, identified for wireless networks. These applications are developed for specific purposes; they allow to enhance the security of routing, composition and management of trusted services, collaboration, intrusion detection, access control and authentication in wireless networks.

2.2 Towards the notion of trust in communication and networking

2.2.1 Definition of trust in communication and networking:

According to Cambridge dictionary [1], trust is "to believe that someone is good and honest and will not harm you, or that something is safe and reliable [6]". At first glance, it seems that trust is easy to define and understand, as we experience and use it in our daily life. However, in communications and net-

[1] https://dictionary.cambridge.org/dictionary/english/trust

working field there is no single definition of trust, each research paper defines it according to its use, which makes the definition of trust very fuzzy and confusing.
Trust is a multidisciplinary subject [7], it belongs to many disciplines such as: sociology, economics, philosophy and many others. Each of these disciplines has produced its own definition and concept of trust. Therefore, these definitions can be exploited to better define and model the trust in communications and networking field. [2]

In sociology : Trust is considered as a subjective probability [7] that allows an individual to believe that the other will accomplish what he expects of him, knowing that the individual is unaware of the other's future actions and has only expectations. Thus, in sociology, trust involves an inevitable element of risk and potential doubt.

In economics : Trust is also closely linked to the economic activity since the economic exchanges cannot take place if there is too much mistrust between economic agents. We don't exchange anything without trusting the value of what we get in return, just as we don't accept to work for someone if we suspect that he won't pay us. Hence, Trust in economy is based on the assumption that human is rational [7], it aims to maximize its gain, which sometimes turns into selfish behavior when the only goal becomes to serve its own interests.

In philosophy : According to Stanford's Encyclopedia of Philosophy [3], trust is depending on the other but at the same time being vulnerable to the other, especially to treason. It is a relationship involving three parts: the trustor (the individual trusting [8], also called the monitor), the trustee (the individual being trusted [8], also called the monitored) and the context. We rarely trust people entirely (i.e., Alice simply trusts Bob), but rather, Alice trusts Bob to do task D [9] or Alice trusts Bob in domain C [10]. This leads us to say that trust between two individuals is based on the moral relationship and depends on the context.

In psychology : Trust emphasizes the cognitive process centered on individual reflection by which human beings learn trust from their past experiences. Indeed, past experiences affect significantly the ability to trust in the future.

[2] https://blog.barracuda.com/2022/07/12/report-the-state-of-industrial-security-in-2022/
[3] https://plato.stanford.edu/entries/trust/

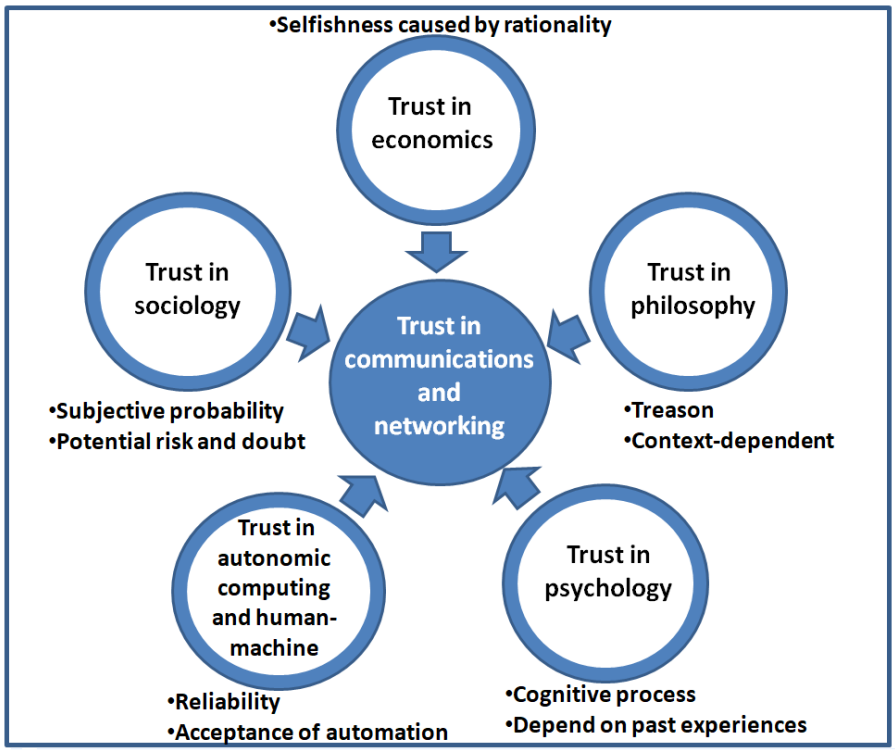


Figure 2.1: The concept of trust defined by the other disciplines can be applied to modeling trust in communications and networking [7]

In autonomic computing and human-machine interactions : The complexity of the technology makes trust in autonomic computing and human-machine interaction increasingly critical, especially when the responses of automated systems can be unexpected. Several works in autonomic computing and human-machine interactions field link trust with reliability where operators will use an automated system only if they believe it is reliable and trustworthy, so the trust specifies the amount of acceptance and use of automation.

Based on the definitions of trust from different disciplines, reviewed above and summarized in Fig. 2.1, we define trust in communications and networking as the degree of subjective belief towards the behavior of a given node. It describes the expectations of a trustor node towards a trustee node, in terms of honesty, quality of service, reliability, integrity, capacity and availability of the activity, and future behavior. Trust depends on context and past experiences, it always involves potential risk and doubt caused by treason and selfish behavior of nodes.

2.2.2 Properties of Trust in communication and networking:

There are several properties that need to be considered when evaluating the trust in communication and networking:

- Subjectivity: Trust depends on the evaluation criteria of the trustor node resulting from its opinions and experiences. Two trustors may have different opinions about the same trustee, each influenced by its own beliefs.

- Transitivity: Trust is not transitive, i.e. although node A trusts node B and node B trusts node C, this does not mean that node A will trust node C. However, sometimes to evaluate the trust, the trustor uses the recommendations received from other nodes in the network. In such a case, the trust is assumed to be transitive. Thus, for this assumption to be acceptable, the trust must be evaluated in the same context.

- Dynamism: In wireless networks, information change rapidly over time due to the dynamic topology of networks and the dynamic behavior of nodes. For this purpose, the measurement of trust must also be dynamic in order to be current.

- Asymmetry: Trust is considered asymmetric. Indeed, even if node A trusts node B, this does not involve that node B also trusts node A.

- Composite: To evaluate trust, one or more metrics can be considered for this purpose. We classify trust metrics (also called trust parameters or trust factors) into two categories, social trust metrics and Quality of Service (QoS) trust metrics [11]. The social trust metrics rely on the social relationships, community interest and honesty of a node to assess its level of selfishness and lack of cooperation and competence. Whereas, the QoS trust metrics assess the reliability of the trustee node according to its capacity, its successful transactions, its resources consumed, its execution time, and the quality of its peer-to-peer communications.

- Context-dependent: The definition and calculation of the trust depends on the context of its use [12]. For example, node A cannot simply trust node B, but node A trusts the node B in a well-defined context. Hence, the properties and the composition of trust change with the change of the context. As well, the selection of the trust modeling method depends on the application context of trust.

2.2.3 Trust vs Reputation:

Trust and reputation are two words that are often confused because many academic papers use them in an interchangeably manner. A given node is trusted if its behavior corresponds always to what the system expects from an

ideal node, whereas an untrustworthy node will frequently deviate from the system's expectations. Hence, trust allows a trustor to accurately predict the behavior of the trustee in the network. In contrast to trust, reputation is not a prediction of the future, but a knowledge of the past. It is a collectively agreed version of how history unfolded. The reputation of the trustee reflects its history, the way it has behaved in all its previous transactions. The trustor is more susceptible to predict the future performance of the trustee if it has information about the past and history of the trustee, reflected by its reputation. Therefore, strong reputation builds strong trust [13].

2.2.4 Trust vs Security:

Security consists of protecting the wireless networks against any violation, intrusion, degradation or theft of data. A good security approach must offer high quality services in terms of availability, confidentiality, authentication, integrity and non-repudiation [14]. Users trust a system if they have guarantees that the system is properly secured. At the same time, the trust allows to strengthen the security of distributed systems. In fact, security and trust each influence the other. The security mechanisms usually deployed in traditional Internet, also called hard security mechanisms, are not suitable to be reused and applicable in wireless networks, because they do not meet the requirements of these networks. Indeed, these mechanisms cannot protect against internal attacks and can not detect the change of behavior of the nodes. Also, they are expensive in terms of computational resources for nodes with limited resources as they are based on pre-existing shared secret and on cryptographic techniques. To address these limitations, traditional hard security mechanisms can be replaced by trust to enhance security and privacy in wireless networks. Trust can be seen as a soft security mechanism.

2.3 Trust management in wireless networks

The development of strategies [15] for estimating and maintaining trust in different entities or systems is called trust management. Trust management is mainly important in wireless networks, whose functionality relies on the collaborative behavior of network nodes. Trust management enhances the traditional security measures of wireless networks by relying on the notion of belief, it also ensures that only legitimate nodes participate in network activities.

2.3.1 Trust Management modules:

According to [16], trust management consists of four main modules: trust composition, trust propagation, trust aggregation and trust update. Trust composition allows the collection of knowledge and information about the behavior of a trustee node. The collection can be done locally by the trustor through direct observation, or remotely through recommendations from neighboring nodes. Trust aggregation allows to combine the collected behavioral observations obtained through trust composition to get a unique trust value for each entity. The Trust aggregation is achieved by using modelling methods such as Basic methods, Bayesian methods, Graph methods, etc. The widely used methods for trust modeling are detailed in chapter 3. Trust propagation defines the method of propagating observations and trust values between entities. Propagation is centralized if it needs a centralized entity and structures like Distributed Hash Table. When the propagation is done through encounters between entities or during interactions between them, it is considered distributed. Trust update is Event-Driven when the update of trust values depends on an event such as an encounter or transaction between entities, and, it is Time-Driven when the update of the trust values takes place in defined time periods.
We illustrate, in Fig. 2.2, the relationships and interactions between the four trust management modules.

2.3.2 Direct and indirect Trust management:

The trust computation process is achieved either based on the direct trust, or based on the indirect trust, or by combining these two components as follows:

Direct Trust : reflects the individual opinion of a trustor towards a trustee [17], in a well defined context, without the intervention and involvement of a third party. This opinion is derived from past experience based on direct observation and interactions between the trustor and the trustee. It provides the trustor an overview of the future intentions and behavior of the trustee. Indeed, if past interactions with a trustee are successful, the trustor assigns a high trust value to that trustee, which will increase the chances of future interactions between them. However, if the trustee is known to be deficient in previous interactions (e.g. packet drop, packet flood leading to excessive consumption of resources in the network, denial of service attacks, etc.), the trustor will assign to it a low trust value to indicate that it does not wish to deal with this trustee in the future. The direct trust can also be deduced from

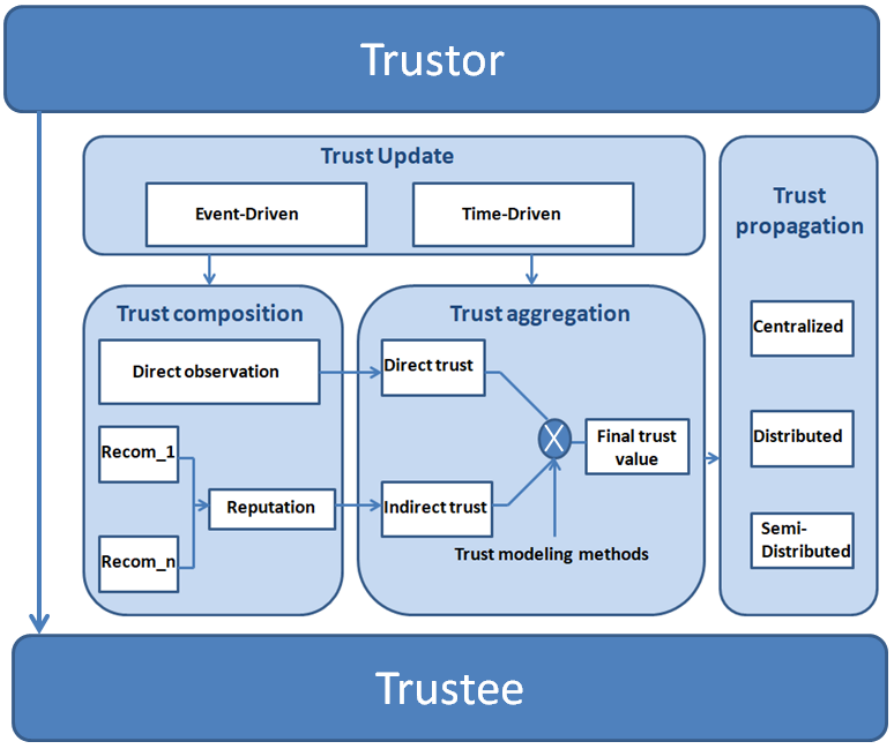


Figure 2.2: Trust management modules.

the knowledge that refers to the features of the trustee such as its community, its capacity, its direct relationships, its profile, etc. Authors of [18] estimated the direct trust of devices by considering their social attributes (centrality, Community of interest, friendship, etc.). Indeed, according to [18], a trustor can trust a trustee if they are directly linked by one or several social relations or they have common interests between them. Authors of [19] considered the features of the devices such as their computational capabilities, when evaluating their direct trust. It is important to know that the choice of features needed for knowledge inference differs from one work to another in the literature. This choice depends mainly on the context of the trust application.
Direct trust is the best source of decision-making support. However, in some cases, the trustor is led to interact with a trustee with which it has neither direct past experience nor sufficient knowledge. Therefore, the trustor must take into account an additional perspective to evaluate the trust level of the trustee, called indirect trust.

Indirect Trust : Indirect trust, also called indirect observation, relies on opinions propagated to the trustor from different nodes in the network having past experiences with the target trustee. These opinions are called recommendations, and the aggregation of all collected recommendations constitutes the global reputation of the trustee. The global reputation will be considered as a

general belief on the trustee in the network. However, recommenders can be dishonest, they may give false recommendations to the trustor to bias the final trust calculation of a target trustee [20]. For example, dishonest recommender can give a good recommendation to a malicious trustee to improve its trust score and to prevent it from being revoked from the network, also it can give a bad recommendation to a legitimate trustee to decrease its trust score and prevent it from being selected as a service provider in the future. To deal with false recommendations, several works consider the trustworthiness of the recommenders [19] and the quality of the collected recommendations [21] in their calculation of the indirect trust.

The majority of research in the literature such as [22], [23], [24], [25] and [26] combined the two components of trust, the direct and the indirect trust, in order to enhance the accuracy and performance of their trust calculations.

2.3.3 Trust-related attacks:

Trust management is used to enhance the security of wireless networks by monitoring continuously the behavior of nodes in these networks. However, this process can be disrupted by malicious nodes that may conduct trust-related attacks whose goal is to distort trust assessments and degrade trust management performance. To take full advantage of the security-enhancing contributions of trust to wireless networks, trust management in these networks must be resilient to the following attacks:

- Bad-mouthing attack: Malicious node can give a wrong bad recommendation to a well-behaved node to decrease its trust value and therefore revoke it from the network or decrease its chances of being chosen as a service provider [123].
- Ballot stuffing attack: Malicious node can give a good recommendation to another misbehaved node to increase its trust value and to allow it to continue its malicious activities in the network [123].
- Self-promoting attack: Occur when a malicious node promotes itself by good recommendations to increase its trust value and avoid revocation.
- Honesty attacks: Occur when a malicious node gains the trust of the system by exhibiting correct behavior for non-critical transactions and changes its behavior for critical and important ones.
- Selfish attacks: Selfish nodes do not carry out attacks in the network; their objective is not to damage the basic functionality of the network

but to conserve their resources to serve their own interests.

- Coalition attacks: Happen when a cluster of devices mobilizes to perform a Bad-mouthing attack against an honest node, or a Ballot stuffing attack to promote a malicious node.
- Whitewashing attacks: Occur when a misbehaving node leaves and rejoins the network each time after a series of untrusted behaviors, to wash away its bad reputation. Afterwards, the node recovers its untrusted behavior as soon as it recovers its trust value.
- On-off attacks: Occur when a malicious node behaves honestly for a period of time to gain the trust of other nodes in the network and then changes its honest behavior to untrusted behavior. The node recovers its honest behavior once the network detects its untrusted behavior, and so on.
- Discriminatory attack: This type of attack is very frequent in social Internet of Things (SIoT) [27] where IoT merges with Social Networks. Thus, devices may inherit human nature and favor devices with which they have more common friends or experiences and behave selfishly with other devices.

2.3.4 Applications of Trust management

The integration of trust management in wireless networks allows to reach various security and confidentiality objectives required to maintain reliable functionality of these networks. The significant applications of trust management, identified for wireless networks, are the following:

Secure Routing In wireless networks, when a node wants to transmit a packet to other nodes that are outside its transmission range, the packet must be forwarded through one or more intermediate nodes without any modification or violation of the information contained in this packet. Routing methods based on routing tables, used in traditional networks, are not appropriate for application in wireless networks because the topology of these networks is dynamic, it can change randomly at unpredictable times.

Also, routing protocols based on pre-existing shared secret and on cryptographic techniques [28], [29], [30] are not recommended to be applied in wireless networks because the distributed and dynamic nature of these environments does not allow to have prerequisites and because cryptographic

techniques are considered computationally expensive on resource-constrained nodes. For all these reasons, several researches in the literature proposed routing protocols based on trust. Indeed, Trust Management allows selecting the reliable path between a source and a destination by calculating the trust value of each forwarding node. Conventional routing protocols as Routing Protocol for Low-Power and Lossy Networks (RPL) proposed for IoT [31], Dynamic Source Routing (DSR) [32] and Ad-hoc On Demand Distance Vector protocol (AODV) proposed for distributed wireless networks do not have routing disruption prevention mechanisms. Therefore, several works such as [33], [34], [35], [36], [31], [24] and [37] have integrated the trust into these protocols in order to strengthen their performance and help them to establish reliable paths based only on legitimate nodes. The established paths do not include selfish and malicious nodes that can disrupt the packet routing process.

Reliable Service Composition and Management In a service-oriented network like IoT, devices process the collected data to provide users with intelligent services and advanced applications such as health monitoring, traffic detection, meteorological reporting, etc. Effective use of these services requires a secure system in which users can trust the service providers. In IoT, nodes can be both providers and requesters of services, in both cases the nodes are able at any time to launch attacks into the network. Hence, choosing a reliable service provider is a serious concern for service Composition. Random service composition, in which a node randomly selects service providers, misleads the decision-making of that node and imperils its security. To this end, several works such as [38], [39], [40] and [41] have chosen to integrate trust management into their service composition applications to allow nodes to select service providers based on their trust levels. Authors of [38] provided an adaptive trust management model to support reliable service composition applications in IoT networks. In the proposed model, each device evaluates the trust value of other devices based on the history of its experiences with them and based on the trust feedback of devices sharing social interests with it. The calculated trust value could be an indicator and predictor of a provider's service quality. Hence, only devices with a high level of trust can be chosen as service providers. Authors of [39] proposed a trust management model based on guarantor and reputation, in which devices use credits to obtain services. Indeed, if a device gets a correct service then the device must pay the service provider some credits as a commission. However, if the service received is malicious and incorrect, then the service provider must pay credits to the device requesting the service as a forfeit. The commission and forfeit rates serve as

a guarantee for the behavior of a node. The node requesting the service sends feedback to the reputation server to describe its satisfaction towards the provided service. The received feedback is used then by the reputation server to update the reputation of the service provider.

Collaboration In Ad-hoc networks, several tasks such as decision-making, key-distribution, service composition, forwarding packets are generally decentralized, and rely on the cooperative participation of all nodes in the network. For example, if a node wants to transmit a packet to a destination that is outside its transmission range, the node needs the collaboration of several intermediate nodes. Nodes can also collaborate to remove a suspicious node from the network by sending accusations against that node whenever it behaves maliciously [42]. The success of these cooperation-based tasks involves complex aspects such as the evaluation of the cooperativeness and honesty of the participating nodes. Indeed, the participating nodes must be carefully selected to avoid those that are selfish and malicious. Trust management provides continuous monitoring of network nodes to predict their future behavior in terms of honesty, cooperativeness, quality of service, integrity and availability of the activity. For this reason several works as [43], [34], [38], [25] and [44] have relied on trust to support cooperation-based tasks and maintain cooperation between nodes. Authors of [43] proposed a trust management model based on fuzzy logic to enhance collaboration and decision making among IoT devices. In the proposed model, each device establishes a direct trust on the other node by considering three trust evaluation metrics namely: end to end packet forwarding ratio, packet delivery ratio, and energy consumed. The indirect trust between devices is calculated based on the recommendations obtained from the neighbor nodes. The final trust value is calculated by applying the fuzzy theory on the direct and indirect trust. Devices with a low final trust value will be isolated from the network. This allows honest devices to avoid possible collaboration and cooperation with selfish and malicious nodes. To evaluate the indirect trust of a trustee, the trustor uses the recommendations sent by the collaborative nodes in the network having past experiences with the trustee. The trustor must keep only the recommendations obtained from honest nodes. Recommendations obtained from malicious nodes must be filtered to enhance the accuracy of trust computations. For this aim, in [38], the trustor first measures its "social similarity" with the recommenders in terms of friendship, social contact and community of interest, then it decides if the recommendations received are trustworthy. Indeed, [38] assumed that nodes are more likely to trust each other and collaborate effectively when they are

socially connected and have common interests.

Intrusion detection Wireless networks are subject to several security attacks that can mainly classified in two types: external attacks and internal attacks. External attacks, also known as outsider attacks, are conducted by nodes that do not have direct access to authorized network resources [45]. These attacks are relatively less harmful because the external attackers have no prior knowledge of the network. In contrast to external attacks, internal attacks, also known as insider attacks, are conducted by legitimate nodes that have successfully passed the authentication phase and are member of the network. Insider attackers can easily launch attacks thanks to their prior knowledge of the network. External attacks may be avoided by many existing methods [46] such as encryption and signature. However, no effective method has been introduced for internal attacks. To detect and isolate internal malicious nodes from the network, it is important to design an intrusion detection system that is capable of periodically collecting information about the operation of the network and investigating the existence of internal malicious nodes. To this aim, several works in the literature introduced trust management into their intrusion detection systems to enhance the security of their networks. Trust allows continuous monitoring of the network and its components to detect unusual, suspicious and malicious behavior and activities that may jeopardize the reliability of network operations. Trust Management is used for the revocation of internal malicious nodes from the network to avoid each transaction with them and prevent them from making attacks. The integration of trust in the security of transactions reduces the uncertainty and the risk in exchanges between entities.
Authors of [47] proposed a cooperative trust-based Intrusion Detection System (T-IDS) to detect internal misbehaving nodes in the network. In T-IDS, each node periodically monitors the behavior of its one-hop neighbors and collaboratively calculates their trust values. Nodes whose trust value is below a certain threshold are reported to a 6LoWPAN Border Router to put their identities in the list of potential malicious nodes. The list is then shared with all nodes.

Access control In wireless networks, some information managed by network entities may be publicly available, such as weather reports, while other information must remain confidential, such as patient records. Indeed, it is necessary to prohibit the exposure of sensitive resources to undesirable adversaries to avoid the disclosure of confidential data. Access control prevents

unauthorized users (i.e., humans and devices) from accessing devices and network resources to maintain the confidentiality of data. However, sometimes nodes that are considered legitimate and have access permissions to resources can change their behavior over time and become malicious. The issue is that traditional access control mechanisms cannot detect such dynamic behavior. As a result, malicious nodes will continue to access network resources and get access to confidential and sensitive data [6]. Conventional access control must be adapted to meet the specific requirements of wireless networks. To this aim, trust management can be used to support traditional access control mechanisms by setting some limits on the calculated trust values of nodes that request access. Trust management provides nodes with a natural way to assess the behavior of other nodes. It can be considered as an additional attribute for validating access requests in such a way that only nodes with a high level of trust can access resources, while nodes with an insufficient level of trust will be denied. Authors of [48] incorporated trust and reputation in the access control policy to reach dynamic, decentralized and trustworthy access control between IoT devices. By combining trust and reputation with the attributes of IoT devices such as sensor types and hardware specifications, the authors of [48] prevented malicious nodes from gaining access to protected resources. Authors of [49] proposed a Fuzzy Approach to Trust Based Access Control in IoT. The approach considers three parameters for trust computation, namely experience, knowledge and recommendation, collected from device communications. The computed trust is then mapped to access permissions in order to perform access control.

Authentication Authentication is essential to validating the identity of nodes and ensuring the integrity of data. Without proper authentication practices, an adversary could potentially access confidential network resources. The adversary could even send false commands to endpoints monitoring sensitive processes of the network and cause critical failures leading to significant downtime, financial damages, and disclosure of nodes' private information. Because the wireless communication medium is accessible to any entity, there are no restrictions on access to the channel as long as the entity has the appropriate equipment and resources. The open and distributed nature of wireless networks and the lack of computing resources on nodes require existing authentication techniques to be adapted to these environments to better meet their security requirements. Since trust management provides a lightweight solution to monitor and judge nodes, it was used by several works to adapt and enforce existing authentication techniques. Authors of [50] proposed an ef-

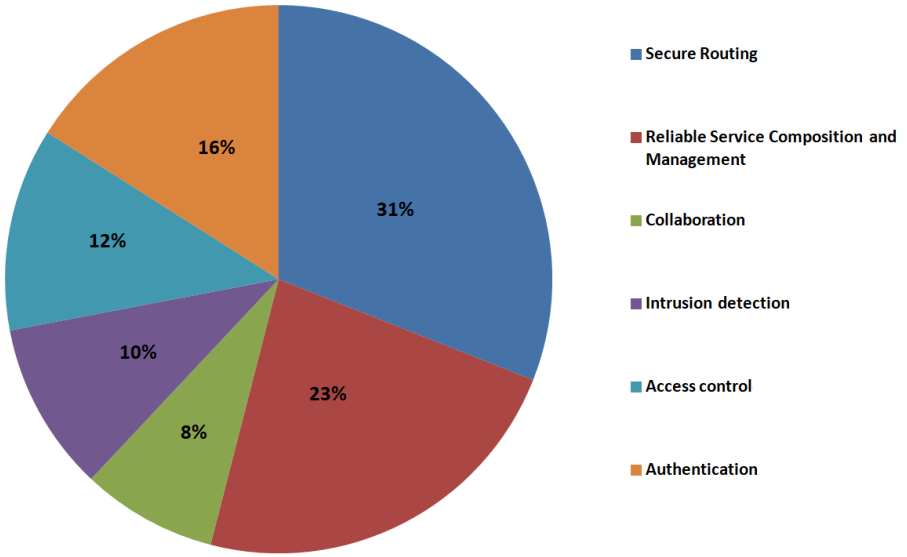


Figure 2.3: Percentage of efforts made in different trust management applications

ficient adaptive security model for IoT networks, based on trust management. The proposed model reduces authentication overhead by authenticating packets only when needed. The purpose of this model is to significantly reduce the energy consumption of nodes due to the message authentication process and to remain secure. Indeed, each node decides locally to authenticate or not the received message according to the level of trust that it assigns to the message sender.

In Fig. 2.3, we give the percentage of efforts made on different trust management applications from the year 2017 to 2022. This figure is based on 1215 papers found from theses, journals, books and conference proceedings. We notice that 31% of the trust management-based models in the literature, from the year 2017 to 2022, have been used to secure routing, followed by Reliable Service Composition and Management (23 %) and Authentication (16 %).

2.3.5 Categories of Trust Management Models:

Trust management models are classified into three categories, namely entity-centric, data-centric and hybrid trust management models:

Entity-centric Trust Management Models : This category of models is concerned by the legitimacy and reliability of the nodes participating in the various activities of the network. Indeed, when evaluating the trust, this category considers the behavior of the trustee in the network, its reputation and the recommendations of the neighbors towards it. However, this requires suffi-

cient information about the trustee and its neighbors for an accurate evaluation of the trust. This can be challenging in dynamic networks such as IoT and VANET networks where nodes change locations frequently. Models in this category do not consider the authenticity of messages exchanged because they assume that there is no guarantee that messages sent by honest entities are not corrupted, which may induce the model to incorrectly judge these honest nodes. Several entity-based trust management models have been proposed in the existing research works such as [23], [34] and [51]. These works computed the trust of a node based on its cooperation rate and its reputation in the network. Consequently, only legitimate nodes are chosen as relay nodes for data dissemination [45].

Data-centric Trust Management Models : In the evaluation of trust, this category of models focuses on the reliability, legitimacy, accuracy, and quality of the data collected and shared with other nodes in the network. This data mainly includes reports, information related to user privacy, event warnings, etc. Data-Centric Trust Management Models aim to verify the authenticity of the data produced, that can be modified and distorted by the malicious nodes to attack the network. The falsification of data can have significant consequences especially in critical environments like IIoT networks where nodes are led to handle critical data and a simple change in this data can lead companies to great financial and human damages. The major limitation of these trust management models is that they do not operate adequately in cases of information scarcity due to the lack of sufficient evidence. Several Data-Centric Models such as [52], [53] and [54] have been proposed in the literature. For example, [54] measures the credibility of a vehicle in VANETs by comparing the alerts shared by this vehicle (concerning an incident) with the alerts shared by vehicles that are close to the incident. If the vehicles confirm the correctness of the alert, the data shared by the vehicle will be considered honest and will be taken into consideration. Otherwise, this data will be ignored. However, this may result in delays and data loss in heavy traffic scenarios.

Hybrid Trust Management Models : This type of model combines both the reliability of the node and the exchanged data for a more efficient trust computation. Indeed, these models consider the authenticity of the data collected and exchanged, the behavior of the trustee in the network, its reputation as well as the recommendations of the neighbors towards it. Several Hybrid Trust Management models have been presented in the literature such as [55].

2.4 Conclusion

In this chapter, we provided a detailed introduction of the trust concept in the field of communication and networking, based on definitions from several disciplines. We also highlighted the crucial role that the trust plays in enhancing security and reliability in wireless networks. Trust ensures the effective use of wireless network services and applications by providing a continuous monitoring of the network and its components. It allows to detect unusual, suspicious and malicious behaviors that may compromise the security of the network.
However, trust modeling is not an obvious task, especially since there is no single and standard methodology for modeling and evaluating trust. Indeed, each work in the literature models and evaluates trust according to its own definition and perception of trust, also according to the needs of its application. Hence, to understand how trust can be modeled, in the next chapter we give an overview on the most used modeling methods identified for wireless networks in the literature.

Chapter 3

Trust Modelling methods

Contents

3.1 Introduction **26**

3.2 Basic Methods **26**

3.2.1 Weight function 26

3.2.2 Relational measurement 27

3.2.3 Fuzzy Logic 29

3.2.4 Game Theory 30

3.2.5 Entropy Theory 31

3.3 Bayesian Methods **33**

3.3.1 Bayesian interference model 33

3.3.2 Dempster-Shafer (DS) theory 34

3.4 Graph methods **36**

3.4.1 Network features 36

3.4.2 Semiring 38

3.5 Machine learning methods **40**

3.5.1 Support Vector Machine (SVM) 40

3.5.2 K-Means Clustering 42

3.6 Conclusion **44**

3.1 Introduction

Trust management is composed of four modules including trust composition, trust propagation, trust aggregation and trust update. In this chapter, we focus on the trust aggregation module, more specifically on the trust modeling methods that constitute this module and allow to have a final trust value from the observations collected by the composition module. The final trust value of an entity reflects its reliability.
As explained in 2.3.4, trust management allows to reach various security and confidentiality objectives required to maintain a reliable functionality of wireless networks. These objectives can be achieved by employing the appropriate trust modeling methods. However, there is no single and standard methodology for modeling and evaluating trust; each work in the literature models and evaluates trust according to its own definition and perception of trust, also according to the needs of its application.
Hence, in this chapter, we give a comprehensive review of the literature on the widely used methods for modeling the trust in wireless networks. We classify the trust modeling methods according to the mathematical field to which they belong. We also discuss for each method its theories and how it was applied to trust modeling.

3.2 Basic Methods

Basic Methods are simple methods that use basic mathematical constructs for modeling trust. The mathematical models used by these methods are based on equations derived from known and common basic constructs because there is no fixed manner to construct these equations.

3.2.1 Weight function

The weight function is the simplest and most used technique for calculating the trust. Indeed, it is a mathematical tool for calculating sums, integrals or averages in which some elements will have more importance or influence than others in the same set. The result is called a weighted sum or a weighted average.

The weighted sum : consists to overlay several elements by multiplying each of them by its weight and adding them together. Several models use the weighted sum to combine multiple trust factors [24] or to have a single trust score (T) from the direct trust value and the indirect trust value [23] as follows:

$$T = \sum_{i=1}^{N} w_i f_i \qquad or \qquad T = w_d T_d + w_{ind} T_{ind} \tag{3.1}$$

Where: N is the number of elements considered in the average, w_i is the weight of the i^{th} trust factor f_i [37], w_d and w_{ind} are the weights of the direct trust (T_d) and the indirect trust (T_{ind}), respectively [56].

The Weighted averages as arithmetic averages, combine the values of a set and normalize this sum by the number of elements, to obtain a representation of the set that takes into account all its elements. The difference in weighted averages is that the values are not weighted the same, which provides a more relevant combination.
The weighted average is widely used in trust modeling because it allows the aggregation of experiences and evidence (x_i), extracted from direct interactions or issued from recommendations, into a single value that quantifies the whole as in the equation (3.2). This resulting value can be used later in a more complex trust calculation models.

$$X = \left(\frac{\sum_{i=1}^{N} w_i x_i}{\sum_{i=1}^{n} w_i} \right) \tag{3.2}$$

Most models assign to the trustors the choice of defining the weight of each factor constituting the trust metric of the trustee. This allows the trustor to customize the trust model according to its own requirements which perfectly reflects the subjectivity of trust. Some models propose a dynamic adjustment of the weights to better meet the needs of the digital environments. For example in [57], the weight depends on the number of direct interactions between two nodes. Indeed, the more this number is important the higher the weight given to the direct trust value. Other works, considered the number of time intervals that have elapsed since the experiences (interactions or recommendations) were received. Indeed, authors of [33], [37], [8], [58] and [59] considered that the weight of newer and recent experiences is more important than older ones.

3.2.2 Relational measurement

This method uses similarity, correlation and variance measures to determine the link and the relationship between two variables. The measurement results of this method predict whether the trustor and the trustee are susceptible to trust each other in their future interactions.

Similarity measures: assess the extent to which two nodes are similar in terms of interest, opinions and ability to judge. The more similar the trustor and trustee are, the more they are able to trust each other and cooperate together in the network. For example, when a trustor wants to measure the indirect trust of a trustee, it only considers recommendations from nodes that have a high similarity with it i.e. recommender nodes that have the same ability to evaluate the trust as it does and the same opinions about the previous trustee nodes. To quantify the similarity of the assessments of two nodes i and k, several works as [60], [61] and [25] used the Cosine similarity metric defined as follows:

$$s(i,k) = \cos(\theta) = \left(\frac{V_i \ . \ V_k}{|| \ V_i \ || \ \ . \ \ || \ V_k \ ||} \right) \tag{3.3}$$

Where V_i and V_k are the vectors that gather the evaluations performed by nodes i and k, respectively. In these evaluations, the two nodes rate the behavior of their common neighboring nodes.
Cosine similarity metric allows a relative comparison of the evaluations provided by the two nodes i and K, rather than an absolute comparison. Even though the two nodes provide different ratings to their common neighboring nodes, for example Vi=[0.4, 0.2, 0.1] , Vk =[0.8, 0.4, 0.2], their opinions remain similar. Indeed, for both nodes the common node number 1 is the highest rated and the common node number 3 is the lowest rated.

Deviation measures: allow to evaluate the deviation of a variable relatively to its normal and usual behavior (results). These measures are used in trust modeling because the degree to which a device's variable deviates from a certain norm can be an indicator of the change in behavior of that device, and a signal about the abnormality and unreliability of its responses and actions in the network. To measure the deviation of a variable $X(i)$ for node i, [62] used the ln-deviation as follows:

$$DEV(i) = -\ln \left(\frac{X(i) + 1}{max_{j \in N}(X(j)) + 1} \right) \tag{3.4}$$

Where N is the number of devices in the network.

The trustworthiness of a recommender node is measured by its ability to provide good and reliable recommendations. This trustworthiness is measured by [37] as the distance between the value provided by the recommender R to the trustor node A in relation to the trustee node B, ($Re^{A,B,R}$), and the direct

trust value given by the trustor node A to the trustee node B, $DT^{A,B}$, as follows:

$$d = |Re^{A,B,R} - DT^{A,B}| \tag{3.5}$$

The distance d reflects the deviation of the reliability of the recommender's assessments from that of the trustor. It also determines the ability of the node R to provide accurate recommendations.

3.2.3 Fuzzy Logic

Instead of Boolean logic, where the only permitted values are "true" and "false", fuzzy logic allows for intermediate values, which take the form of fuzzy sets such as "not very true" and "rather false". This provides a better representation of the situations where a statement cannot be identified as completely false or true. A fuzzy inference system is a system composed of three main blocks: fuzzification, inference engine and defuzzification.

- **Fuzzification:** allows to interpret the input variables of a decision model. Therefore, it is needed to make explicit, for each variable in its value interval, the different states it can take; i.e. the subsets to which this variable can belong. The fuzzification is done through membership functions. These functions are specified for each subset, their role is to define the degree of membership of a variable to a given subset.
- **Inference engine:** In this block, the user can parameter the logical operators and the decision rules as " If...AND/OR ...., then... " according to its own experiences and requirements. Thanks to the inference engine, the user applies the rules that it have fixed to the fuzzy input variables. This allows to have the membership degree of the output variable to the different subsets to which it can belong.
- **Defuzzification:** This block consists of translating the fuzzy set of output variable obtained through the inference engine into a crisp numerical value. Defuzzification can be performed by several methods, the most widely used are the average of maxima method and the center of gravity method.

Several works as [49], [31], [54] and [37] used the fuzzy logic to apply uncertainty in trust decisions. The process described above was implemented in [49] to assess the trust of IoT devices according to three variables: Experience

(EX), Knowledge (KN) and Recommendation (RC) [12]. For each linguistic input variable (i.e. EX, KN and RC), three linguistic terms (i.e. Good, Average, Bad, etc.) were associated using membership functions. Authors of [49] defined fuzzy rules and logical operators for two reasons. The first reason is to map the values of the fuzzy sets to the values of the possible fuzzy sets (Average, Low, good) describing the trust. The second reason is to measure the membership degrees for the trust value in each of these possible fuzzy sets. Lastly, [49] used the Center-of-Gravity method to convert the fuzzy value of trust into a crisp value.

3.2.4 Game Theory

Game theory is a sub field of economics and applied mathematics [63]. It is used to study situations where individuals named players make decisions knowing that the outcome of their own choices depends on that of others.
By definition, trust is dynamic, it changes over time. Specifically, history and past experiences significantly affect the ability to trust in the future. This dynamic aspect of trust can be modeled by game theory using dynamic games where players change their strategies according to their past experiences and history. In wireless networks, trust is based on the assumption that nodes are rational in the sense that they make appropriate choices to maximize their gains and minimize their losses. Game theory can be applied to model the rational behavior of nodes using non-cooperative games when these nodes selfishly pursue their own interests, or using cooperative games when the nodes cooperate with each other to reach a specific objective. For all the reasons explained before, game theory is widely applied for trust modeling [64]. It enhances the trust-based recommendation systems by rejecting suspected fake scores [65] [66]. Also, it helps the trustor to define the best strategy for launching its anomaly detection technique to mitigate dishonest activities [67].

Authors of [65] proposed to use game theory to reach robust trust management in IoT networks. For this purpose, the proposed model classifies IoT devices into two categories, trusted devices called TN and To-Be TrustedNodes called TBTN. The TN computes the trust value of the TBTN and sends this value with the id of the TBTN as a message to the edge node. The TN can be honest, i.e., it always sends correct and truthful information, or malicious, i.e., it attempts to deceive the edge node by providing fake information about the reputation score of some TBTN. At the reception of the message, the edge node, not aware of the specific type of TN, must define its strategy: Either accept the message and pass it to the blockchain participants to update the

TBTN reputation, or reject the message. Since each strategy (accept or reject) is characterized by a payoff, the edge node will choose the strategy that will maximize its payoff.
The interactions among a TN and an edge node are modeled in [65] as a game, where the payoff structure of this game is presented as follows:

- The edge node receives a positive reward when it accepts a message sent by an honest TN.
- The edge node receives a penalty when it accepts a message sent by a malicious TN.
- The edge node does not receive any reward or penalty when it rejects a message.
- The TN receives a reward when its message is accepted. Otherwise, it gets a punishment.

As the interactions among the two nodes are repeated over the time, the edge node can build its belief on the TN. Thus, based on the payoff structure and the belief, the edge node can define the best strategy to adopt when receiving a message, so that suspected fake trust values are rejected when computing the reputation score of the TBTN.

3.2.5 Entropy Theory

The purpose of entropy theory is to quantify and measure the expected uncertainty associated with a random variable according to the knowledge of all outcomes and the probabilities of these outcomes. If X is a discrete random variable, its entropy will be defined as follows:

$$H(X) = \sum_{i=1}^{n} P(x_i) log_b P(x_i) \tag{3.6}$$

In trust modeling, the base of the logarithm is generally chosen equal to 2.

Several works used entropy in trust modeling as a trust value [68] [69] or as a tool to calculate the similarity between two recommenders [70]. Other works such as [41] [71], [72] and [21] used entropy in trust modeling as a weight. In [68], trust is defined as the probability that a node cooperates in the network. Then, the trust of a node in [68] is measured by the uncertainty based on the

cooperation probability p of that node as follows:

$$T(p) = \begin{cases} H(p) - 1 & if \quad 0 \leq p < 0.5 \\ 1 - H(p) & if \quad 0.5 \leq p \leq 1 \end{cases} \tag{3.7}$$

Where the entropy H(p) reflects the level of uncertainty on the cooperation behavior of the node, it is calculated based on equation (3.8) as follows:

$$H(p) = -plog_2p - (1-p)log_2(1-p) \tag{3.8}$$

When the probability of cooperation is in the two extremes, the amount H(p) according to equation (3.8) decreases rapidly, as the certainty about the behavior of the node increases. When the probability of cooperation is too low (too high), the trustor is no longer uncertain about the node's cooperative behavior but has certainty, which leads to a fast decrease (increase) in the trust of the node according to the equation (3.7). On the other hand, when the probability is far from the extremes, around 0.5, the trust evolves slowly because the trustor is uncertain of the node's cooperative behavior.
To evaluate the indirect trust and for a node A accept the recommendations of node B, A and B must be similar. Indeed, they must have the same ability to judge and assess the trust of the nodes in the network. To this aim, the authors of [70] relied on entropy to measure the similarity of nodes in a recommender system.

$$H(A,B) = \sum_{i=1}^{n} p(d_i)log_2p(d_i) \tag{3.9}$$

Where, d_i is the difference between the trust scores measured by the two nodes A and B for their common nodes. $p(d_i)$ is the probability that the difference d_i occurs. Afterwards, [70] used the entropy calculated in equation (3.9) to quantify the similarity between two nodes A and B as follows:

$$Sim = \left(\frac{1}{1 + H(A,B)} \right) \tag{3.10}$$

The entropy measures the degree of confusion of a dataset. Indeed, the larger the entropy, the more confused the data are and the smaller the similarity between A and B. The smaller the entropy, the more concentrated the data and the greater the similarity between A and B.

3.3 Bayesian Methods

3.3.1 Bayesian interference model

The outcome of any interaction can be seen as a random variable, so the decision to initiate an interaction with a node depends on the trust given to that node. Indeed, trust can be assessed by the subjective probability [73] that the desired interaction will result in an estimated positive outcome. This interpretation has inspired several researchers to use probability theory and inference for modeling trust.
Bayesian inference consists in evaluating the distribution of a random variable $'X'$ given by some observations from a prior knowledge, $(p(X))$, corrected by the likelihood of the observed data, $(p(observation|X))$. This can be expressed by the following proportionality relation:

$$p(X|observation) \propto p(observation|X)p(X)$$

The probability obtained can in turn be used as a prior probability that will be corrected if new data are available. By doing this in an iterative manner, the probability of the hypotheses will converge towards "the truth".

Bayesian inference is widely applied for trust modeling [74], [21], [75]. Assuming a trustee node j that has already performed "c" successful cooperation and "u" non-cooperation with a trustor node i. The belief of node i that node j is legitimate, considering its past cooperative behavior as evidence, can be determined by Bayes rule as:

$$P(T|c,u) = \left(\frac{P(T)P(c,u|T)}{Normalization}\right) \tag{3.11}$$

T is a random variable that denotes the trust of node j, it reflects the proportion of successful cooperation of this node with node i over the total of $c+u$ cooperation performed. According to statistics, a proportional random variable can be modeled as a Beta distribution. Hence, the prior knowledge P(T) towards node j can be represented by distribution $Beta(\alpha, \beta)$, where α and β are the parameters of the distribution. These parameters represent the prior expectation of the number of successful cooperation and the prior expectation of the number of non-cooperation, respectively. $P(c,u|T)$ describes the probability that successful cooperation would be the outcome of our study if we knew that T was true. This probability is represented in several works such as [21] and [76] by the binomial distribution as $Binomial(c+u, T)$. Therefore,

the posterior belief described in equation (3.11) becomes:

$$P(T|c,u) = \left(\frac{Beta(\alpha,\beta)Binomial(c+u,T)}{Normalization} \right) \tag{3.12}$$

$$= Beta(\alpha + c, \beta + u) \tag{3.13}$$

The steps of the derivation from equation (3.12) to equation (3.13) are detailed in [77]. Then, the expected trust value toward node j from node i is the statistical expectation of $Beta(\alpha + c, \beta + u)$ which is:

$$E[T|c,u] = \left(\frac{c+\alpha}{c+\alpha+u+\beta} \right) \tag{3.14}$$

3.3.2 Dempster-Shafer (DS) theory

DS theory is a well-developed evidence theory [78], it allows to combine evidence from different independent sources in order to achieve a degree of belief that considers all available evidence.
Let X be the set of all possible states of a considered system and 2^X the power set which gathers all the subsets of X, including the empty set $\emptyset$. These subsets represent the propositions regarding the real state of the system, and contain all and only the states in which the proposition is true. DS theory attributes a belief mass m to each element in the power set, via the basic probability assignment function (bpa) [79] formally defined as follows:

$$m : 2^X \rightarrow [0,1] \tag{3.15}$$

This function has two properties, first the mass of the empty set is zero

$$m(\emptyset) = 0 \tag{3.16}$$

Second, the masses of all members of the power set is 1:

$$\sum_{A \epsilon 2^X} m(A) = 1 \tag{3.17}$$

The mass m(A) is a measure of the belief that is attributed to the set A exactly. To combine two mass functions m_1 and m_2 over X, the Dempster's combination rule [78] is used in the following way:

$$m_{1,2}(A) = (m_1 \oplus m_2) \tag{3.18}$$

$$m_{1,2}(A) = \begin{cases} \left(\frac{1}{K}\right) \sum_{B \cap C = A} m_1(B) m_2(C) & when A \neq \emptyset \\ 0 & when A = \emptyset \end{cases} \tag{3.19}$$

Where:

$$K = \sum_{B \cap C = \emptyset} m_1(B) m_2(C) \tag{3.20}$$

K measures the level of conflict between the two masses. The normalization factor $K-1$ allows to ignore these conflicts and to attribute any mass involved in the conflict to the null set. The contribution of DS theory compared to probability theory is that it allows a joint evaluation of any set of possible states belonging to X. Several works have been based on the DS theory as a decision-making tool [77] [34] in their trust process or as an aggregation tool for recommendation scores [78] [58]. For example, [78] applied this theory to calculate the indirect trust by aggregating recommendations from different neighboring nodes. For this purpose, [78] considered the set X={Legitimate, Malicious} where Hypothesis H = {Legitimate}, ¬ H = {Malicious} and U = {H, ¬ H}. Each node $x \epsilon N$, where N is the set of common neighbours between the monitoring node A and the monitored node B, computes three masses m(H), m(¬ H) and m(U) that reflects its belief on node B as follows:

$$m_x^B(H) = \left(\frac{R_A^x T_x^B}{\sum_{x \epsilon N - \{A\}} R_A^x} \right) \tag{3.21}$$

$$m_x^B(\neg H) = 0 \tag{3.22}$$

$$m_x^B(U) = 1 - \left(\frac{R_A^x T_x^B}{\sum_{x \epsilon N - \{A\}} R_A^x} \right) \tag{3.23}$$

Where: The recommendation credibility R_A^x is used as a weight to differentiate legitimate and malicious recommendations. T_x^B is the direct trust computed by x for node B. To calculate the indirect trust IT_A^B of node B, A uses the Dempster's combination rule described in equation (3.18) to combine all the masses sent by the neighboring nodes belonging to N.

$$IT_A^B = m_x^B \oplus m_y^B ... \oplus m_z^B \tag{3.24}$$

3.4 Graph methods

Graph methods allows to model a wide variety of problems by reducing them to a study of vertices and edges. Since most wireless networks can be represented as graphs with nodes and edges, the properties and characteristics of graphs can be used to model the trust between the nodes of the network.

3.4.1 Network features

Graph methods allow to represent the structural features of the network, which can model the trust of a node based on its structural position in the network as well as on its relations with the other nodes of the network.

a) Networks overlap: Based on the modeling of trust in human networks and especially in social networks, the overlap of the networks of two users can be an indicator of the mutual trust between them. Indeed, this overlap evaluates the role of each user in the other's network. In this direction, several works model the trust between two nodes by using the number of common friends between them as an indicator of the overlap of their two networks. Indeed, nodes tend to trust their friends or nodes with strong social ties (with many mutual friends). The more friends two nodes have in common, the more these nodes are likely to trust each other because they are indirectly linked.

For node 'i' to trust node 'j', 'i' should measure the degree of centrality of node "j" in its network. This centrality is measured in [18] and [19] as follows:

$$C(i,j) = \left(\frac{N_{i,j}}{N_i}\right) \tag{3.25}$$

Where N_i is the friends of node 'i' and $N_{i,j}$ is the set of common friends between 'i' and 'j'. According to equation (3.25), the larger $N_{i,j}$ is, the more the two nodes are indirectly linked.

Jaccard's index has also been used by several works as [61] and [80] to measure the overlap of the networks of two nodes 'i' and 'j' as follows:

$$J(i,j) = \left(\frac{N_i \cap N_j}{N_i \cup N_j}\right) \tag{3.26}$$

Where N_i and N_i are the set of friends of node 'i' and 'j', respectively.

b) PageRank: PageRank has proven to be a useful tool for ranking nodes in a graph in many contexts [81]. The original PageRank algorithm was designed for web search, but many researchers as in [82] and [60], have proposed an adaptation of this algorithm to use it for trust modeling. In trust modeling context, PageRank algorithm allows to rank the nodes in the network according to their popularity and importance measured as their global reputation in the network. The Pagerank score reflecting the global reputation of node 'i' in the network can be calculated as follows:

$$RP(i) = \left(\frac{1-\alpha}{N}\right) + \alpha \sum_{j \epsilon M(i)} \left(\frac{RP(j)}{L(j)}\right) \tag{3.27}$$

Where N is the total number of nodes, $M(i)$ is the number of nodes that trust node 'i' and $L(j)$ corresponds to the number of nodes that node 'j' trusts. As $L(j)$ increases, the score $RP(j)$ reflecting the overall reputation of node j becomes less important and less considerable in computing the reputations of other nodes according to the equation (3.27). Indeed, a node that trusts a large number of nodes in the network is considered to be undemanding in its trust judgments.

c) Adjacency matrix: Let $G = (X, E)$ be a finite and directed graph, where $X = \{x_1, x_2, ..., x_n\}$ is the set of vertices of the graph and E is the set of edges which links each two vertices. The adjacency matrix of graph G is the matrix $M(G) \in R^{n \times n}$ whose coefficients $m_{i,j}$ are defined by :

$$m_{i,j} = \begin{cases} 1 & if \quad (x_i, x_j) \in E \\ 0 & if \quad x_i, x_j) \notin E \end{cases} \tag{3.28}$$

$M(G)$ is a square matrix, its dimension is $n \times n$. $m_{i,j} = 1$ if the vertices x_i and x_j have a relationship between them in the graph. Otherwise, $m_{i,j} = 0$.

In trust modeling, the trust network was presented by several works [11], [83], [84] and [85] as a finite, directed and weighted graph which has trust values as a weight of its edges instead of 0 and 1. Indeed, the input $m_{i,j} \in$ [0,1] represents the trust value of node x_j evaluated by node x_i. The adjacency matrix is used to represent and visualize the trust relationships that link the different nodes in the network.

d) Katz centrality Katz centrality quantifies the relative influence and importance of a node within a network by measuring the number of direct neigh-

bors (first-degree nodes) and also all other nodes in the network that connect to the node via these direct neighbors. Several works apply Katz centrality on trust networks for finding the most similar users as in [86], and for computing the indirect trust of a node using recommendations from direct and indirect neighbors [87].

The Katz centrality of a node 'i' is given mathematically as follows:

$$C_{Katz}(i) = \sum_{l=1}^{\infty} \sum_{j=1}^{n} \alpha^l (M^l)_{ji} \tag{3.29}$$

Where M is the adjacency matrix of the graph, the powers of M indicate the presence (or absence) of links between two nodes via intermediate nodes. For example, in the matrix M^3, if $m_{3,8} = 1$, this indicates that nodes x_3 and x_8 are connected via first and second degree neighbors of node x_3.

α is the attenuation factor, it penalizes connections established with distant neighbors via intermediate nodes. Hence, the longer the path, the lower the weight assigned to it. This reflects an important behavior of the trust in recommendation systems; trust becomes increasingly dispersed, as the number of intermediate recommenders in a path increases. For equation (3.29) to converge, the attenuation factor α must satisfy [86] the following condition:

$$\alpha < \left(\frac{1}{\lambda_M}\right) \tag{3.30}$$

Where λ_M is the largest eigenvalue [86] of M.

3.4.2 Semiring

A semiring is an algebraic structure $(S, \oplus, \otimes)$ that, for all elements $x, y, z \in S$, the binary operators $\oplus$ and $\otimes$ have the following properties:

- $\oplus$ is associative, commutative, with 0 as a neutral element:

$$(x \oplus y) \oplus z = x \oplus (y \oplus z) \tag{3.31}$$

$$x \oplus y = y \oplus x \tag{3.32}$$

$$x \oplus 0 = x \tag{3.33}$$

- $\otimes$ is associative, with 0 and 1 as an absorbing element and neutral element, respectively:

$$(x \otimes y) \otimes z = x \otimes (y \otimes z) \tag{3.34}$$

$$x \otimes 0 = 0 \tag{3.35}$$

$$x \otimes 1 = x \tag{3.36}$$

- $\otimes$ is distributive over $\oplus$:

$$(x \oplus y) \otimes z = (x \otimes z) \oplus (x \otimes y) \tag{3.37}$$

$$x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z) \tag{3.38}$$

These properties have been exploited by several research works as [88], [89] and [90] to interpret and model trust in wireless networks. The authors of [88] used semirings to find the trusted path between two nodes. This is formulated as a path problem on a weighted graph, where the edge from vertex i to vertex k represents the direct trust relationship between nodes i and k. This edge is weighted by the opinion (T_{ik}, c_{ik}) that node i has about k. T_{ik} corresponds to the trust value of node k provided by i and the confidence, c_{ik}, denotes the accuracy of T_{ik}. In [88], the problem of trusted path is addressed by using a path semiring $(S = [0,1] \times [0,1], \oplus, \otimes)$ where T_{ik} and c_{ik} has a value in [0,1], and operators $\oplus, \otimes$ are defined as:

$$(t_{ik}, c_{ik}) \otimes (t_{kj}, c_{kj}) = ((t_{ik}t_{kj}, c_{ik}c_{kj}) \tag{3.39}$$

$$(t_{ij}^{p1}, c_{ij}^{p1}) \oplus (t_{ij}^{p2}, c_{ij}^{p2}) \begin{cases} (t_{ij}^{p1}, c_{ij}^{p1}) & if \quad c_{ij}^{p1} > c_{ij}^{p2} \\ (t_{ij}^{p2}, c_{ij}^{p2}) & if \quad c_{ij}^{p1} < c_{ij}^{p2} \\ (t_{ij}^{*}, c_{ij}^{p2}) & if \quad c_{ij}^{p1} = c_{ij}^{p2} \end{cases} \tag{3.40}$$

Where pk is the k^{th} path and $t_{ij}^{*} = max(t_{ij}^{p1}, t_{ij}^{p2})$

The operator $\otimes$ combines opinions along a path, while the operator $\oplus$ aggregate opinions across multiple paths. To take into account the properties of trust, authors of [88] required that the operators $\otimes$ and $\oplus$ have each an additional property to those already defined by the semiring structure as follow:

- For $\otimes$: $x \otimes y \leq x$,y to express the deterioration of trust along a path.
- For $\oplus$: $x \oplus y \geq x$,y it concerns the aggregation through multiple paths, it reflects the improvement of the quality of opinions with the abundance

of opinions.

As long as the path contains more than two nodes, the operator $\otimes$ can be extended to iterative computation. Therefore, the trust value between a source node S and a destination node D is evaluated as follows:

$$(t_{SD}^{pk}, c_{SD}^{pk}) = \bigotimes_{E_{ij} \in pk} (t_{ij}, c_{ij}) \tag{3.41}$$

Where i and j are the intermediate nodes in the path pk linking the source node S to the destination node D, and E_{ij} is the edge between nodes i and j. Also, given more than two paths, the aggregate trust value of the trusted path between the source node S and the destination node D is:

$$(t_{SD}, c_{SD}) = \bigoplus_{pk \in paths} (t_{ij}^{pk}, c_{ij}^{pk}) \tag{3.42}$$

3.5 Machine learning methods

Machine learning is a type of artificial intelligence, it aims to give machines the ability to "learn" from data, via mathematical models. These machines improve their performance over time. In trust modeling, Machine learning methods have been used to build trust models based on historical data rather than on human perception of trust, which may be unrepresentative and difficult to model.

3.5.1 Support Vector Machine (SVM)

SVM belongs to the category of linear classifiers, which use a linear separation of the data. SVM is based on the principle of margin maximization to find the border between categories. For the SVM to find this border, it is necessary to give it training data. Indeed, we give the SVM a set of input data, of which we already know their output and to which category they belong. From this data, the SVM estimates the most plausible location of the border: this is the training phase necessary for any machine learning algorithm. Once the training phase is achieved, the SVM will be able to predict to which category an entry that it had never seen before belongs, without human intervention.
In trust modeling, SVM is a widely applied method for classifying nodes in a network into two main categories: legitimate and malicious.
Assuming a vector $x \in R^k$ composed of k features of the trust for each node. Authors of [41] use five trust features to train their model namely Co-work

Relationship, Cooperativeness, Frequency and Duration, Reward System, Mutuality and Centrality, and Community of Interest. In [91], packet drop rate, packet modification rate and RTS flooding rate are used as trust features to train the model. If we also assume that the decision is $d \in \{-1, 1\}$, where $d = 1$ represents the trust category and $d = -1$ represents the untrust category, the set of n trust data points allowing learning will be:

$$(x_1, d_1), (x_2, d_2),...,(x_n, d_n) \tag{3.43}$$

The purpose of the SVM algorithm during training is to find an optimal hyperplane $h(x) = w_1x_1 + ... + w_nx_n + b = \sum_{i=1}^{n} w_ix_i + b = w^Tx + b$ that correctly divides the n trust data points into trusted and untrusted with a maximum margin between the two categories. W is the weight vector, b is the bias and the margin is the distance between the two support vectors of each category. The formula of the margin is given by:

$$\left(\frac{2}{||w||}\right) \tag{3.44}$$

Then:

$$Max\left(\frac{2}{||w||}\right) \longleftrightarrow Min\left(\frac{||w||^2}{2}\right) \tag{3.45}$$

Finding the optimal hyperplane can be formulated by :

$$Optimization\ Problem \begin{cases} Min\left(\frac{||w||^2}{2}\right) \\ S.c.\ y_i(w^Tx_i + b) \geq 1 \quad \forall i = 1..n \end{cases} \tag{3.46}$$

This kind of problem is called a convex single-objective quadratic optimization problem under linear constraints. There are many methods to solve it, the most famous is the method of Lagrange multipliers. Once training is achieved and the parameters w and b of the optimal hyperplane are found, the SVM can classify a new input x by looking at the sign of h(x) as follows:

$$\begin{cases} h(x) \geq 0 \Rightarrow x \in trust\ category \\ h(x) < 0 \Rightarrow x \in untrust\ category \end{cases} \tag{3.47}$$

When there is no hyperplane capable of correctly separating the two categories, it is because the training data are non-linearly separable. Indeed, this is what happens almost all the time in practice. To circumvent the problem, several

works as [41] and [92] have used the kernel trick as a solution.

3.5.2 K-Means Clustering

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into k different clusters, in such a way that each dataset belongs only one group that has similar properties. Each cluster is represented by its centroid; the data point that represents the center of the cluster. It is calculated as a weighted average of all the points within the cluster. The k-means clustering algorithm mainly performs two tasks called expectation-maximization. The expectation task attributes each data point to its most proximate centroid. Followed by the maximization task that determines the new centroid for each cluster by computing the average of all points in the cluster. The functioning of the algorithm is detailed as follows:

Algorithm 1 K-Means Clustering algorithm

1. Select the number K to decide the number of clusters;
2. Select randomly k centroids;
4. **Repeat**
5. **Expectation:** Attribute each data point to its closest centroid;
6. **Maximization:** For each cluster, calculate the new centroid.
7. **Until** The position of the centroids does not change;

In trust modeling, to train a supervised learning algorithm, the training dataset given as input to this algorithm in the training phase must be labeled. Each training set must be identified as (xi, yi) according to what we have explained previously. However, training labels are not readily available, hence the need for a method for labeling. In order to overcome the problem of unavailability of a labeled training set based on trustworthiness relationships, authors of [41] use the k-means clustering algorithm to identify two different clusters. Each cluster groups similar interactions based on the extracted trust features, also each cluster represents a label y = [0,1] where y=1 is the label of trustworthy interactions and y=0 is the label of untrustworthy interactions between two IoT devices.

K-means clustering is also one of the methods used in trust-based recommender systems [93] to allow a trustor to find the recommender nodes most similar to it, and therefore predict the trust level of nodes with which it has no previous experience. Indeed, k-means clustering allows to gather in k groups the nodes that have similar rating towards the same trustee nodes.

In Table 3.1, we compared the different methods proposed for modeling trust in the literature. We highlighted the advantages and disadvantages of

each proposed method.

Table 3.1: Comparison between the Trust Modelling methods

Methods	Approaches	Advantages	Disadvantages
Basic methods	Weight function	-Combine multiple trust parameters -Combine direct and indirect trust -Aggregation of experiences. -Support the subjectivity of trust.	- Sometimes unable to capture network dynamics due to static weight assignment.
	Relational measurement	-Measure the similarity and deviation between nodes in term of interest and assessments. -Detect the deviation of recommendations.	-Vulnerable to Bad-mouthing attacks and Ballot stuffing attacks when measuring similarity. -To measure the deviation, it is necessary to know the normal behavior of the nodes, which is not always possible.
	Fuzzy Logic	-Apply uncertainty to trust decisions. -Allow for intermediate values of trust. -Can take several trust parameters.	-High computational overhead
	Game Theory	-Model the dynamic aspect of trust using dynamic games. -Model the rational behavior of nodes using cooperative and non-cooperative games. -Enhance the trust-based recommendation systems.	-This theory is not widely applied to trust modeling because it is still an evolving theory. -Useful for decision making but not for the calculation of the trust metric.
	Entropy Theory	-Reflect the level of uncertainty on the behavior of a node in term of cooperation, honesty, etc. -Evaluate the similarity between two recommenders.	-Inappropriate when the number of trust parameters increases.
Bayesian methods	Bayesian interference model	-The trust of a node is expressed as a belief updated by new observations. -Take into consideration the behavioral history of the nodes. -Effective against on-off attacks	-Updating the belief at each individual stage is a difficult task.
	DS theory	-Calculate the indirect trust by aggregating recommendations. -Combine evidence from different independent sources.	-High computational overhead, node must calculate to each element in the power set a mass.
Graph methods	Network features	-Model the trust of a node based on its structural position in the network and on its relations with other nodes. -Measure the overall reputation of nodes in the network.	-Only social metrics are used. -The evaluation of trust is very subjective; it must be reinforced with QoS trust metrics. -Vulnerable to trust-related attacks. -Each node must have a global view on the whole network which is not always possible.

Methods	Approaches	Advantages	Disadvantages
Graph methods	**Semiring**	**-Find the trusted path between two nodes. -Widely used in routing. -Combine opinions along a path.**	**-Assume that the trust metrics are already calculated. -Vulnerable to trust-related attacks.**
Machine learning methods	**Support Vector Machine**	**-Build trust models based on historical data rather than on human perception of trust.**	**-Inappropriate when the trust features are reduced. -Training data for trust does not exist. -Appropriate training data are needed for each trust application. -High computational overhead and convergence time.**
	K-Means Clustering	**-Give a label to the training dataset to train supervised learning algorithms. -Gather nodes that have similar rating.**	**-Difficult to determine the optimal value k from the beginning.**

3.6 Conclusion

The effectiveness of the trust management models, proposed to improve security systems of wireless networks by using the notion of belief, depends on how these models manage the trust. Indeed, the most difficult step in the trust management process is the trust modeling, because there is no single standardized method for this purpose. Each model proposes a trust modeling method according to the perception that its designers have of the notion of trust. Hence, the main challenge is to effectively aggregate all the data collected about an entity, during the monitoring process, in order to evaluate its trust level.

To clarify the readers on the trust modeling methods existing in the literature, in this chapter we provided a classification and deep analysis of the different methods used for trust modeling in wireless networks. We considered four classes of trust modeling methods including, Basic Methods, Bayesian Methods, Graph methods and Machine learning methods. We defined the mathematical theories available for each methods and how they can be applied to modeling trust. The purpose of analyzing the trust modeling process is to provide a clear vision towards the design of an effective trust management model for future researchers.

Chapter 4

Trust Management in IoT networks

Contents

4.1 Introduction 45

4.2 Requirements and constraints of IoT networks 47

4.3 Related work 48

4.4 The proposed trust management model analysis 53

4.4.1 Trust management Model Overview 53

4.4.2 Trust parameters 54

4.4.3 Pseudo code of the estimation algorithm 58

4.4.4 Probabilities of transition from one state to another . . . 58

4.5 Performance evaluation 60

4.6 Conclusion 64

4.1 Introduction

To provide advanced and intelligent services, IoT devices have to manage data related to the privacy of their users which makes these devices vulnerable to security threats. The openness and the large scale of IoT networks coupled

with the limited resources of IoT devices, make the traditional security measures usually deployed in traditional Internet inappropriate to be applied in IoT networks. For this purpose, several researches rely on trust management as a solution to improve the reliability of IoT networks, especially in the presence of internal malicious devices.
Indeed, trust management provides continuous monitoring of IoT devices to predict their future behavior and detect any suspicious activity that could jeopardize the security of the network. This ensures that only legitimate devices participate in network activities such as routing, collaboration, service Composition, etc. Certainly, there are several works in the literature that propose trust management models for IoT. However, a deep analysis of these models reveals that a large majority do not consider the inherent characteristics of IoT networks, such as limited resources, heterogeneity, scalability, etc. Further, these trust management models are vulnerable to trust-related attacks, which weakens the security of IoT networks instead of enhancing it.

Based on the research gaps and open challenges, in this chapter, we propose a dynamic analytical trust management model where each IoT device in the network evaluates the trust level of its neighbors using a local trust metric, before interacting with them. This allows the device to predict the future behavior of its neighbors and avoid probable security threats. In our proposed trust management model, the trust metric of a monitored device can change state depending on two principal aspects:

Cooperation: The monitored device cooperates correctly in the network if it forwards data and provides recommendations [39].

Honesty: we define two types of honesty:

- **Honesty related to the direct interactions:** reflects the reliability of the direct interactions executed by the monitored device. To evaluate this honesty, the monitor device observes the honest interactions and suspicious untrusted interactions performed by the monitored device such as untrusted recommendations, untrusted forwarding, etc.
- **Honesty related to the indirect interactions:** reflects the reputation of the monitored device in the network. It is calculated on the basis of recommendations given to the monitored device by devices in the network that have already had experiences with it.

Hence, the trust metric of a monitored device can increase, decrease, remain unchanged or tend to zero according to its behavior. We formalized the state

changes of the trust metric by using a discrete-time Markov chain, which is an effective mathematical method that relies on the previous state of a process to predict its future state.

The rest of this chapter is structured as follows. In section 4.2, we give an overview of the requirements and constraints that need to be considered to manage the trust in IoT networks. In section 4.3, we review some related trust management models designed for IoT networks. In section 4.4, we describe and detail the distributed analytical trust management model proposed for IoT. In section 4.5, we discuss the numerical results. Finally, we conclude the chapter in section 4.6.

4.2 Requirements and constraints of IoT networks

IoT networks are characterized by many requirements and constraints that need to be considered when designing and deploying trust management in these networks. The main requirements and constraints of IoT networks are:

- **Heterogeneity of devices:** The evolution of communication protocols and applications has enabled the IoT to interconnect heterogeneous devices with different operating (e.g., operating platforms) and hardware (e.g., processing power, storage capacity, battery) characteristics. These different characteristics must be considered when deploying trust management in IoT networks.
- **Scalability:** IoT is evolving rapidly; according to statistics made by the IoT ANALYTICS Research [1], the number of connected devices is increasing significantly to reach 21.5 billion by 2025. Indeed, this significant increase raises scalability issues and affects the ability to properly manage and monitor IoT devices. To operate optimally and efficiently, trust management must take into account the rapid evolution of IoT networks.
- **Resource-constrained devices and communication cost:** IoT devices are limited in terms of processing power and storage capacity. Indeed, operations requiring high computational and storage costs directly affect the performance of IoT devices. As well, costly communication operations affect network performance in terms of bandwidth and availability. The cost of computation and communication is one of the main concerns of IoT networks, therefore, trust management must involve op-

[1] https://iot-analytics.com/lpwan-technologies-cellular-mnos/

erations that are adapted for resource-constrained devices, and that do not consume the available resources of the network.

- **High dynamism** The topology of IoT networks is very dynamic; it can change randomly at unpredictable times. This is due to the mobility of IoT devices and the fact that these devices can leave or join the network at any time. Also, IoT devices can have a dynamic behavior, e.g., a device may behave honestly at the beginning of its deployment to gain the trust of the network and then change its behavior to become malicious in order to perform internal attacks. Therefore, trust management must be able to adapt to the dynamic topology of the IoT network and be sensitive to the dynamic behavior of devices.

- **Energy consumption** One of the most crucial challenges of IoT devices is energy consumption, as most of these devices run on batteries. During the trust management process, IoT devices should not handle a large amount of information. Indeed, this will lead to a rapid drain of their batteries and a reduction in the life of the IoT network. Thus, trust management must provide a good compromise to ensure network monitoring while conserving the energy of devices.

In this section, we have provided an overview of the requirements and constraints that can be considered when designing and deploying trust management in IoT networks. It is well known that these constraints are related to the context of IoT deployment. For this reason, we focus only on the resilience to trust-related attacks whose purpose is to distort trust assessments. A trust management model that is vulnerable to these attacks will more weaken the security of IoT networks instead of strengthening it. To this end, the last section of this chapter is focused on evaluating the resilience of our proposed model to trust-related attacks.

4.3 Related work

In this section, we present a concise overview of the trust management models proposed in the literature to solve some security issues related to IoT networks.

Authors of [31] embedded the secure Trust (SecTrust) framework proposed in [94] into the RPL protocol [159] to provide protection against Rank and Sybil attacks. Indeed, the SecTrust framework supports secure communication, detects and isolates malicious and selfish nodes in IoT networks. For this purpose, every IoT node computes the trustworthiness of its direct neighbors

based on the computed direct trust value and the recommended trust value. The direct trust of a node is given as the probability of that node to have successful interactions with its direct neighbors. This reflects the reliability and the ability of the node to safely transmit packets to their appropriate destination. Recommended trust value is obtained by gathering the opinions collected from indirectly linked neighbors. SecTrust framework uses the concept of fuzzy threshold to define the trust level of an IoT node ["V1= no trust"," V2= poor trust", "V3= fair trust"," V4= good trust", "V5= complete trust"]. Hence, only nodes belonging to level V5 and V4, with a high level of trust are chosen for secure routing. Devices belonging to V3 are used just in the absence of V4 and V5 node categories. However, nodes belonging to V1 and V2 are considered as malicious or selfish nodes. The authors have proven the effectiveness of integrating SecTrust framework into RPL. This integration allows to make secure routing decisions and to cope with Sybil and rank attacks. However, the SecTrust framework does not deal with attacks launched by selective nodes that gain trust by transmitting a certain type of flow without another.

Authors of [24] proposed a Metric-based RPL Trustworthiness Scheme (MRTS) to address the trust issue in the building and the maintenance of routing paths from each node to the Border Router. MRTS integrates in the Objective function F0 a new trust-based metric ERNT (Extended RPL Node Trustworthiness). ERNT is employed to compute paths costs, and to choose the preferred parent. It is calculated as an average of the direct trust evaluation computed using three trust components: honesty, energy and unselfishness of the direct neighbor node, and the indirect trust evaluation given by the collaboration of the other nodes in the network. When calculating the direct trust, the authors of [24] assigned a different weight to recent and older experiences to differentiate them. This proposed trust management scheme addresses Rank falsification attacks in RPL and secures exchange of routing path between nodes. It also deals with different attacks such as Self-promotion, Bad-mouthing, and Ballot-stuffing attacks. However, this proposed trust management scheme suffers from coalition attacks and from the overhead costs of communication and energy consumption due to the increased computation. Moreover, authors of [24] have not validated their trust management scheme with simulations and numerical results to evaluate its performance.

To make proper and secure routing decisions in IoT networks, authors of [37] proposed to integrate a multi-fuzzy trust model (FDTM-IoT) into the RPL protocol as an objective function. In the FDTM-IoT, trust between two nodes is calculated based on a multi-stage fuzzy model that in the first stage

calculates the trust of three dimensions, namely Peer to peer communication quality (QPC), service quality (QOS) and contextual information. In the second stage, the outputs of the fuzzy inference systems of the three dimensions, calculated in the first stage, are entered as an input to the final fuzzy inference system. The output of the second stage is the final trust between two nodes. Each of the three dimensions used for the calculation of the final trust consists of sub-dimensions. The QPC dimension is evaluated based on the Last direct QPC observation, historical QPC information and indirect QPC information. The last direct QPC observation is evaluated by using the packets forwarding indication of the monitored node. In the calculation of the historical QPC, authors attributed a high weight to recent communications. And lastly, the indirect QPC is the trust level gained from recommendations received from neighbors. The untrusted recommendations in this model are filtered to avoid the Bad mouthing and Ballot-stuffing attacks. Indeed, the monitor node calculates the difference between its last direct QPC observation and the received recommendation value. If this difference exceeds a threshold of 0.2, the monitor node excludes this recommendation. However, this filtering requires that the monitor node already has a last QPC observation of the monitored node, otherwise it cannot be performed. To evaluate the trust of the "QOS" dimension, the authors used HETX and Delay criteria. The last dimension "Contextual information" depends on the context of the trust calculation. Since the trust in [37] is computed for routing context, the authors have set three criteria for the computation of the contextual information dimension: mobility of nodes, stability of link and remaining energy. The final trust obtained by the second fuzzy stage can belong to one of the fives trust levels T0, T1, T2, T3 or T4, where T0 is the level that contains the malicious nodes and T4 is the level that contains the complete trusted nodes. Only nodes belonging to T3 and T4 are suitable for secure communication. The trust model proposed in [37] is flexible; the criteria used to calculate the contextual information dimension can be changed according to the context of the application and the characteristics of the network.

Authors of [95] proposed a trust-based monitoring (TBM) scheme for protecting against illegal user access and media access control (MAC) spoofing in IoT communications. The intelligent agent evaluates the trust metric of each node in the network based on the number of acceptable and malicious messages the node requests or transmits during a communication period, and based on the physical attributes of its signal (Received signal strength (RSS)). The trust metric calculated for each node allows building a neural network.

Indeed, the trust metrics are the input of the learning process that takes a final decision about the legitimacy of a giving node. Only nodes present in the legitimate record of the access list are given ECC-based message authentication. This model proposed the integration between trust management and machine learning to make decisions concerning the authentication of IoT nodes and to minimize the detection time of MAC spoofing attacks. Performance analysis proved that the model allows a lower response and detection time. It also allows a reduced energy consumption, which improves the lifetime of the network. However, the proposed trust management model do not consider the mobility of nodes. Indeed, the mobility may wrong the estimation of the position and velocity vector of the nodes calculated by the RSS that allow the detection of spoofing attacks.

Authors of [96] proposed a trust management scheme to build a reliable SIoT network and improve decision making among the cooperative nodes in this environment. The trustworthiness of SIoT nodes is calculated by using social trust metrics, including direct trust, centrality, cooperativeness, community interest and service score. The direct trust of the trustee node is calculated based on the transactions made with the trustor node during their direct contact. Each transaction is weighted by a factor to differentiate between relevant and irrelevant transactions. Indeed, only relevant transactions will be taken into account when calculating the direct trust. Therefore, nodes that behave well during irrelevant transactions and change their behavior when the transactions are more relevant are considered as untrusted. Centrality represents the importance of the trustee for the trustor, it is proportional to the number of common friends between the two nodes. Cooperation is also calculated based on the number of common friends between the two nodes. Indeed, the authors assumed that the more the nodes have friends in common, the more they are susceptible to communicate with each other. However, this measure is subjective and does not really reflect the willingness of the devices to cooperate with each other. An additional limitation of this trust management scheme is during the trust update process, malicious nodes have the ability to recover their trust value if they behave well in the network. This enables On-off attacks, as nodes can recover their untrusted behavior as soon as they rebuild their trust values.

Authors of [97] proposed a distributed trust management protocol for a reliable service composition in IoT networks. Indeed, the proposed protocol evaluates the trust of IoT devices to detect misbehavior in service composition. In [97], the trust towards a trustee is built on the basis of three social attributes

including honesty, cooperativeness, and community-interest. Honesty is calculated by counting the number of suspected dishonest experiences of the trustee observed by the trustor in a period of time. The cooperativeness determines whether or not the trustee is cooperative with the trustor, this trust attribute is calculated by using the social friendship relationship among nodes. Authors of [97] are the first to consider social relationships in trust management for IoT networks. Indeed, the authors consider that friendly nodes are more likely to cooperate with each other. The community-interest attribute reflects whether or not the trustee and the trustor have similar capacity or common interests. This trust attribute is calculated by using the social relationship among nodes like co-location, co-work relationship and parental relationship. Each of these three trust attributes can be computed based on direct observations in the case where the trustee and trustor interact directly. Otherwise, on the basis of recommendations from neighboring nodes in the case where the two nodes do not interact directly. The proposed trust management protocol measures for the trustee the value of each trust attribute separately, but there is no aggregation of the three attributes to have a single trust value for the trustee. Also, the protocol deals with several trust-related attacks such as Self-promotion, Bad-mouthing, and Ballot-stuffing attacks. However, it is vulnerable to selfish attacks because the measure of cooperativeness in this work is not reliable.

Authors of [98] proposed a trust management scheme to cope with on-off attacks in IoT networks. In the proposed scheme, each node evaluates the communication trust and data trust of its neighbor. Indeed, the communication trust is evaluated based on the number of cooperation and non-cooperation performed by the monitored node. Trust in the data is evaluated based on the number of interactions containing normal data and the number of interactions containing erroneous data performed by the monitored node. Afterwards, the monitor node combines the result of its observations with those sent by the recommender nodes to evaluate the amount of good and bad experiences had with the monitored node. This combination is obtained by using the DS belief theory, and the final trust value of the monitored node is computed by using Bayes Probability distribution over the good and bad experiences. The interactions of the monitored node are weighted by an aging factor that gives more importance to the recent interactions and allows the scheme to detect the changing behavior of IoT nodes. The proposed scheme allows also the detection of on-off attacks by considering the time duration between the lower and higher trust value. However, the scheme is ineffective to address dishonest recommendation attacks such as Bad mouthing attack, ballot stuffing attacks

and coalition attacks.

The deeper analysis of the reviewed works shows that the trust management models proposed for IoT networks are vulnerable to trust-related attacks, which weakens the security of IoT networks instead of strengthening it. Also, these models did not consider the inherent characteristics of IoT networks. In this chapter, we propose an effective and distributed trust management model for IoT networks, while taking into account the requirements and constraints of these networks.

4.4 The proposed trust management model analysis

We use a discrete-time Markov chain to model the management process of the trust metric Tm of a monitored IoT node at the monitor node. Indeed, Markov chain is an effective mathematical method that relies on the previous state of a process to predict its future state. Hence, in this section we introduce the states transitions diagram corresponding to our proposed Markov chain, we also detail the calculation of the trust parameters allowing to compute the different transition probabilities from one state to another in the Markov chain.

4.4.1 Trust management Model Overview

In this section, we propose a dynamic and distributed trust management model for IoT networks, where each IoT node is monitored by its neighbors. The monitor node is one hop away from the monitored node. Indeed, the monitor node assigns to the monitored node a local Tm according to its cooperativeness and according to the honesty related to its direct and indirect interactions. Each node that joins the network is assigned an initial trust metric $T0$. The metrics Tm and $T0$ have a values in [0, 1].
Therefore, depending on the result of the monitoring process, the Tm of the monitored node may increase, decrease, remain unchanged or goes down to zero as shown in Fig. 4.1.

We assume that the change of states of the Tm depends on its previous state, on the cooperation of the monitored node, and on the honesty of the monitored node during its direct and indirect interactions. We formalize these state changes by using a discrete-time Markov chain with $M+1$ states, state 0 corresponds to the lowest level of trust where $Tm = 0$ and state M represents the uppermost level of trust where $Tm= 1$. We divide the interval [0, 1] of the

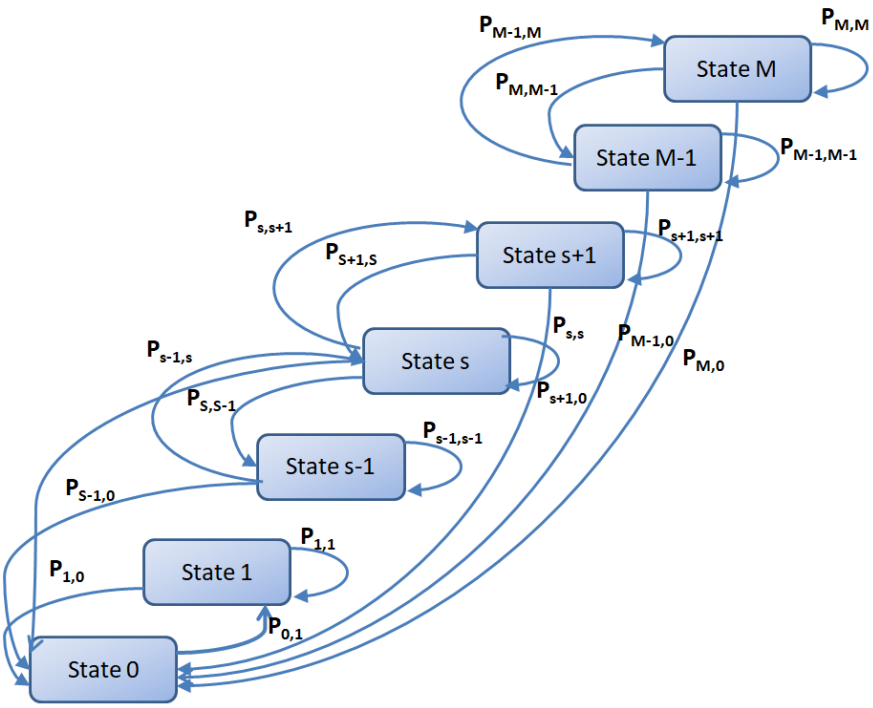


Figure 4.1: State transition diagram

Tm into $M+1$ states, each state corresponds to a step of ϕ.
In our trust management model, the number of transitions needed to reach Tm= 1 and the trust interval ϕ can be customized according to the requirements of the monitoring process and according to the level of security we want to ensure in the network. This makes our model flexible and adaptable.

The transition matrix corresponding to the proposed discrete-time Markov chain is:

$$P = (P_{s,s'}(t))_{0 \leq s,s' \leq M} \tag{4.1}$$

The current local Tm attributed by the monitor node is represented by a random variable $(Y_t)_{t \geq 0}$. Indeed, (Y_t) refers to a given state of the monitored node. The probability to transit from state s to state s' is:

$$P_{s,s'}(t) = P(Y_t = s' \mid Y_{t-1} = s) \tag{4.2}$$

4.4.2 Trust parameters

Before detailing the calculation of the different transition probabilities allowing the update of Tm, we will calculate three trust parameters related to the behavior of the monitored node as follows:

1) Honesty related to the direct interactions: This parameter is evaluated

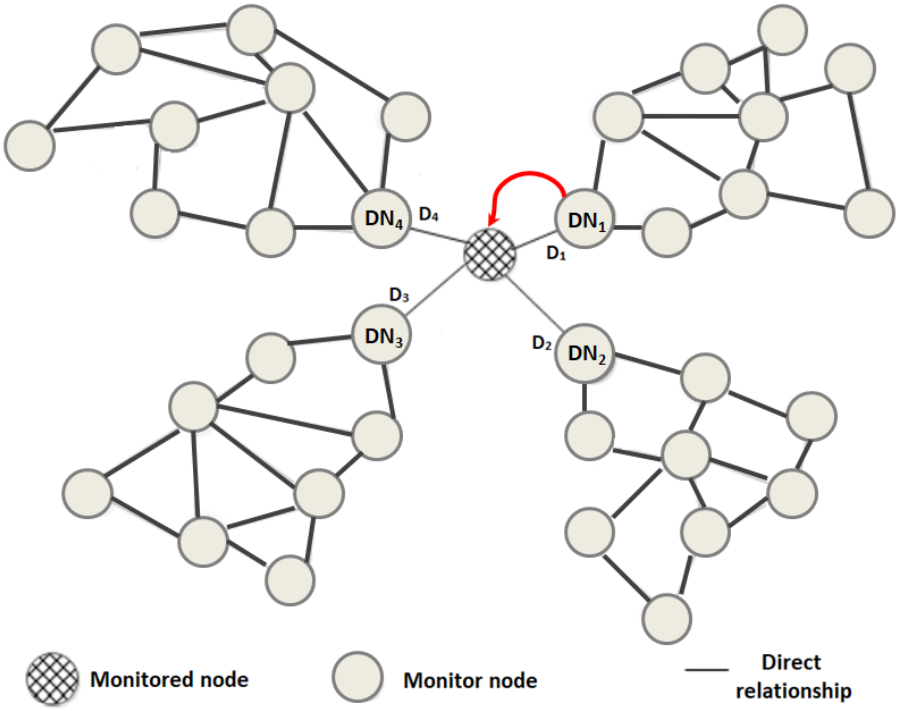


Figure 4.2: The honesty related to the direct interactions

based on the interactions related to routing and based on the recommendations provided by the monitored node. For application-related interactions, the monitor node will not be able to determine the good will of the monitored node because it does not have access to its collected and shared data to assess their reliability.
In order to evaluate the honesty related to the direct interactions carried out by the monitored node, first, we assume that the monitored node has N direct neighbors that are one hop away from it. let $DN_j|j \in \{1,2,\ldots,N\}$ represents the set of the direct neighbor nodes. Hence, each DN_j calculates a rate of honesty related to the direct interactions called D_j as shown in Fig. 4.2.

The rate D_j is the number of honest interactions (packet successfully transmitted and trusted recommendation) between the monitored node and its DN_j, weighted by an interaction factor E_{tr} and divided by the total number of interactions (TNV_j) between these two nodes. The factor E_{tr} represents the relevance of each interaction tr between two nodes. Indeed, this factor differentiates critical interactions from non critical ones [19], its value is in [0,1].

$$D_j = \left(\frac{\sum_{tr=1}^{TNV_j} (V_{tr} * E_{tr})}{TNV_j} \right) \tag{4.3}$$

The parameter V_{tr} has a value of 1 if tr is a honest interaction or 0 if tr is

an untrusted interaction.

Then, the probability p_d that the monitored node makes honest interactions with the monitor node DN_j is calculated as follows:

$$p_d = D_j * p_{net} \tag{4.4}$$

The probability p_{net} depends on the transmission conditions in the network, it is specific to each pair of nodes. It depends on the state of the network including congestion, retransmission, obstacles, quality of links between the sender and the receiver, which induce at the same time false positives and false negatives during the monitoring process. The purpose of introducing this probability is to consider the constraints of the monitoring environment in trust management.

2) Honesty related to the indirect interactions: To assess this honesty, the monitor node DN_j relies on the recommendations propagated from different nodes having past experiences with the monitored node. Recommendations are not distributed among all nodes in the network otherwise they will consume network resources. Indeed, in our proposed model, the monitor node DN_j requests recommendations just from its direct neighbors and not from all nodes in the network. For this purpose, first, each DN_j with its neighbors organize themselves into a set of groups. The groups formed have not necessarily the same size. Let $MD_i|i \in \{1,2,\ldots,k\}$ represents the set of member nodes contained in the group formed by the DN_j. Each DN_j calculates the rate I_j of honesty related to the indirect interactions as shown in Fig. 4.3.

The rate I_j is the value of recommendations (R_{MD_i}) from each member MD_i belonging to the group and that have already experiences with the monitored node, divided by the total number (k') of recommendations.

$$I_j = \left(\frac{\sum_{MD_{i=1}}^{k'} (Tm_{(MD_i)} * R_{MD_i})}{k'} \right) \tag{4.5}$$

To calculate the rate I_j, DN_j only considers the values of R_{MD_i} which are close to each other, extreme recommendations (too high or too low) are considered as spam. The MD_i that provides spam values will be classified as a malicious node that intends to perform Bad-mouthing attack or Ballot stuffing attack. The spam elimination process is performed by the estimation algorithm 2. This algorithm is detailed in the next subsection.

k' is the number of recommendations after removing the spam values.

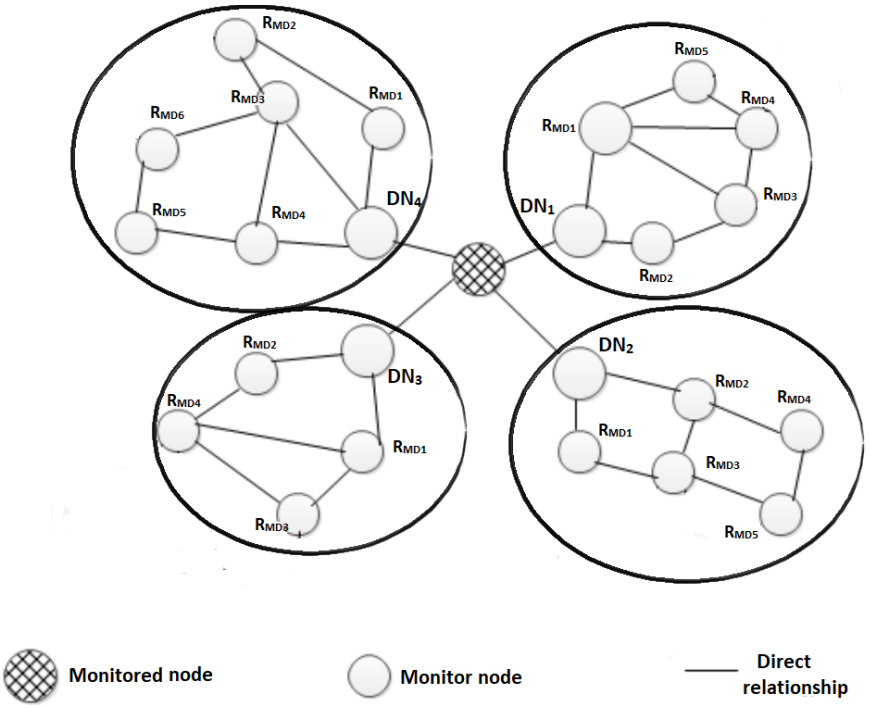


Figure 4.3: The honesty related to the indirect interactions

$Tm_{(MD_i)}$ is the local trust that node DN_j assigns to node MD_i. $Tm_{(MD_i)}$ is used to weigh the recommendation values with the trust level of their providers.

Secondly, the probability p_I that the monitored node has a good reputation in the network is evaluated as follows:

$$p_I = I_j * pnet \tag{4.6}$$

3) Cooperation: A node cooperates correctly in the network if it agrees to forward data and provide recommendations. To assess the cooperation of the monitored node, the monitor node DN_j calculates a cooperation rate C_j. The C_j is the number (nc) of the good replies of cooperation (c_i^+) done by the monitored node, divided by the total number (TNC) of recommendation requests and data forwarding requests sent by the DN_j to the monitored node:

$$C_j = \left(\frac{\sum_{i=1}^{nc}(c_i^+)}{TNC} \right) \tag{4.7}$$

We assume that the monitoring application of the DN_j generates cooperation requests from the monitored node according to a Poisson process with a rate $\lambda 1$.

The probability p_c that the monitored node cooperates in the network is:

$$p_c = C_j * p_{net} \tag{4.8}$$

4.4.3 Pseudo code of the estimation algorithm

The pseudo code of the estimation algorithm begins with computing the average avr of a set of values Var={var1,var2,...,varN} given as an argument to its estimate function (lines 3-6). Afterwards, the algorithm calculates the difference d between each value $var_h \, \epsilon \, Var$ and the average avr (lines 7-8). If this difference is greater than a threshold TH determined by the monitoring process, var_h will be considered as spam and will be deleted (lines 9-15). This operation avoids the spam values of Var that can distort the final estimation of trust. The algorithm then returns the number of values retained after eliminating spam.

Algorithm 2 The estimation algorithm

```
int Estimate(float var[ ], int N)
1.    float som=0;
2.    int cmp=0;
3.    FOR (int h=0; h< N; h++) DO
4.        som= som+ var[h];
5.    END FOR
6.    float avr=som/N;
7.    FOR (int h=0; h< N; h++) DO
8.        float d=abs(var[h]-avr);
9.        IF d> TH THEN
10.         var[h]=0;
11.         cmp=cmp+1;
12.       END IF
13.   END FOR
14.   FOR (int h=0; h< N; h++) DO
15.       Remove var[h]=0;
16.   END FOR
17.   Return N - cmp;
```

4.4.4 Probabilities of transition from one state to another

1) *Probabilities related to increasing or decreasing the Tm by one step*: We assume that the state of the Tm of the monitored node at time $t-1$ is s. If this node shows a trusted behavior in the network, its Tm will transit from state s to state $s+1$ at t. Otherwise, its Tm will transit to state $s-1$ at t according to the following equations:

$$P_{s,s+1}(t) = p_s(t-1) * p_c * p_d * p_I; \qquad 0 \leq s < M \tag{4.9}$$

$$P_{s,s-1}(t) = p_s(t-1) * (1-p_c) * (1-p_d) * p_I; \quad 0 < s \leq M \tag{4.10}$$

Where $p_s(t-1)$ is the probability that the monitored node is in state s at time $t-1$, p_c expresses the probability that the monitored node cooperate positively in the network, p_d expresses the probability that the monitored node is honest during its direct interactions, and p_I is the probability that defines the reputation of the monitored node in the IoT network. The computation of p_c, p_d and p_I was detailed in section 4.4.2.

2) *Probability to remain in the same state s with $s \neq M$:* If the monitored node doesn't have any interaction or cooperation to do, its Tm will maintain the same state and the same value for some time.
We assume that the monitoring application of the monitored node generates interactions or requests for cooperation according to a Poisson process with a rate $\lambda 1$, and receives interactions or requests for cooperation from other nodes according to a Poisson process with a rate $\lambda 2$ ($\lambda 1 \leq \lambda 2$). We also assume that an interaction or cooperation request requires a delay ts to be processed in the upper layer before being executed or sent. We consider that the values of $\lambda 1$, $\lambda 2$ and ts are the same for all nodes in the network. Hence, according to [101], the probability that the transmission queue is empty can be expressed as follows:

$$p_q = 1 - \left(\frac{1 - (1-\theta) * \theta^{\gamma}}{1 - \theta^{(\gamma+1)}} \right) \tag{4.11}$$

Where γ is the size of the transmission queue, and:

$$\theta = (\lambda 1 + \lambda 2) * ts \tag{4.12}$$

Then, the probability that the Tm of the monitored node remains in the same state s is defined as follows:

$$P_{s,s}(t) = p_s(t-1) * p_q \tag{4.13}$$

3) *Probability to sojourn in the trusted state M:* The Tm of the monitored node remains in the state M, where its value is equal to 1, either because it still shows trusted behavior in the network or it has no interactions or cooperation to do after reaching the higher trust state. Hence, the probability $P_{M,M}$ to maintain the trusted state M is:

$$P_{M,M}(t) = p_M(t-1) * (p_c * p_d * p_I + p_q) \tag{4.14}$$

4) *Transition to state 0:* State 0 corresponds to the untrusted state where the Tm of the monitored node is equal to 0 (Tm= 0). The Tm reaches this state if the monitored node act dishonestly in the network according to the following probability:

$$P_{s,0}(t) = p_s(t-1) * (1 - p_d) * p_I \tag{4.15}$$

Assuming that the initial state of the Tm of a node is $Y_0 = 1$, the probability for the Markov process to reach state z at time $t_z > 0$ is $P_z(t_z) = P_{1,z}(t_z)$. Furthermore, the probability to reach state s at time $t > t_z$ given that Y_{t_z} = z is $P_{z,s}(t)$. To compute the probability to be in state s at time t, $P_s(t)$, from the transition matrix P expressed in the equation. (4.1) , we use the Chapman-Kolmogorov equation [102] as follows:

$$P_s(t) = \sum_{z \in [1...M]} P_{1,z}(t_z) * P_{z,s}(t) \tag{4.16}$$

4.5 Performance evaluation

In this section, we use MATLAB R2015a to evaluate the resilience of our proposed trust management model to the dynamic behaviors of IoT nodes and to several trust-related attacks:

1) Bad-mouthing and ballot stuffing attacks: In bad-mouthing attacks, the malicious node destroys the reputation of a well-behaved node to isolate it from the network. In ballot stuffing attacks, the malicious node increases the reputation of another malicious node by providing it with good recommendations to increase its Tm.

Scenario: We suppose that the neighbor nodes of the monitor node gave respectively the following R_{ij} values: [0.5, 0.55, 0.1, 0.45, 0.55, 1]. By using the proposed estimation algorithm 2, the two extreme values 0.1 and 1 provided respectively by nodes 3 and 6 will be considered as spams. Indeed, these two values will not be taken into account in the calculation of the final rate I_j to avoid distorting the final value of the Tm. Afterwards, nodes 3 and 6 will be reported to the system as malicious nodes that aim to make bad-mouthing attack (for node 3) and ballot stuffing attack (for node 6). They will be the next monitored nodes.

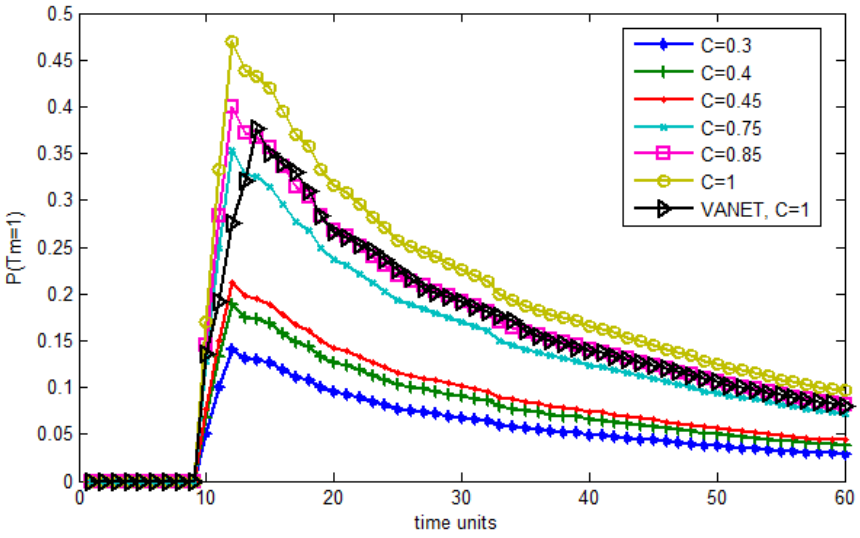


Figure 4.4: $p_d = 0.80$, $p_I = 0.95$

2) Attacks of honesty: In IoT networks, some nodes increase their Tm by doing too much cooperation, while many of their cooperation are dishonest. These malicious nodes only cooperate honestly for non-critical interactions and change their behavior for critical and important cooperation. Indeed, they cooperate just to send untrusted services or bad recommendations, to redirect nodes to malicious service providers, or to disclose the sensitive data received from other nodes.

Scenario: To prove the resilience of our trust management model to the attacks of honesty, we use the Markov chain defined in Fig. 4.1 with $M = 10$ states, each state refers to a step ϕ= 0.1. We set $T0$ for each node to 0.1 and p_q to 0.25 as computed in [99]. p_{net} is specific to each pair of nodes, to simplify we assume that it is equal to 0.95 in all network. We notice from Fig. 4.4, Fig. 4.5 and Fig. 4.6 that the probability to reach $Tm = 1$ noted by $P(Tm = 1)$ is equal to 0 for the first 10 time units. This is due to the fact that the Tm of the monitored node is initialized to 0.1 and it has to pass by all states from 1 to $M = 10$ to get 1. Afterwards, this probability constantly increases up to obtaining its maximum in the 14 th time unit.

In Fig. 4.4, for C=1, p_d=0.8 and p_I=0.95, the monitored node fully cooperates in the network but just 80 % of its cooperation are honest. According to Fig. 4.4, we notice that once $P(Tm = 1)$ reaches its maximum, it begins to be attenuated. This attenuation is due to the cumulative effect of p_d. Indeed, despite the fact that the monitored node cooperates fully in the network, 20 % of its cooperation are untrusted and can harm the basic functionality of the IoT network.

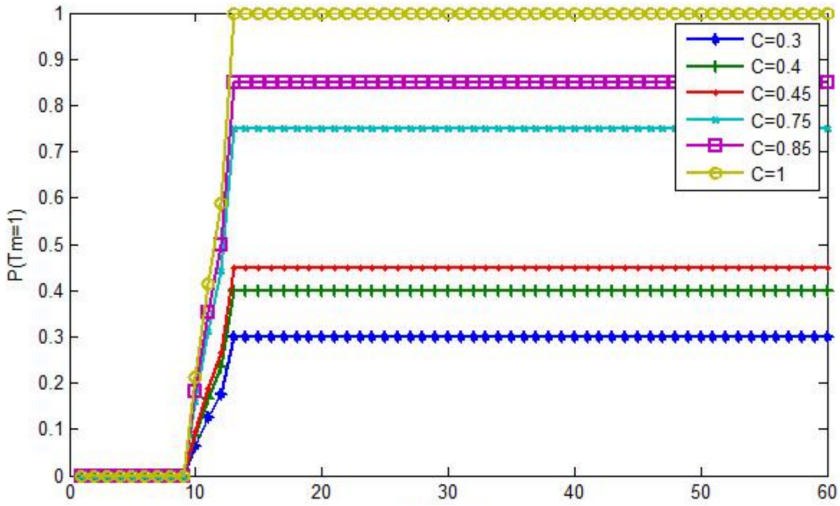


Figure 4.5: $p_d = 0.95$, $p_I = 0.95$

As long as the monitored node does not enhance the honesty related to its direct interactions p_d, its probability of getting $Tm = 1$ will only decrease. Whereas, in Fig. 4.5 where $p_d = 0.95$ and C=1, once $P(Tm = 1)$ reaches the maximum, it does not attenuate as the preceding case where $p_d < 0.95$, because the monitored node fully cooperates and its cooperation are honest at 100 %.

Furthermore, we remark in Fig. 4.4 and Fig. 4.5 that for higher values of C (0.75, 0.85 and 1) the $P(Tm = 1)$ is more significant compared to small values of C (0.3, 0.4 and 0.45). Therefore, when the monitored node reaches the trusted state and continues to behave correctly in the network, it remains in this higher state.

In Fig. 4.6, we suppose that the monitored node changes the honesty related to its indirect interactions from 0.95 to 0.6. According to Fig. 4.6, we notice that the values of $P(Tm = 1)$ decrease compared to the previous case in Fig. 4.4 where p_I=0.95. We deduce that the evaluation of the trust in our model is sensitive to both parameters p_d and p_I.

We compare our proposed trust management model with the trust management model proposed for VANETs in [99], in which the variation of Tm is also modeled by a discrete-time Markov chain with 10 states. The trust management model proposed in [99] only evaluates the honesty related to the direct interactions of vehicles when forwarding data. However, this model is insensitive to the variations of the honesty related to indirect interactions of vehicles. According to the results of Fig. 4.6, despite the fact that the p_I of the monitored vehicle decreases, the $P(Tm = 1)$ keeps its old value already obtained in Fig. 4.4.

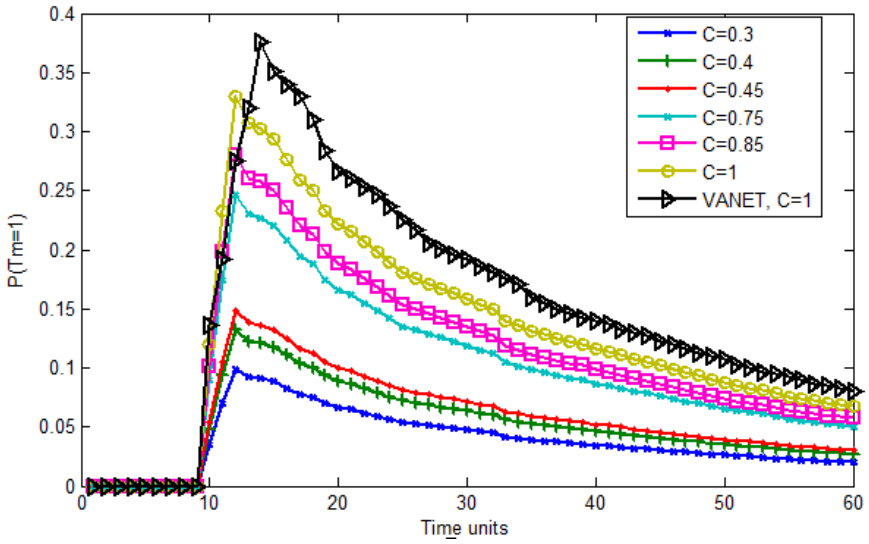


Figure 4.6: $p_d = 0.8$, $p_I = 0.6$

The results obtained show that our trust management model is more robust to the attacks of honesty than the model proposed for VANETs in [99].

3) Selfish attacks: These attacks are launched by uncooperative nodes that preserve their resources and energy only to perform their own tasks. Indeed, a selfish node is not intended to carry out malicious attacks in the network, but rather is a node that acts solely for its own interest.

Scenario: We assume that once the monitored node achieves the maximum level of trust, it will behave selfishly in the network by degrading its cooperation rate C. During the first 20 time units, the node behaves positively in the network (p_d=0.95, p_I=0.95 and C=1) to increase its Tm. Afterwards, the monitored node changes its behavior where it reduces its cooperation rate C during 10 time units followed by a positive behavior during 10 time units and so forth, as shown in Fig. 4.7.

When the monitored node reduces its cooperation rate in the 21 th time unit, the $P(Tm = 1)$ decreases severely as shown is Fig. 4.7. This probability returns to its previous value once the node resumes its normal behavior. $P(Tm = 1)$ increases slowly during the phase when the node reduces its cooperation rate, because selfish nodes do not act maliciously in the network $p_d = 0.95$ and $p_I = 0.95$. We notice that our proposed trust management model is robust to the dynamic behaviors of IoT nodes.

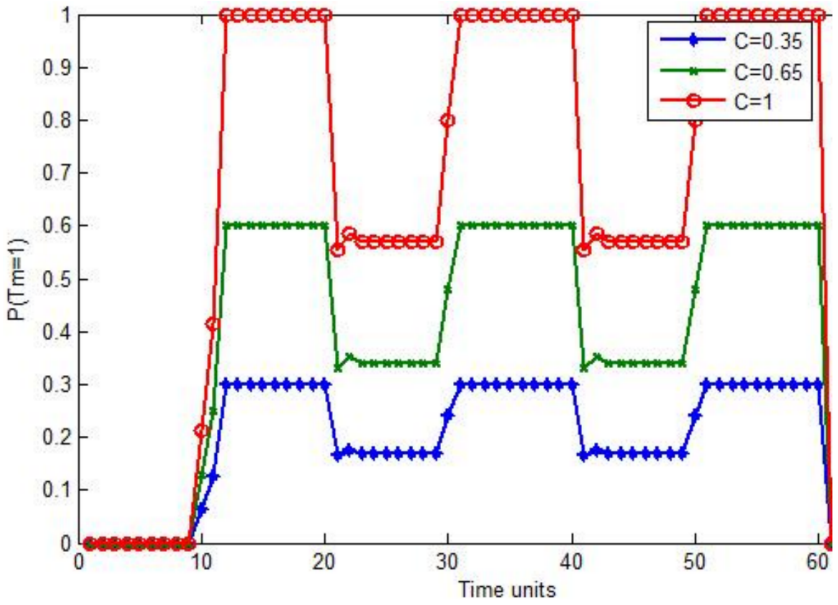


Figure 4.7: $p_d = 0.95$, $p_I = 0.95$

4.6 Conclusion

In this chapter, we proposed a distributed analytical trust management model in order to enhance the security and reliability of IoT networks. The proposed model is hybrid, it relies on the cooperative behavior of IoT devices and the legitimacy of the information they provide. The number of transitions to achieve the high level of trust as well as the trust interval in the proposed model can be adapted according to the monitoring process and network characteristics. This makes our trust management model flexible and adaptable. The numerical results illustrated the strong resilience of our trust management model to the dynamic behaviors of IoT nodes and to various popular trust-related attacks.

In the next chapter, we focus on trust management in IIoT networks; considered as the result of the integration of IoT in industrial processes.

Chapter 5

Trust Management in IIoT networks

Contents

5.1 Introduction **66**

5.2 Related work **67**

5.3 Future automotive factory architecture based on IIoT . **70**

5.4 The proposed trust management model analysis **72**

5.4.1 Phase I: Designation process of CL 73

5.4.2 Phase II: The community establishment method 77

5.4.3 Phase III: The proposed trust management model 80

5.5 Performance evaluation **83**

5.5.1 Part 1: Comparative energy study 84

5.5.2 Part 2: The sensitivity, the responsiveness and the resiliency of our proposed trust management model to trust-related attacks launched by malicious IIoT nodes. 89

5.6 Conclusion **97**

5.1 Introduction

To accompany the industrial revolution 4.0 ([103], [104], [105]), all automobile manufacturers will rely heavily on the IIoT in order to change the traditional architecture of their plants into a fully automated and highly connected architecture. IIoT networks [120] refer to industrial devices (sensors, production machines, robots, etc.) connected to wireless networks, and which collect and share data on their environments. IIoT devices will be deployed in all industrial services and processes: manufacturing, maintenance, monitoring and other tasks, which makes them increasingly attractive targets for security threats.
The integration of security measures [106] [107] into IIoT devices is very important but it needs to be reinforced by securing the entire IIoT network of the plant. Indeed, an IIoT network can behave in untrustworthy manner even after implementing all necessary security and confidentiality measures in its IIoT devices ([108], [109], [110]) as the Stuxnet worm attack [111] on Iran's nuclear installations showed in 2010. This attack destroyed a large number of centrifuges in the Natanz power station by slightly disrupting their operations. Hence, behavior-based analysis of IIoT devices is required to predict their performances over time.
Trust management can provide behavior-based analysis of IIoT devices by using their past behavior and their reputation in the network. Usually, trust is managed either in a distributed manner, where each device evaluates the trust metrics Tm of its neighbors, or in a centralized manner, where a single entity manages the trust of the entire IIoT network. However, a single centralized trust management entity is not able to continuously and accurately manage the trust in the plant's IIoT network composed of a large number of heterogeneous and sensitive devices. Admittedly, distributed trust management models are the most widely used in the composition of services in IoT, but, they are not appropriate to manage the trust in industrial environments. The ultimate solution is to change the traditional architecture of the plant's IIoT network by organizing IIoT devices into clusters in order to support and enhance the trust management process. In SIoT [27], devices build their clusters according to their common social relationships [112]. However, social relationships cannot be established between IIoT devices. Hence, new specific relationships more appropriate for industrial environments must be defined.
Inspired from SIoT, our contribution in this chapter is three fold. In the first contribution, we define a new concept called industrial relationships where IIoT devices are able to establish industrial relations between them. Instead of traditional architecture of the plant's IIoT networks, the industrial relation-

ships define a new hierarchical architecture called H-IIoT. Indeed, the H-IIoT architecture is constituted of clusters called industrial communities. Within the same community, the evaluation of trust becomes more accurate because it is calculated and defined in the same context. Each community is monitored by a trusted leader which evaluates the trust of the member devices and returns the results to the IIoT server to record them. The trust is evaluated according to three parameters namely, cooperation, honesty related to direct interactions and honesty related to indirect interactions. In the second contribution of this chapter, we use the three parameters to propose a dynamic trust management model called Tm-IIoT model suitable for IIoT networks. In the last contribution, we implement our proposed H-IIoT architecture by using contiki/cooja simulator to compare its efficiency with that of the traditional architecture of the plant's IIoT network. We also demonstrate the sensitivity, responsiveness and resiliency of our proposed trust management model to trust-related attacks launched by malicious IIoT devices.

The rest of this chapter is structured as follows. To the best of our knowledge, at the time we started our research on trust management in IIoT, the related work does not exist in this research area. Hence, in section 5.2, we discuss some related trust management models proposed for IoT to prove that these models are not appropriate for IIoT networks. The purpose of this discussion is to justify the need for a new trust management model that considers the requirements and the constraints of IIoT networks. In section 5.3, we describe the applications of IIoT in the future connected factories. This defines the need for IIoT networks in terms of security and trust management. In section 5.4, we rely on a new concept called industrial relationships to propose new hierarchical architecture for IIoT networks. Indeed, the proposed architecture supports the trust management in IIoT networks. In section 5.4, we propose a dynamic trust management model that considers the requirements and constraints of IIoT networks. We present the performance evaluation, the comparisons and the simulation results in section 5.5. Finally, in section 5.6, we conclude the chapter.

5.2 Related work

The use of trust management to monitor the behavior of IIoT devices in factories was not considered as a main concern by researchers. To the best of our knowledge, there existed no trust management model proposed specifically for IIoT networks. Thus, in this section we analyze the trust management mod-

els proposed for IoT and SIoT networks to investigate the possibility of their applications in an IIoT context.
To make decision about the trustworthiness of an IIoT device, the IIoT server must have the global Tm of that device.
In distributed trust management models such as [80], [113], [114], [115], [116], [12] and [18] each IoT node calculates the trust metrics of its neighbors and stores them locally in order to use them for its own interests: service composition [116], decision making [117] or access control [118]. In order to have the global trust metric of a node, it is necessary to gather all local trust metrics calculated for that node. This process is very demanding in terms of energy consumption and also in terms of time wasted to have the final convergence trust metrics. In centralized trust management models such as [80], [12] and [119], just one trust management entity must monitor, alone, the trust of all IIoT nodes in the network. These models are very difficult to apply especially when the IIoT network evolves. Therefore, given the above discussion, whether centralized or distributed trust management models cannot be applied to manage the trust in IIoT networks.
The ultimate solution is to combine the two types of models in order to have a hierarchical trust management model based on clustering. Authors of [121] proposed a scalable hierarchical trust management solution for IoT environments based on clustering. The trust metric of each cluster node is managed by a master node and collected from the peer cluster nodes. The work in [121] proposed an algorithm to eliminate outliers, hence, the resulting trust metric is calculated as the average of the remaining evaluations. However, this work is vulnerable to coalition attacks because the proposed algorithm makes its decisions based on the evaluations given by the majority of cluster nodes. If the proposed algorithm has as a majority the evaluations given by malicious cluster nodes, then it eliminates the evaluations given by legitimate and fair nodes. Another limitation of [121], is the lack of accuracy in the calculation of trust values as the evaluations are not carried out in a well-defined context. In fact, it is required to have a context between the monitored and the monitor nodes in order to have a high accuracy when managing the trust.
Authors of [122] proposed a clustering architecture based on the similarity of interest and the relationships between devices to create a trust management context. By analyzing this work, in order to construct the communities of interest, authors considered the social relationship between owners instead of considering the social relationships between devices. Indeed, the choice of belonging to a particular community is made by human. Therefore the devices are not considered autonomous, which contradicts the objectives of SIoT con-

cept. To elect the leader of a community, the authors of [122] considered the following metrics: trust level, capability and scalability. Scalability promotes nodes with a large number of friends to be elected as a leader. This makes the model highly vulnerable to coalition attacks where a set of malicious nodes can meet between them and elect an untrusted leader.
Authors of [116] proposed a trust management protocol to support service composition in SOA-based IoT systems. They developed a technique based on distributed collaborative filtering to select feedback from owners of IoT nodes sharing similar social interests. To measuring social similarity, the authors relied on three social relationships, i.e., friendship, social contact and community of interest. To compute similarity rates, nodes exchange their profiles containing the friend list and the location list of their owners in clear text, which does not preserve the privacy of users and allows their traceability. Authors of [112] proposed a new protocol based on three trust factors: honesty, cooperativeness and community interest. The authors of [112] calculated the cooperativeness as the ration of the number of common friends over the total number of friends between two nodes. The calculation of the cooperativeness factor in this work is very subjective and the fact that two nodes are friends does not really reflect their willingness to cooperate together [123]. The biggest limitations of this work are the energy efficiency and the adaptability of the protocol. To improve this work, authors of [124] reused the same trust factors as [112] and take into consideration other aspects such as scalability and adaptability of the protocol. As in [112], the update of trust metrics is always events-driven and the trust metrics are computed only for a limited set of nodes to minimize computation and to ensure scalability. However, the proposed trust management protocol in [124] does not detect On-off attacks.
The proposed work in [19] builds a reliable trust management model based on the behavior of IoT nodes. Hence, each node in the network computes the trust metrics of its friends based on its own experience and on the opinion of its common friends. The basic trust parameters used for calculating the trust metric in this work are: Feedback system, transaction factor, total number of transactions, relationship factor, computation capability, credibility and centrality. This work is scalable and resilient against self promoting, bad-mouthing and ballot-stuffing attacks but it does not ensure the power efficiency and survivability [123].
The trust management protocols proposed in [112], [124] and [19] are very greedy in terms of energy consumption. They are based on the change of state of IoT nodes (monitored monitor and vice versa) which can cause a fast drainage of their battery. In these works, there is no differentiation between

nodes i.e. any node can be a monitor even small sensors with constrained resources. Furthermore, the evaluation of trust metrics in these works is very subjective. Indeed, each node calculates the trust metric of its neighbors according to its own context, stores them locally and uses them in case it needs to interact with other nodes in the network.
Several works used blockchain technology [125] [126] to propose secure trust management models for IoT. Indeed, this technology concerns distributed trust management models that need to securely store and exchange trust scores between nodes within the network. Blockchain technology cannot be used to manage trust in the plant's IIoT network because distributed trust management models are not appropriate to manage the trust in industrial environments for all the reasons explained above.
In the previous chapter 4, we proposed a distributed trust management model for IoT networks based on three trust metrics namely, cooperation, honesty related to direct interactions and honesty related to indirect interactions. However, distributed models cannot be used to properly manage the trust in IIoT networks. In the light of the existing trust management models, we propose in this chapter an appropriate trust management model for IIoT networks, which is based on a hierarchical architecture. We use the same trust parameters as in chapter 4, but these trust parameters will be evaluated differently according to the IIoT context as shown in section 5.4.
What differentiates IIoT from IoT is first of all the application context. Therefore, trust management in IIoT networks must take into consideration the industrial context. Also, IIoT devices are more heterogeneous. Indeed, in IoT networks any node can manage another, however in IIoT the monitor node must prove at least the same technological capabilities as the monitored node in order to predict its behavior correctly over time. Moreover, some IIoT devices manage sensitive data related to industrial secrecy, which requires customized trust management for each node depending on the sensitivity of the data it manages. For all these reasons, we would like to mention that the core idea of the two chapters is totally different.

5.3 Future automotive factory architecture based on IIoT

Many automobile manufacturers [1] aim to make everything in their future plants smart, autonomous and connected. For example, manufacturers will

[1] https://volkswagen-newsroom.com/en/stories/industry-40-we-make-it-happen-4779

rely on IIoT to better manage the plant's inventory. Indeed, smart devices will monitor the stock availability and will send a report to the inventory management system, which will automatically communicate with suppliers to order necessary parts.
If there is complicated maintenance to be done, operators can easily contact the experts by using their smart devices (i.e. connected tablets, watches, etc) to intervene remotely. They can also use their connected glasses to follow the expert's constructions in order to update the production machines in the plant. The connected glasses can also allow the expert to follow all the manipulations of the operators and to intervene if necessary. IIoT devices can provide detailed information about their functioning (i.e. unusual operations, alerts, potential problems, etc.), to predict failures earlier and more accurately. This IIoT-based predictive maintenance allows the manufacturers to reduce downtime and increase their efficiency.
In the plant, there are also connected robots that support smart production and send report about the number of vehicles produced per time period. Even after its delivery, the vehicle remains connected with the after-sales service to send periodic reports about the technical condition of engine and parts. In case of problem, the customer will be notified to do the necessary maintenance. This will allow the automobile manufacturer to improve both its after-sales service and its choice of suppliers. The smoke sensors of the plant will be connected to the fire department, once the temperature exceeds the limits, the water-filled pipes will be made available throughout the plant and a report will be sent back to the fire department to intervene.
In the automotive plant, connected meters will be installed everywhere to measure the energy consumed by machines, production lines and installations in real time. Thanks to these measures, managers can improve their understanding of energy consumption in the plant, determine what is consuming the most energy and identify inefficient machines.
As explained above, automotive manufacturers intend to rely on IIoT in all industrial services and processes. Indeed, IIoT devices will lead to manage sensitive data, which will make them a very attractive target for hackers that aim to compromise the reliability of the IIoT network. Hence, it is necessary to monitor the plant's IIoT network continuously by using trust management. To support trust management in the IIoT network, we organize the plant's IIoT network by defining a new concept called industrial relationships where devices can establish industrial relations between them. The industrial relationships used in our contribution are defined as follows:

- Parental relationship (PR): exists between devices which have the same technological characteristics and capabilities in the network.
- Co-worker relationship (CWR): binds devices that perform the same tasks or that always collaborate together to provide intelligent services.
- Ownership relationship (OR): exists between devices belonging to the same employee or the same production line.
- Social relationship (SR): is defined by the social relationship (i.e. hierarchy relations, collaboration relations, etc.) that exist between the owners of the IIoT devices.

The monitor node easily detects the behavior changes and the untrusted behavior of a monitored node when they are linked by the CWR relationship since they work together to achieve the same objective. In order to have more accurate Tm and more homogeneous clusters, the CWR relationship can be strengthened by OR and SR relationships. The evaluation of trust becomes more accurate when the monitor and the monitored nodes are linked by several industrial relationships.

5.4 The proposed trust management model analysis

As discussed above, we aim to change the traditional architecture of IIoT networks in automotive plants in order to support trust management. Hence, we organize the IIoT network as a set of clusters called industrial communities. Clusters are not necessarily the same size.

As depicted in Fig. 5.1, the IIoT network is composed of K community leaders and L member nodes. L is not the number of member nodes in each community, but rather the total number of member nodes in the network. Let $CL_j|j \in \{1,2,\ldots,K\}$ and $MN_i|i \in \{1,2,\ldots,L\}$ represent the set of community leaders and member nodes in the network, respectively.

The leader CL_j manages the Tm of each MN_i belonging to its community by calculating three trust parameters: cooperation, honesty related to direct interactions and honesty related to indirect interactions. CL is a specific node that must have high level of trust, sufficient computing power, storage and energy resources to perform monitoring tasks.
Each node that wants to join the network must send a request to the IIoT server. IIoT server assigns to nodes a unique identifier "ID" and creates a

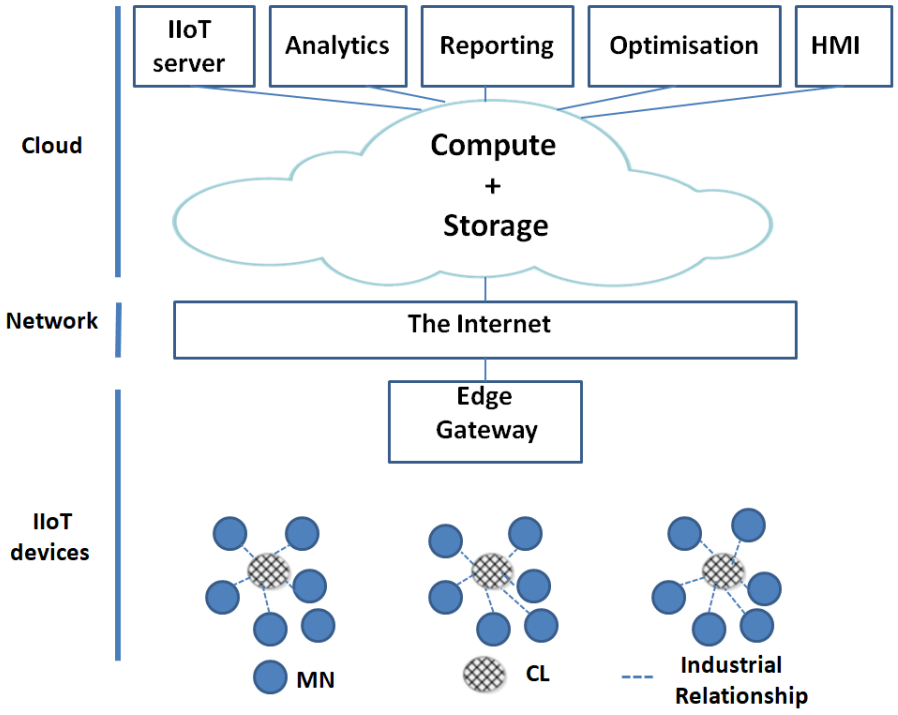


Figure 5.1: Proposed H-IIoT architecture

profile for them. The profile is stored in both the IIoT node and the IIoT server. Indeed, the profile sets the tasks that can be performed by the IIoT node (such as measurements, transactions, operations, etc.) and the data that can be shared by each node in the network. It determines also the types of the industrial relationships that can be established between two nodes. Based on these defined rules, each node can allow to start, update or terminate any relationships with another node.

Each node that wants to be a CL must request authorization grant from the IIoT server. According to the conditions explained below in phase 5.4.1, the server gives or not the permission to a specific node to be leader. Once a node becomes a CL, it establishes its community.

The following section is organized into three phases. The first phase concerns the designation of CLs based on several conditions. The second phase describes the establishment of communities around the designated CLs based on the industrial relationships. The last phase concerns the monitoring process where CL_j manages the Tm of its community members in order to detect suspicious and untrusted behaviors.

5.4.1 Phase I: Designation process of CL

CL contacts the IIoT server either to send the results of its monitoring or to retrieve data needed to evaluate the Tm of its community members. CL

evaluates the Tms of its MNs according to their behaviors in the network and according to the monitoring frequency defined for each MN.
The monitoring frequency is customized depending on the sensitivity of the data managed by the monitored MN. Indeed, IIoT nodes containing critical and production-sensitive data must be continuously monitored with a higher frequency compared to nodes containing less critical and less sensitive data. The monitoring frequency of a MN_i does not only depend on its nature (critical or not critical) but also on the community parameters: the number Ds of nodes which must be monitored by the same trust management entity and the period Tr for which Tms of all IIoT nodes in the community must be updated and registered in the IIoT server.

The monitoring frequency for each MN_i will be calculated as follows:

$$fi = \kappa fi_r + (1 - \kappa) fi_N \tag{5.1}$$

Where: fi_r is the recommended frequency for each node according to the sensitivity of the data it manages. fi_N is the recommended frequency by the network and the community parameters. It is the frequency that must be respected to ensure that the Tm of all nodes in the IIoT network will be managed within a fixed period Tr. It is calculated as follows:

$$fi_N = \left(\frac{Ds}{Tr}\right) \tag{5.2}$$

The selection of $(0 \leq \kappa \leq 1)$ is important for setting the order of monitoring. κ is used to weigh the importance of the sensitive data relative to the network parameters in the evaluation of the monitoring frequency. The monitoring process sets the parameter κ to a high value in order to ensure continuous monitoring of critical nodes and to have a better accuracy when calculating the Tms. The parameter κ is set to a smaller value when the monitoring process aims to monitor all nodes in a very specific time period "Tr", regardless of their sensitivity.

Each CL_j will have to continuously monitor the behavior of the MN_i in its community. For this purpose, it compares the current behavior of its member nodes with their profiling stored in the IIoT server. As discussed above, the CL is a specific node, it must have a high level of trust to manage its members, it must have also a sufficient reserved energy to perform monitoring operations and to communicate with the server, and it must have a sufficient computing power to calculate cooperation and honesty rates necessary for the evaluation of Tms.

Before the implementation of our proposed hierarchical architecture (designation of CLs and construction of communities), the plant's IIoT network will be deployed to allow nodes to acquire knowledge about their neighbors. In this deployment phase, we adopt a centralized trust management architecture where the IIoT server is the only entity in the network responsible for managing the Tms of all nodes.
Each node has an initial trust metric $T0$. When the Tm of a node reaches a high level of trust, the server asks that node to calculate its designation indicator DI according to equation (5.3) and return the result of this calculation to it. None of these queried nodes can report false data (e.g., a high DI value), since they are legitimate nodes. These nodes will probably be designated as a leader if their designation indicator DI checks some conditions explained later in this first phase. Only nodes with high Tm will be contacted by the server to calculate and send their DI. Indeed, the CL before any other condition, it must be a legitimate node.
If a node does not send its designation indicator DI, it will be penalized and judged by the server as a selfish node because it does not participate in the monitoring process. The designation indicator DI is calculated as a follows:

$$DI = \zeta Tm + \beta Er + \varrho Pm + \omega Cm \tag{5.3}$$

Where Er is the energy reserved for monitoring, we assume that each node has a reserved energy just for monitoring. Pm is the computing power level specific for each node and Cm designates the centrality of a node in the network. The weight $\zeta + \beta + \varrho + \omega = 1$ and $0 \leq \zeta, \beta, \varrho, \omega \leq 1$. In practice, the monitoring process determined by the IIoT server gives weight (ζ, β, ϱ, ω) to each component (Tm, Er, Pm, Cm) according to its importance and according to the network conditions. Indeed, a CL must always have a high level of trust since it must monitor the Tms of other nodes in the network. Hence, the weight ζ must always be higher than the other parameters. If the IIoT network is very dense, in addition to the high level of Tm, the CL must also prove a high computing capacity Pm and sufficient monitoring energy Er to perform the monitoring tasks. Otherwise, if the IIoT network is less dense with isolated nodes, the CL must be chosen based on its Tm and also based on its centrality to avoid being placed in the extremities of the network and building empty communities.

The server checks all received DI and decides for each node whether to give it the authorization to be a CL. Only nodes with a DI greater than a threshold set by the monitoring process take the authorization to announce

themselves as CLs. The threshold value reflects the type of nodes deployed in the network. Indeed, if the network contains critical, sensitive and very heterogeneous nodes, the threshold value should be set to a high value in order to require more selectivity in the designation and the choice of CLs. Once approved by the server, the designated CL broadcast beacons to all other nodes within its transmission range to announce itself as a leader.
If two CLs are close to each other, it is better to eliminate one of them to optimize the number of CLs and the number of communities formed, especially if the network is not dense as explained in Algorithm 3.

Algorithm 3 The designation algorithm

CL_j is a designated leader;
$CL_{j'}$ is a new designated or an already designated leader, where $j \neq j'$
when CL_j receives a beacon from $CL_{j'}$;
begin
1. **if** DI($CL_{j'}$) < Threshold **then**
2. **Reject beacon**;
3. Send the feedback to the server; **GO TO** (End)
4. **else**
5. **if** ($dij < D$) **then**
6. **if**(DI(CL_j)>DI($CL_{j'}$))
7. Status(CL_j)=CL;
8. Status($CL_{j'}$)=MN;
9. Send the feedback to the server;
10. $CL_{j'}$ stops broadcasting beacons to other nodes and CL_j continues to send them;
11. **else if**(DI(CL_j)<DI($CL_{j'}$)) **then**
12. Status($CL_{j'}$)=CL;
13. Status(CL_j)=MN;
14. Send the feedback to the server;
15. CL_j stops broadcasting beacons to other nodes and $CL_{j'}$ continues to send them;
16. **End if**
17. **else if** ($dij > D$) **then**
18. Status($CL_{j'}$)=CL;
19. Status(CL_j)=CL;
20. Send the feedback to the server;
21. The two nodes continue to send beacons to their communities respectively;
22. **End if**
23. **End if**
24. **End**

The value of D is chosen according to the density of the IIoT network. D is chosen small when the network is dense to increase the number of CLs. If the network is dense and the value of D chosen is large, it will cause congestion at the CL nodes and a quick drainage of their batteries. When the network is not dense, the distance D must be large to optimize the number of CLs in the network. Choosing a small D while the network is not dense will constitute communities with very few member nodes or even empty. Each CL_j sends

a periodic beacons to its community members. As soon as a member does not receive any more beacons, it will consult its CL preference list to choose another CL. When an MN_i receives a beacon from a new CL, it runs the community establishment algorithm that will be introduced later in the next phase, either to join the community of this new CL or to add this CL to its preference list.

5.4.2 Phase II: The community establishment method

The purpose of this phase is to form communities around the designated CL_j based on the industrial relationships. Therefore, it is assumed that each MN_i will choose to join community of one and only one CL_j and each CL_j will monitor one and only one community.

MN_i computes the selection indicator SI_{CL_j} for each CL_j from which it has received beacons and it chooses the CL_j having the maximum SI_{CL_j}. SI_{CL_j} is calculated based on the physical distance, the quality of link and the industrial relationships between the two nodes as follows:

$$\begin{aligned} SI_{CL_j} =& \delta(1 - \left(\frac{d(MN_i,CL_j)}{max_{CL_j}(d(MN_i,CL_j))}\right)) \\ &+ \xi(1 - BER_{CL_j}) \\ &+ \eta(RI_{CL_j}) \end{aligned} \tag{5.4}$$

Where $\delta+\xi+\eta = 1$ and $0 \leq \delta, \xi, \eta \leq 1$, $d(MN_i,CL_j)$ is the physical distance between MN_i and CL_j. When a node receives a signal from its adjacent node, it uses the power of this received signal to calculate the distance between them. In our work we calculate the distance between two nodes based on the number of hops between them. It is assumed that IIoT nodes already have an appropriate identity management system. MN_i has an interest to choose a leader close to it in order to save its energy and to avoid retransmissions and data losses. SI_{CL_j} is unitless, its value ranges from 0 to 1. $max_{CL_j}(d(MN_i,CL_j))$ is the maximum physical distance that can be between MN_i and CL_j nodes, it is used to normalize the SI_{CL_j}.
The bit error rate (BER) is the number of bit errors divided by the total number of transferred bits during a studied time interval. BER is a unitless, it is defined as following:

$$BER_{CL_j} = \left(\frac{The\ number\ of\ bit\ errors}{The\ total\ number\ of\ transferred\ bits}\right) \tag{5.5}$$

The relationship indicator RI_{CL_j} evaluates the industrial proximity of MN_i with CL_j based on the industrial relationships between the two nodes. It is defined as follows:

$$\begin{aligned} RI_{CL_j} =& \Gamma(PR(MN_i,CL_j)) + \Theta(CWR(MN_i,CL_j)) \\ &+ \rho(OR(MN_i,CL_j)) + \nu(SR(MN_i,CL_j)) \end{aligned} \tag{5.6}$$

Where $\Gamma + \Theta + \rho + \nu = 1$ and $0 \leq \Gamma, \Theta, \rho, \nu \leq 1$, each node sets its own weights conforming to its profile.

Once a node is designated as a leader, it starts sending beacons to nodes in its transmission range to announce its presence. The clustering is done at one hop. MN_i can receive periodic beacons from one or more CL_j nodes. The beacons contain a set of information about the CL_j sending them such as the manufacturer code (that reflects the technological capabilities of the node), owner code (the code of the production line or the user who owns the node), task list (that contains all the tasks that a node must perform in the network) and social friend list (defined according to the social relations of the user of the node (i.e. hierarchical relations, collaborative relations, etc.)). These information are already predefined in the profile of each node. They are sent hashed to MN_i nodes by the CL_j nodes, because they contain the technological and industrial data of the CL_j that sends them. MN_i uses the received beacons to evaluate its industrial relationships with the CL_j as a follows:

- Parental relationship $PR(MN_i,CL_j)$: In the heterogeneous IIoT network, there are nodes that require specific abilities from their monitor nodes to accurately monitor their behavior and assess their Tm. For example, a connected production robot cannot be monitored by a simple sensor. Indeed, the sensor does not have sufficient technical resources to accurately evaluate the Tm of the connected production robot. Hence, MN_i will privilege parental relationship in order to have approximately the same technological capabilities as its leader. To evaluate this relationship, MN_i compares its manufacturer code, that designates its technical capacity, with the manufacturer code sent in the beacon of the CL_j. If the two codes are similar or the code of the leader is larger, the value of $PR(MN_i,CL_j)$ takes 1 otherwise it takes 0.

- Ownership relationship $OR(MN_i,CL_j)$**:** During the monitoring process, CL_j asks the MN_j nodes for a set of information about their operations. In automotive plants, IIoT nodes contain critical and production-sensitive data.

Hence, critical MN_i chooses its leader based on this relationship to restrict the circulation of its monitoring data just between the nodes belonging to the same production line or user as it. To evaluate this relationship, MN_i compares its owner code with the owner code sent in the beacon of the CL_j. If the two codes are similar, the value of $OR(MN_i,CL_j)$ takes 1 otherwise it takes 0.

- Co-worker relationship $CWR(MN_i,CL_j)$**:** This relationship makes the calculation of trust more accurate because the Tm of MN_i will be evaluated by a leader that performs the same tasks. To evaluate this relationship, MN_i calculates the similarity between its task list $A(MN_i)$ and the list $A(CL_j)$ that contains the set of hashed tasks of node CL_j as follows:

$$CWR(MN_i,CL_j) = \left(\frac{A(MN_i) \cap A(CL_j)}{A(MN_i) \cup A(CL_i)}\right) \tag{5.7}$$

- Social relationship $SR(MN_i,CL_j)$**:** is chosen when the MN_i nodes are based on the social relations of their users (i.e. hierarchy relations, collaboration relations, etc.) to choose their leaders. To evaluate this relationship, MN_i calculates the similarity between the list of its social friends $S(MN_i)$ and the list $S(CL_j)$ that contains the set of hashed social friends of the CL_j as follows:

$$SR(MN_i,CL_j) = \left(\frac{S(MN_i) \cap S(CL_j)}{S(MN_i) \cup S(CL_j)}\right) \tag{5.8}$$

We propose the algorithm 4 to establish communities between nodes CL_j and MN_i as follows:

Each MN_i has a preference list where it stores the identities of its favorite leaders. When MN_i receives beacon from a $CL_{j'}$ other than its leader, MN_i calculates the selection indicator $SI_{CL_{j'}}$ of $CL_{j'}$ and stores its identity in the preference list. $CL_{j'}$ nodes are ranked in the list in descending order of their selection indicators. The leader with the largest selection indicator will be placed at the top of the preference list. If the list is full, the MN_i compares the calculated $SI_{CL_{j'}}$ with the selection indicator of the last node in the preference list. If the calculated $SI_{CL_{j'}}$ is greater, MN_i removes the last leader from the list and adds the $CL_{j'}$ which will be ranked in the list according to its selection indicator value. MN_i can join the community of leaders stored in its preference list in case it no longer receives beacons from its own leader.

CLs monitor the MNs belongings to their communities and return the results to the IIoT server. The IIoT server ensures that all nodes in the network are monitored. When a set of nodes does not exist among the monitored nodes,

Algorithm 4 The community establishment Algorithm

Input: $CL_j|j \in \{1,2,\ldots,K\}$ and $MN_i|i \in \{1,2,\ldots,L\}$;
Output: Industrial Communities;
Begin
1. **FOR** each MN_i
2. Set the coefficients δ, ξ, η, Γ, Θ, ρ, ν;
3. **END FOR**
4. **FOR** each CL_j
5. Compute the initial community $CM_j = \{MN_i, d(MN_i,CL_j) \leq transmission\ range\}$;
6. **END FOR**
7. **FOR** each MN_i
8. **FOR** each CL_j
9. **if** $MN_i \in CM_j$
10. Compute $d(MN_i,CL_j)$, BER_{CL_j}, $PR(MN_i,CL_j)$, $CWR(MN_i,CL_j)$, $OR(MN_i,CL_j)$, $SR(MN_i,CL_j)$;
11. Compute SI_{CL_j};
12. **END If**
13. **END FOR**
14. Choose CL_j with max SI_{CL_j};
15. **END FOR**
16. **FOR** each CL_j
17. Form the Industrial community CM_j;
18. **END FOR**
19. Return the Industrial Communities $CM_j|j \in \{1,2,\ldots,K\}$;
End

it means that these nodes are isolated and do not have a CL to manage them. Therefore, the IIoT server gives permission to a node among these isolated nodes to be a CL.

5.4.3 Phase III: The proposed trust management model

In this phase, we propose the Tm-IIoT trust model for IIoT networks, based on the H-IIoT architecture described in the previous phases 5.4.1 and 5.4.2.

The IIoT server sends to each CL_j all necessary information to evaluate the Tm of its members. Indeed, CL_j relay on these information to calculate three trust parameters called cooperation, honesty related to direct interactions and honesty related to indirect interactions, needed to evaluate the Tm. We remind that each node integrates the network by having an initial trust metric $T0$. The value of Tm and $T0$ varies within the interval [0, 1].
The Tm of a monitored MN_i can increase, decrease, remain unchanged or go down to zero as described in the previous chapter 4. Hence, to formalize the transitions of Tm, we use the state transition diagram at $M + 1$ states depicted in Fig. 4.1 of the chapter 4. We remind that each state corresponds to a specific value of Tm. The state 0 corresponds to $Tm = 0$, and state M corresponds to a high level of trust where $Tm = 1$. The interval [0, 1] of the

Tm is divided into $M+1$ states, each state represents a step ϕ [99].
As expressed in equations (4.9), (4.10), (4.13), (4.14) and (4.15), the transition of Tm from one state to another depend on the current state of the monitored MN_i and on the trust parameters namely, cooperation, honesty related to direct interactions and honesty related to indirect interactions. The Tm-IIoT model relies on the same trust parameters as in chapter 4, but evaluates these metrics differently according to the IIoT context.

To evaluate the different transition probabilities allowing to update the Tm of a MN_i, CL uses the information received from the IIoT server to calculate first the three trust parameters describing the behavior of this monitored MN_i over time.

- The cooperation rate : evaluates the behavior of the monitored MN_i related to its cooperation in the network. The cooperation rate C_{MN_i} of MN_i is the number of the successful forwarded messages divided by the total number NF of the messages transmitted to it by its CL_j. The cooperation rate of the monitored MN_i is calculated as follows:

$$C_{MN_i} = \left(\frac{\sum_{i=1}^{NF}(C_{mi})}{NF} \right) \tag{5.9}$$

Where C_{mi} has a value of 1 if the message mi is forwarded correctly or 0 if the message mi is not forwarded. Then, the probability p_c that the MN_i cooperates in the network is expressed as follows:

$$p_c = C_{MN_i} * p_{net} \tag{5.10}$$

As explained in chapter 4, the probability p_{net} depends on the transmission conditions in the network. We would like to remind the reader that the purpose of introducing this probability is to consider the constraints of the monitoring environment in the final calculation of trust.

- The rate of honesty related to direct interactions : measures the compatibility between the tasks performed by the MN_i in the network and the tasks that MN_i must perform and predefined in its profile. The node that monitors a production machine in order to send reports every hour to the production system is considered as a dishonest node when it does this task every four hours. Hence, this node must be isolated from the network in order to define the causes of its suspicious behavior. The rate of honesty related to direct interactions allows to detect this type of untrusted tasks, it also allows

to detect behavior changes produced by MN_i nodes.
The IIoT server sends to the CL_j, in a list $A(MN_i)$, the set of hashed tasks, that the monitored MN_i must perform in the network, i.e. the monitoring to carry out, the reports to produce, the recommendations to issue, etc. The tasks are sent hashed to CL_j nodes in order to prevent data leakage due to probable interception of monitoring messages by malicious nodes and in order to ensure that in the event of CL_j intrusion, the tasks will not be disclosed. To evaluate the rate D_{MN_i} of honesty related to direct interactions of MN_i, CL_j compares the set of hashes contained in the two lists $A(MN_i)$ and $AF(MN_i)$ by calculating their similarity ratio as follows:

$$D_{MN_i} = \left(\frac{A(MN_i) \cap AF(MN_i)}{A(MN_i) \cup AF(MN_i)} \right) \tag{5.11}$$

p_d is the probability that the monitored MN_i is honest, it is calculated as follows:

$$p_d = D_{MN_i} * p_{net} \tag{5.12}$$

The list $AF(MN_i)$ contains the set of feedback tasks and behaviors of MN_i. After the comparisons, $AF(MN_i)$ will be deleted automatically by the CL_j to optimize its memory space.

Untrusted tasks and behaviors are defined in our work as behavior changes and non-cooperation of IIoT nodes in the network. In the SIoT concept [27], nodes have an infinite number of tasks to perform because they are related to the unlikely activities of the human being. But in a factory the number of tasks defined for each IIoT node is finite, each node has a finite number of tasks to perform previously defined by the IIoT server. Hence, when an IIoT node changes behavior, its leader easily detects the changes by calculating the rate D_{MN_i}. Behavior changes of IIoT nodes are dangerous, simple disruptions can disclose the company's trade secret or can cause plant failure and major material damage as the Stuxnet worm attack [111] on Iran's nuclear installations showed in 2010.

- The rate of honesty related to indirect interactions: reflects the reputation of the monitored MN_i inside its community. The nodes in the community that already have experiences with MN_i must give it a recommendation according to its behavior in the network. The nodes that will give extreme recommendations, i.e. too big or too small recommendations than the other member nodes, will be reported as malicious nodes that aim to perform ballot

stuffing attacks or bad-mouthing attacks. CL_j uses routing to collect recommendations from the MNs belonging to its community. Afterward, CL_j removes spams by using the estimation algorithm 2, proposed in chapter 4, and sends to the IIoT server the list of nodes that generate spams to sanction them. CL_j calculates the rate of honesty related to indirect interactions of MN_i as an average of the recommendations given by the community members as follows:

$$I_{MN_i} = \left(\frac{\sum_{MN_{i'}=1}^{N} R_{MN_{i'},MN_i}}{N} \right) \tag{5.13}$$

Where $R_{MN_{i'},MN_i}$ is the reputation given by the $MN_{i'}$ to the MN_i, $MN_i \neq MN_{i'}$. MN_i does not give a score for itself to avoid self-promotion attacks. N is the total number of recommendations towards MN_i after removing spam.

The probability p_I that the monitored MN_i has a good reputation is calculated as follows:

$$p_I = I_{MN_i} * p_{net} \tag{5.14}$$

Based on the probabilities p_c, p_d and p_I describing the behavior of the monitored MN_i, CL_j calculates the increasing and decreasing state probabilities ($P_{s,s+1}(t)$ and $P_{s,s-1}(t)$), the state stay probability $P_{s,s}(t)$, the probability to sojourn in the trusted state M $P_{M,M}(t)$ and the probability to transit to state 0 $P_{s,0}(t)$. The mathematical formulas of these probabilities are already detailed in chapter 4.
According to the state transition diagram in Fig. 4.1 and the monitoring process, the CL_j can evaluate the Tm of its monitored MN_i. Indeed, knowing the Tm of the monitored MN_i in time $t-1$, at time t the value of this Tm can increase or decrease by 0.1, remain unchanged or take 0.

5.5 Performance evaluation

In order to prove the sensitivity, the responsiveness and the resiliency of our proposed Tm-IIoT model, we use the InstantContiki 2.7 platform [2]. The various simulation parameters are listed in Table. 5.1. We run the simulation experiments within an automotive plant of 100 heterogeneous nodes that are able to establish industrial relationships between them based on their profile previously defined by the IIoT server.

The performance evaluation in this chapter is done in two main parts. In the first part, we compare the energy efficiency of our proposed H-IIoT architecture

[2] `http://www.contiki-os.org/download.html`

Table 5.1: Simulation parameters

Simulation tool	contiki/cooja 2.7
Mote type	Tmote Sky
Nodes Distribution	Random
Total number of node	100
Deployment environment	automotive sector
Network Layer	IPV6 + 6loWPAN
Transport protocol	UDP
Mac layer	ContikiMAC
Physical layer	IEEE 802.15.4
Radio Medium	Unit Disk Graph Medium
Transmission ranges Trg	100 m
Interference ranges Irg	100 m
Monitoring period	10 min
Number of states M	10
T0	0.5
ϕ	0.1

with the energy efficiency of the traditional architecture of the plant's IIoT network. In the second part, we evaluate the sensitivity, the responsiveness and the resiliency of our trust management model compared to other trust management models proposed in the literature, including the TMCoi-SIoT model [122], the Adaptive IoT Trust model [116] and the CITM-IoT model [121].

5.5.1 Part 1: Comparative energy study

In this part, the topology of the plant's IIoT network is represented according to two architectures: traditional architecture in Fig. 5.2 and H-IIoT architecture in Fig. 5.3. The H-IIoT architecture is the result of the application of equation (5.3), equation (5.4), Algorithm 3 and Algorithm 4 to the traditional architecture represented in Fig. 5.2. The clustering is done at one hop. CLs use routing to collect monitoring data from the MNs belonging to their communities. This monitoring data is related to the behavior of MNs. It also contains the recommendations provided by the MNs.

IIoT nodes contribute to the trust management process by sending monitoring packets to their trust entities. The monitoring packets contain cooperation and transaction histories, they also contain recommendations for other nodes in the network. We apply the Tm-IIoT model on both architectures, and we evaluate for each one, in a time period Tr, the average power consumption of IIoT nodes during their contributions to the trust management process. The

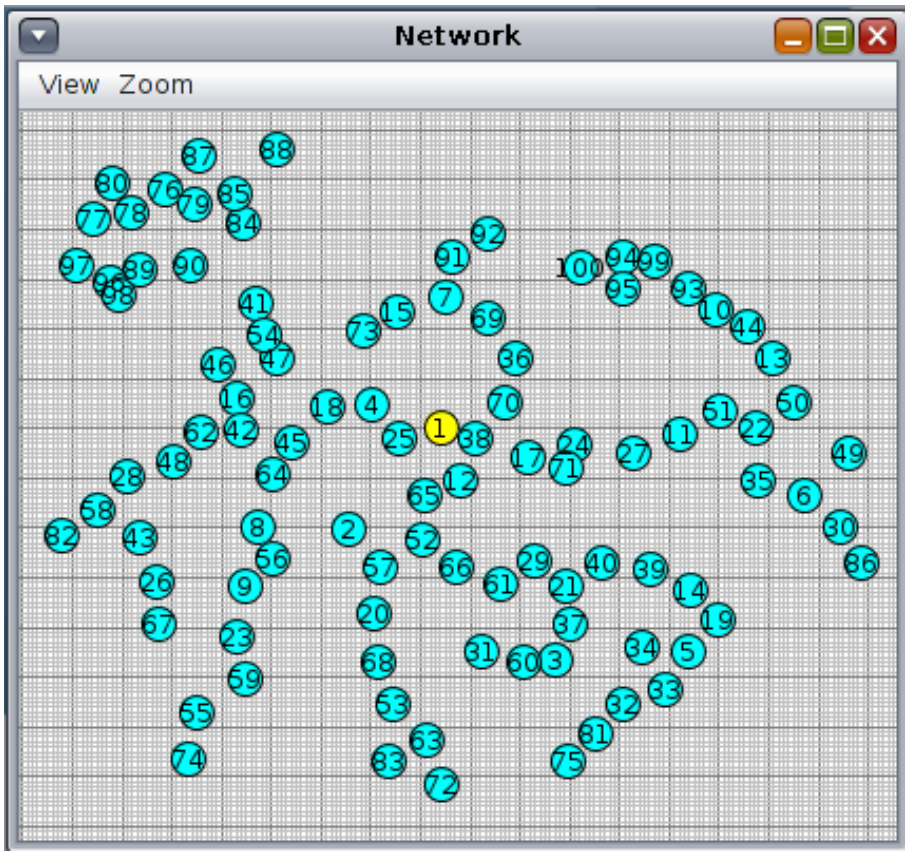


Figure 5.2: The topology of the traditional centralized IIoT architecture

results of this evaluation are shown in Fig. 5.4 and Fig. 5.5. For each architecture, we choose to represent only the average energy of nodes with the highest and the lowest consumption.

According to the histograms in Fig. 5.4 and Fig. 5.5, the central processing unit (CPU) of nodes belonging to the same architecture consumes on average the same amount of power because all of these nodes generate the same type and quantity of monitoring data. The average power consumed by the CPU of nodes in Fig. 5.4 is more significant than the average consumed by CPU of nodes in Fig. 5.5. Indeed, this is explained by the fact that nodes in traditional architecture generate more recommendations. They must give recommendations for all other nodes in the network which is impossible to achieve because nodes do not have a total knowledge of all network, they can give ratings just for nodes with which they have already had experiences. While in H-IIoT architecture, each MN_i provides recommendations just for the members of its community.

According to Fig. 5.4 and Fig. 5.5, we notice that the overall power consumption of IIoT nodes during their contributions to the monitoring process is impacted by their radio operations. Indeed, in the traditional architecture,

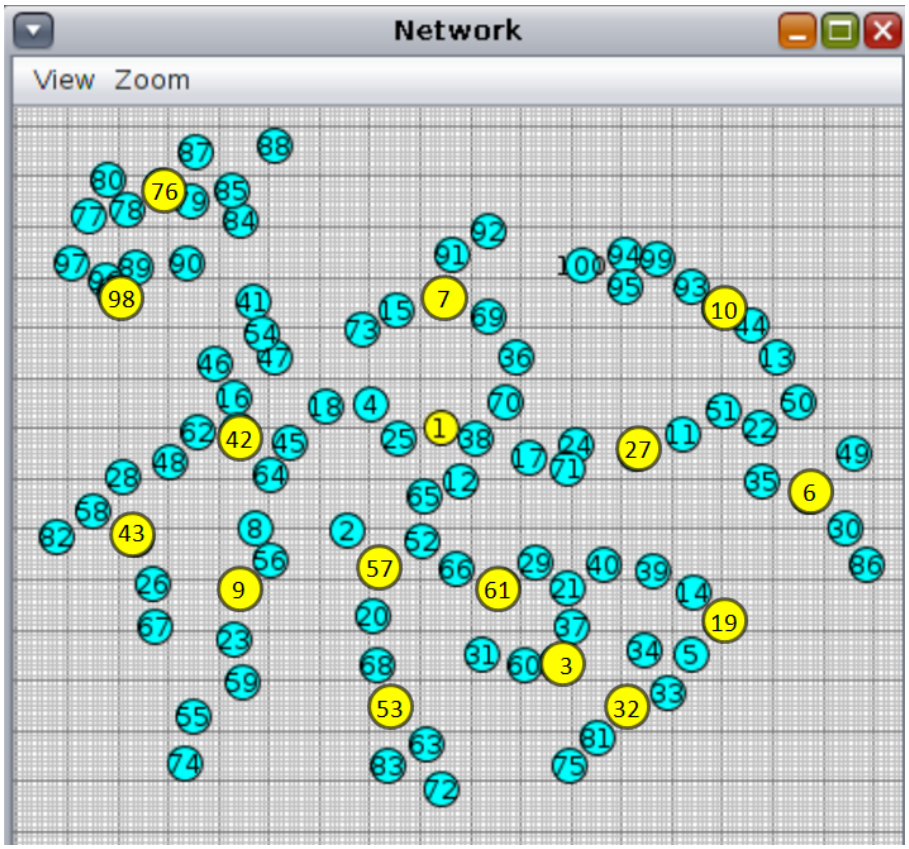


Figure 5.3: The topology of the proposed H-IIoT architecture

IIoT nodes consume more radio resources in order to relay monitoring packets from the entire IIoT network. Whereas in our proposed architecture, nodes only relay the monitoring packets from nodes belonging to their communities. For instance, node 24 in Fig. 5.4, consumes on average 8.149 mW (listen radio Power: 3.894 mW, radio transmission Power: 2.658mW) compared to Fig. 5.5 where it consumes 1.2 mW (listen radio Power: 0.578 mW, radio transmission Power: 0.05 mW). These obtained simulation results are estimated for 100 IIoT nodes and for an accuracy $\kappa = 0.5$. Over time, with the evolution of the network and the need to have more accuracy κ, these results will increase exponentially as shown in Fig. 5.6.

The obtained results in Fig. 5.6 estimate the average energy consumption of IIoT nodes according to the evolution of the network (from 100 nodes to 1600 nodes) and according to the level of accuracy that we want to achieve in the evaluation of trust. As shown in Fig. 5.6, the average energy consumption increases with the increase of the accuracy. The consumption increases even more when the number of nodes in the network increases, especially in the traditional centralized architecture Fig. 5.6a. As shown in Fig. 5.6a, nodes can consume on average until 75% of their energy just in order to relay monitoring

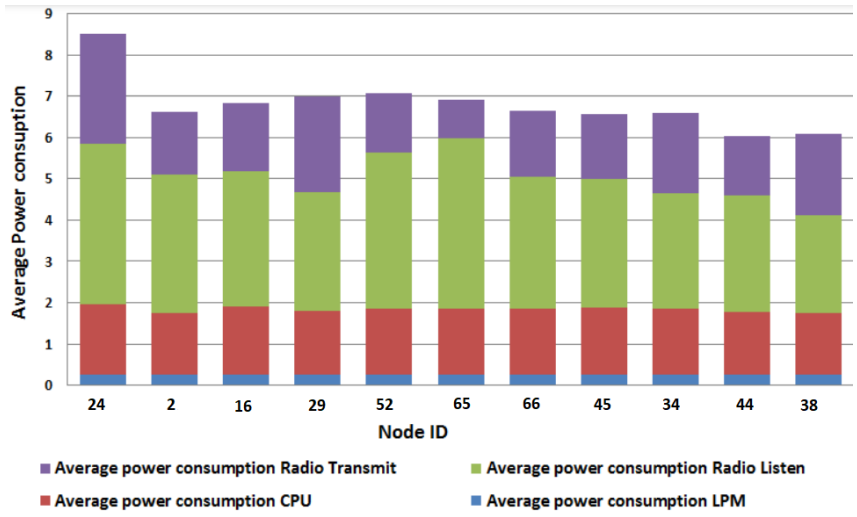


Figure 5.4: The average power consumption in the traditional IIoT architecture

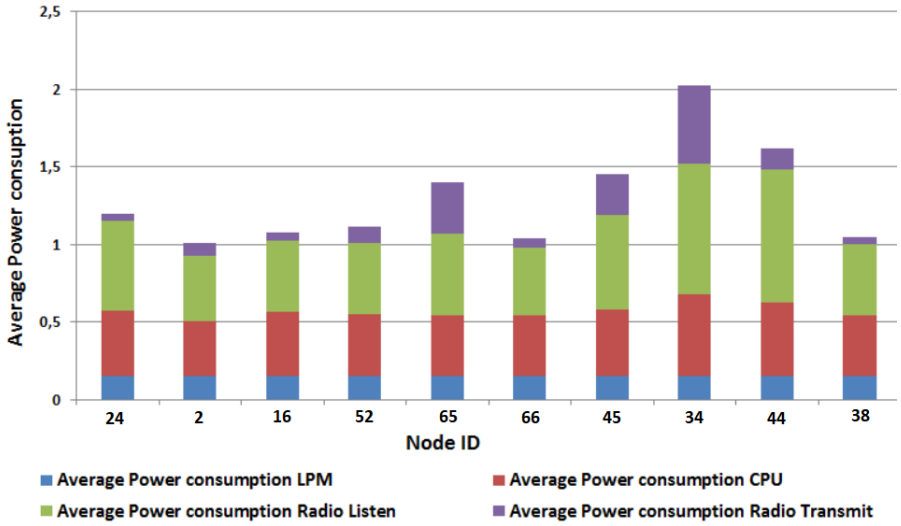
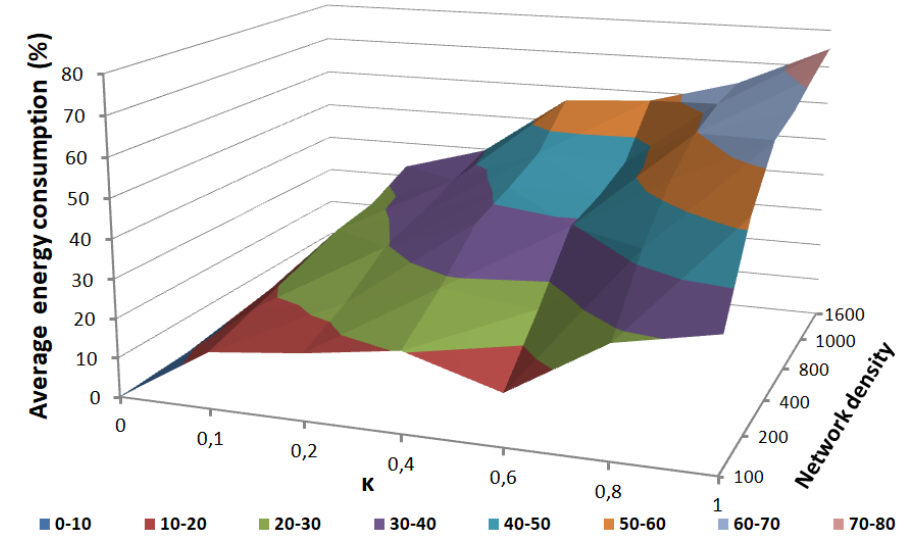
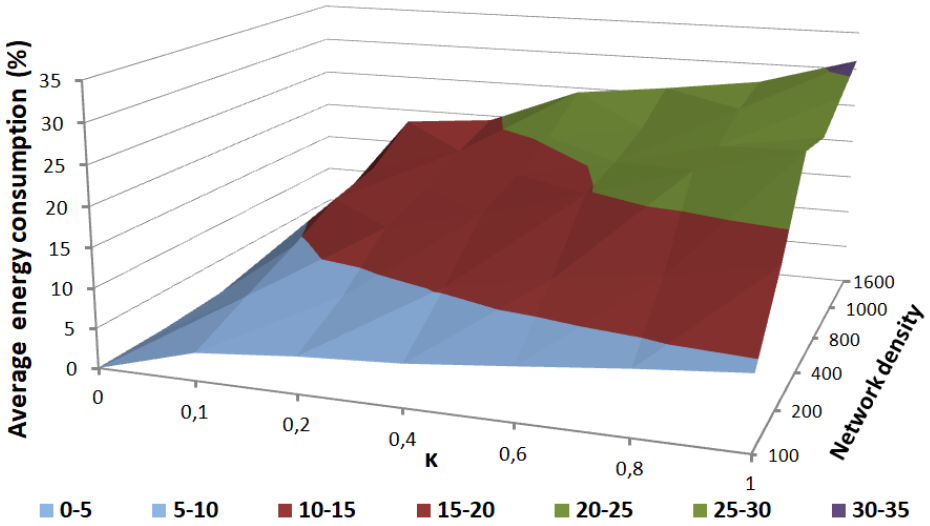


Figure 5.5: The average power consumption in the proposed H-IIoT architecture

packets, while at the base, nodes must use this energy for their own tasks. We justify these results by the fact that the more the network evolves and the accuracy increases, the more the generation of monitoring packets increases and the more radio operations become more important and consume more energy. In Fig. 5.6a and Fig. 5.6b, with a network of 1600 nodes and a maximum accuracy $\kappa = 1$, MNs in the H-IIoT architecture consume on average until 30% of their energy compared to 75% of energy in the traditional architecture. This is due to the fact that in the H-IIoT architecture, the network is managed in communities so each MN_i relays the monitoring packets of its community and not of the whole network. Furthermore, in the H-IIoT architecture, the number of CLs is proportional to the density of the nodes in the network.



(a) Centralized architecture

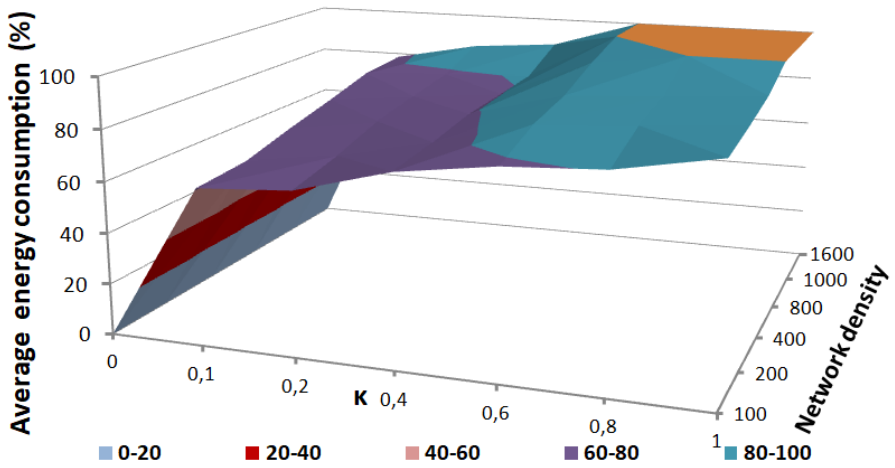


(b) H-IIoT architecture

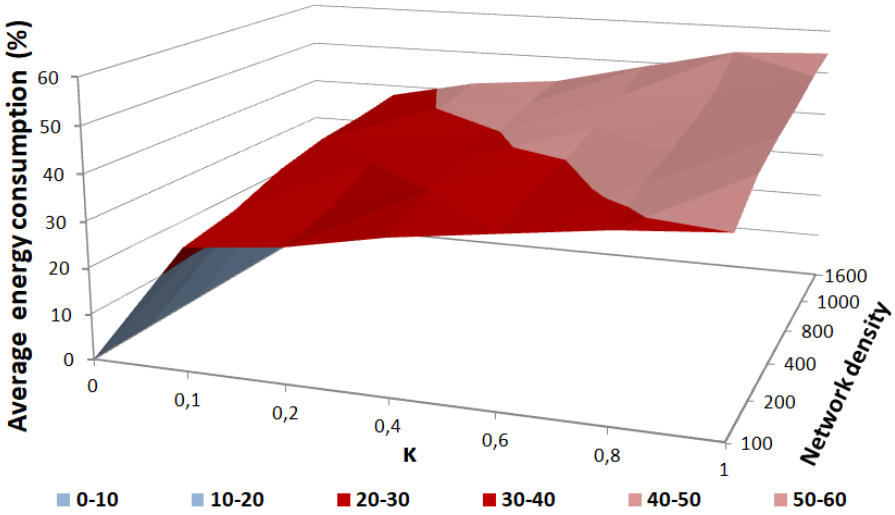
Figure 5.6: Average energy consumption of nodes with the evolution of the network population and the increase of κ

To manage the Tm of 1600 nodes with an accuracy $\kappa = 1$, CLs consume on average until 60% of their energy reserved for monitoring as shown in Fig. 5.7b. While for the traditional architecture as seen in Fig. 5.7a and for the same conditions, the trust management entity consumes on average until 100% of its energy reserved for monitoring.

According to Fig. 5.8, the average number of lost monitoring packets is more significant when the accuracy and the density of the IIoT network increase. The losses are higher in the centralized architecture Fig. 5.8a compared to the H-IIoT architecture in Fig. 5.8a. The lost packets contain monitoring data that will allow the trust entity (CLs in our architecture) to manage the



(a) Centralized architecture



(b) H-IIoT architecture (consumption of CL)

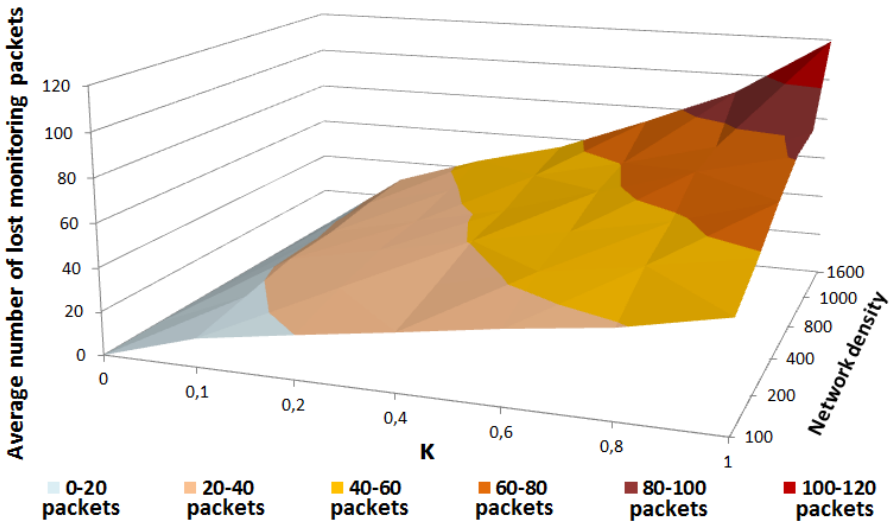
Figure 5.7: Average energy consumption of trust management entities, with the evolution of the network population and the increase of κ

Tms. Indeed, each lost packet introduces errors in the accuracy of the Tms evaluation and causes retransmissions as well as energy losses.

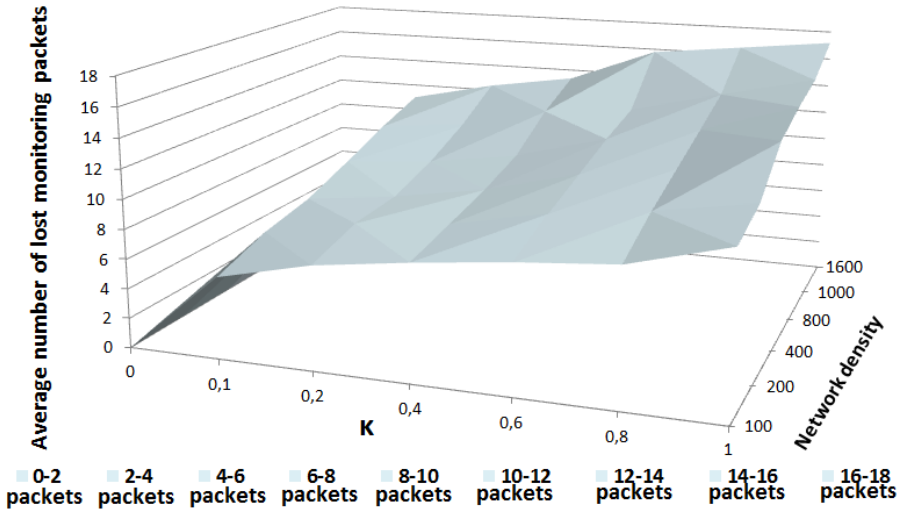
All obtained results in this part justify the need to change the traditional centralized architecture of IIoT networks to the H-IIoT architecture.

5.5.2 Part 2: The sensitivity, the responsiveness and the resiliency of our proposed trust management model to trust-related attacks launched by malicious IIoT nodes.

In this part, we analyze the trust convergence properties of an MN_i belonging to a community of 20 nodes, in an environment containing varying percentage



(a) Centralized architecture



(b) H-IIoT architecture

Figure 5.8: Average number of lost monitoring packets, with the evolution of the network population and the increase of κ

of malicious nodes Pm ranging from 20% to 50%. For this purpose, we use the simulation parameters listed in table. 5.1. We observe in Fig. 5.9 that the convergence time and the trust bias increase with the increase of malicious nodes. However, even with a malicious nodes population of 50%, the trust bias remains small. This proves the resilience of the proposed Tm-IIoT model in the most unfavorable environments.

We demonstrate the effectiveness of our proposed Tm-IIoT model with a comparative performance analysis against the TMCoi-SIoT model proposed in [122], Adaptive IoT Trust model proposed in [116] and CITM-IoT model proposed in [121]. The TMCoi-SIoT model is applied directly to the industrial context without modification, but the Adaptive IoT Trust model and

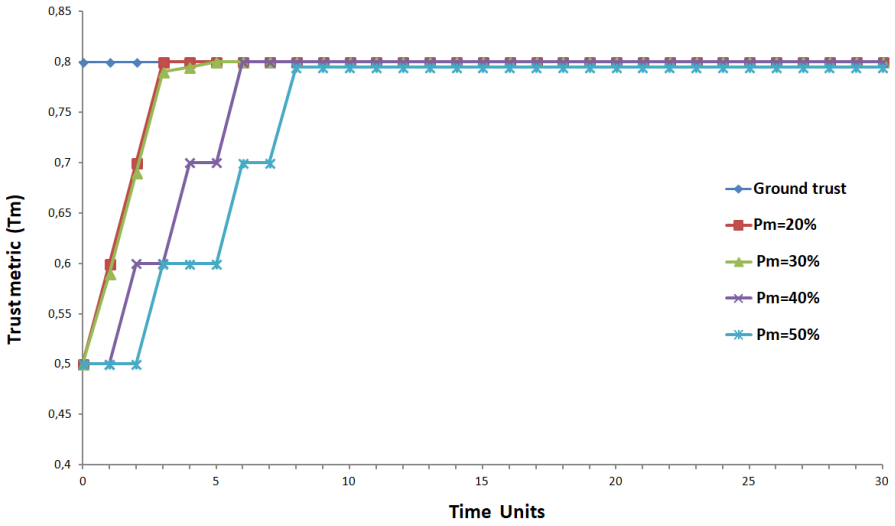


Figure 5.9: Tm of a Good node with Pm ranging from 20% to 50%

the CITM-IoT model require some modifications and adaptations; we have replaced social relations with industrial relations.

As shown in Fig. 5.10, with a malicious nodes population of 20%, the convergence time in CITM-IoT model is faster compared to that of our model. This is due to the non-complexity of the calculations performed in the CITM-IoT model; Tm of a cluster node is the average of the evaluations collected from peer cluster nodes after removing outliers. We notice also that in CITM-IoT model, the Tm never reaches the exact value 0.8. This is justified by the lack of trust management context, the authors do not specify on which basis the nodes manage the peers of their cluster. The trust bias in CITM-IoT model increases brutally even more when the number of malicious nodes reaches 50% as shown in Fig. 5.11, this is due to the lack of context and also to the vulnerability of the model to coalition attacks. The spam algorithm used in this model is unable to detect coalition attacks when the number of malicious nodes increases. Indeed, the algorithm makes its decision based on the evaluation given by the majority of cluster nodes. It has as a majority the evaluations given by untrusted cluster nodes and it eliminates the evaluations of legitimate nodes.

When the number of malicious nodes reaches 20%, the Adaptive IoT Trust model reaches convergence at 13 time units as shown in Fig. 5.10. This is due to the distributed type of the model. Indeed, in order to have the final Tm of a monitored node, we have to do an aggregation of the all local Tms given to the monitored node by its neighbors. When the number of malicious nodes reaches 50%, the trust bias increases brutally because there is no spam value extraction before calculation of the final Tm value. We notice that the Tm

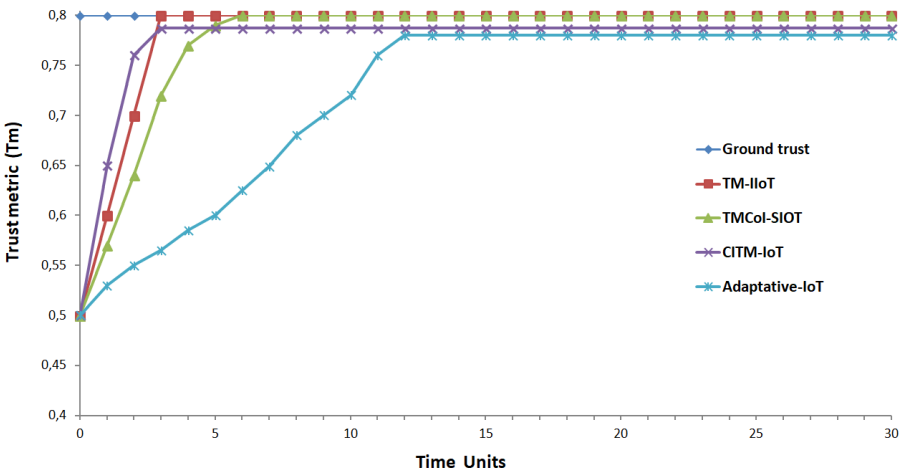


Figure 5.10: Tm of a good node, Pm=20%

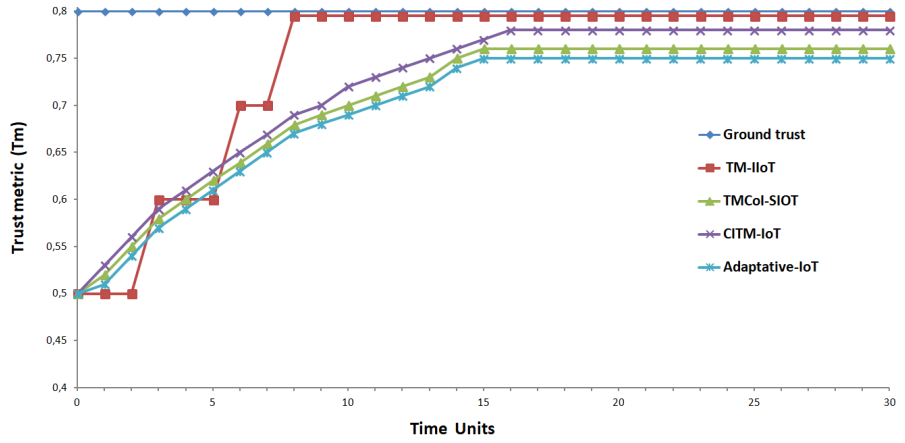


Figure 5.11: Tm of a good node, Pm=50%

never reaches the value 0.8. The trust bias increases also for the TMCoi-SIoT model specially when the population of malicious nodes increases as shown in Fig. 5.11. The TMCoi-SIoT model does not detect spam recommendations, it takes them into account in its trust calculation.

In order to prove the resiliency of our proposed Tm-IIoT model to the dynamic behavior of IIoT nodes, we evaluate the behavior change detection of two nodes: a normal node containing non-sensitive data in Fig. 5.12 and a critical node containing more sensitive data in Fig. 5.13. Hence, we assume that during the first 20 time units, both nodes behave positively in the network to increase their Tm. Thereafter, the nodes change their trusted behavior to an untrusted behavior following an attack for 30 units of time followed by a trusted behavior for 30 units of time and so on as shown by Fig. 5.12 and Fig. 5.13.

According to the obtained results, for the less critical node in Fig. 5.12,

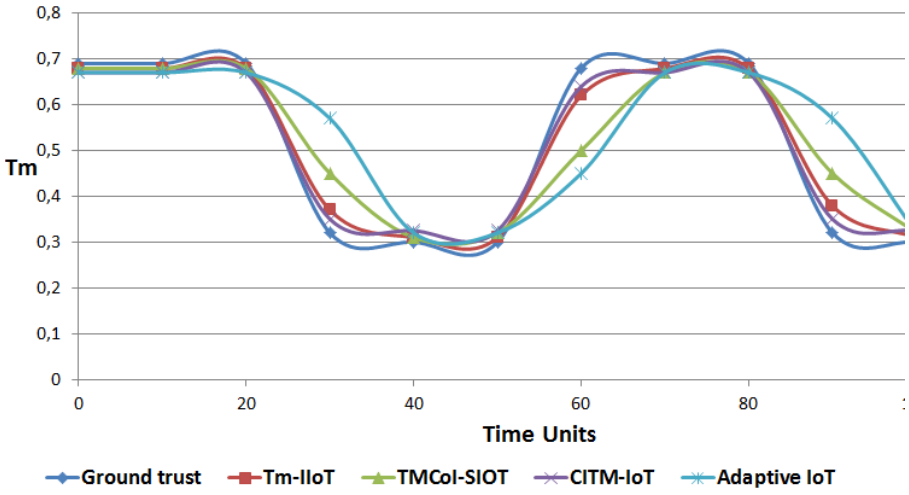


Figure 5.12: Behavior changes of a non-critical node, Pm=30%

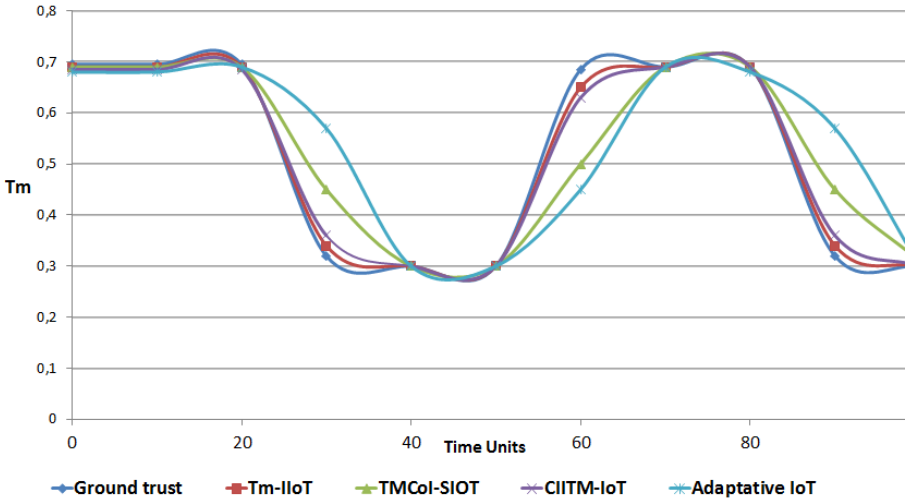


Figure 5.13: Behavior changes of a critical node, Pm=30%

the Tm-IIoT model is more sensitive to behavior changes. Indeed, the model accurately detects the node behavior perturbations at the 24 th time units compared to the TMCoi-SIoT model and the Adaptive IoT Trust model which take more than 9 time units and 15 time units respectively to detect the disruptions. Our model takes less time (22 time units) to detect behavior changes of the sensitive node in Fig. 5.13, unlike other models that take the same average time as for the less critical node. They do not differentiate between sensitivity levels of IIoT nodes, unlike our model.

When a node returns to its initial trusted behavior, the Tm-IIoT model quickly detects this behavior recovery and the Tm of this node returns to its previous value after 2 time units if it contains critical data else after 4 time units if it contains less critical data. TMCoi-SIoT model, Adaptive IoT Trust model and CITM-IoT model are less sensitive to behavior changes, they take a longer

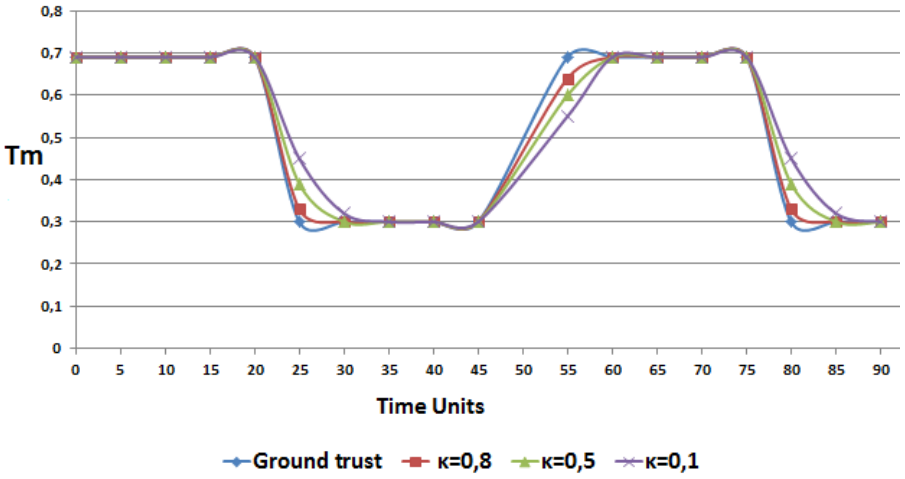


Figure 5.14: Behavior changes of a critical node, $\kappa = 0.8$, $\kappa = 0.5$, $\kappa = 0.1$

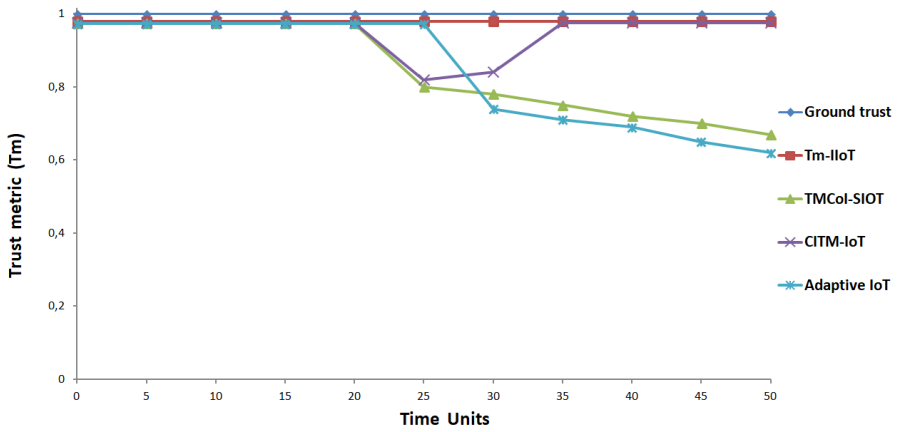
time to recover the previous Tm value of a node that resumes its initial good behavior. Regardless of the sensitivity of nodes, the recovery of the Tm takes at least 10 time units for TMCoi-SIoT model and 15 time units for Adaptive IoT Trust model. This proves the sensitivity and responsiveness of our trust management model to behavior change. The results in Fig. 5.12 and Fig. 5.13 are obtained for $\kappa = 0.8$.

As shown in Fig. 5.14, with the variation of κ, the behavior change detection will not maintain the same accuracy. The Tm-IIoT model detects behavior changes of critical nodes after 6 time units when $\kappa = 0.1$ and after 4 time units when $\kappa = 0.5$ compared to 2 time units when $\kappa = 0.8$. Despite this variation, our model shows a high resiliency against behavioral changes compared to other models which regardless of the sensitivity of nodes they keep the same treatment.

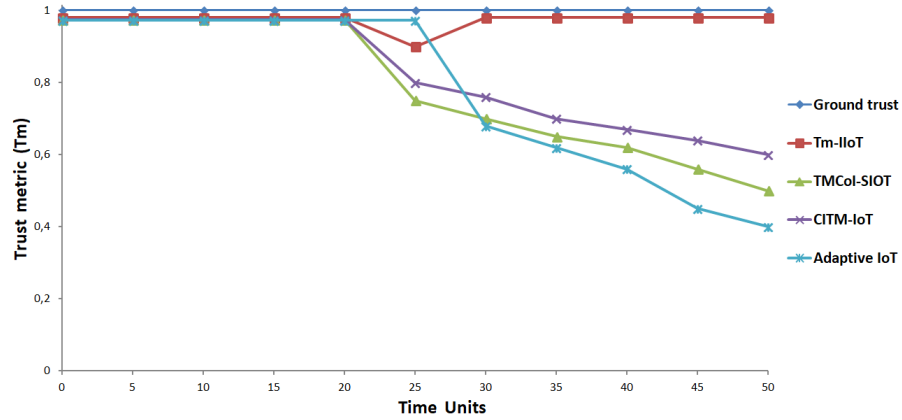
Coalition attacks: we would like to remind the reader that Coalition attacks occur when a group of malicious nodes:

- Mobilize against a good node in order to reduce its Tm by sending it bad recommendations. In this case, the coalition attack is called bad-mouthing attack [123].
- Increase the Tm of a malicious node. In this case the coalition attack is called ballot-stuffing attack [123].

To evaluate the resiliency of our trust management model to coalition attacks, we propose the following two scenarios: In Fig. 5.15, we assume a bad-mouthing attack against a legitimate MN_1 with Tm=1 and in Fig. 5.16 we assume a ballot stuffing attack performed to increase the Tm of a malicious



(a) Pm=20%



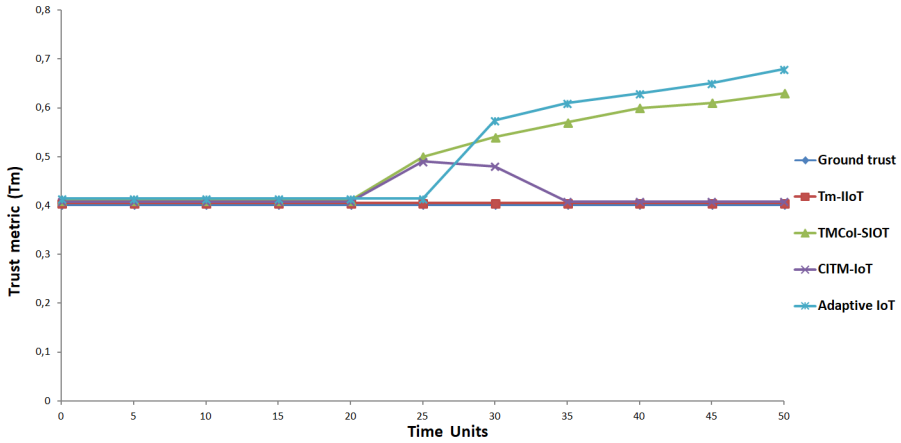
(b) Pm=50%

Figure 5.15: Coalition attack: Bad-mouthing attack

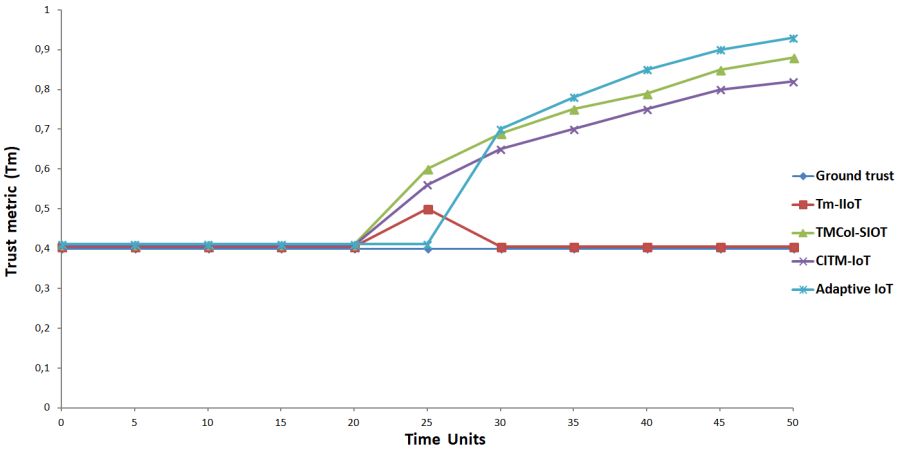
MN_2 with Tm=0.4. Each of MN_1 and MN_2 belongs to a community of 20 IIoT members. The evaluation is conducted in an environment that initially contains 20% of malicious nodes and then 50%, to properly evaluate our model in the most unfavorable environments.

It is assumed for both scenarios that malicious nodes perform coalition attacks at the 21 th time unit. We notice that for 20% of malicious nodes, the Tm in the proposed Tm-IIoT model does not vary as shown in Fig. 5.15a and Fig. 5.16a. However, when the percentage of malicious nodes reaches 50%, the Tm in our model varies briefly until reaches the maximum value of 0.9 in Fig. 5.15b and 0.5 in Fig. 5.16b. The Tm disruptions in the two scenarios will not affect the security of MN_1 and MN_2 because their Tm vary slightly and quickly return to their previous values.

When the percentage of malicious nodes reaches 50%, we notice that the Tm in our model takes time to converge to its previous value. This is due to the estimation algorithm described in chapter. 4, it needs more time to



(a) Pm=20%



(b) Pm=50%

Figure 5.16: Coalition attacks: Ballot-stuffing attack

detect spams. The estimation algorithm must be reinforced by the rate of honesty related to direct interactions that compares the current behavior of IIoT nodes with their profiles described in the IIoT server. In comparison with the TMCoi-SIoT model and Adaptive IoT Trust model, we clearly notice that the Tm strictly diverges and never recovers its previous value in both models. This is due to the fact that these models do not use a mechanism to detect spam recommendations from malicious nodes. Indeed, these two trust management model take spam into account when calculating and updating the Tms.

Fig. 5.15b and Fig. 5.16b show that the CITM-IoT model is also vulnerable to coalition attacks when the percentage of malicious nodes reaches 50%. Indeed, the proposed algorithm to eliminate outliers in this work is based on the evaluation given by the majority of cluster nodes.

5.6 Conclusion

In the first part of this chapter, our purpose was to change the traditional architecture of IIoT networks in the automotive plants into a hierarchical architecture called H-IIoT in order to manage the trust of the IIoT nodes. The H-IIoT architecture is based on a new concept called industrial relationship between IIoT nodes. In the second part of this chapter, we proposed the Tm-IIoT model to manage the Tm of IIoT nodes in the proposed H-IIoT architecture. The simulation results have proven the energy efficiency of our proposed H-IIoT architecture compared to the traditional IIoT network architecture. They have also proven the sensitivity, responsiveness and resiliency of the Tm-IIoT model to trust-related attacks launched by malicious IIoT nodes.
Finally, we focused on designing a trust management model appropriate for IIoT networks to detect the untrusted and suspicious behavior of nodes in these environments. However, we have not proposed an effective method to isolate malicious nodes from the network in order to prevent them from launching internal attacks. This is foreseen for the next chapter 6.

Chapter 6

Certificate revocation in IIoT networks using Signaling game

Contents

6.1 Introduction 100

6.2 Related certificate revocation strategies 101

6.3 Stage Certificate Revocation Game 103

6.4 Equilibria of the stage Certificate Revocation Game . . 106

6.4.1 Pure-strategy BNE of the stage Certificate Revocation Game 107

6.4.2 Mixed-strategy BNE of the stage Certificate Revocation Game 109

6.5 Multi-stage dynamic Certificate Revocation Game 111

6.6 The PBE of the Certificate Revocation Game in mixed-strategy 113

6.7 The proposed certificate revocation mechanism based on the PBE of the multi-stage Game 115

6.8 Performance evaluation of the certificate revocation mechanism 116

6.9 Conclusion 121

6.1 Introduction

The malicious MNs may disclose data related to the industrial processes and damage the basic functionalities of the network through internal attacks. For this purpose, it is necessary to isolate these nodes from further contributing to network activities by revoking their certificates. However, malicious MNs are difficult to detect because they are considered to be internal attackers who have successfully passed the authentication phase and are member of the network. As detailed in the previous chapter, each CL continuously monitored the behavior of MNs belonging to its community. In this chapter, the CL uses the result of these monitoring to manage effectively the certificates of MNs and enhance the security of the IIoT network. Hence, we assume that each CL hosts an agent, named CL-UR agent, that renews the certificate of legitimate MNs and revokes the certificate of nodes that show untrusted behavior in order to isolate them from the network and prevent them from launching internal attacks. In the hierarchical architecture illustrated in Fig. 5.1, each MN knows the type of its CL, i.e. the legitimate node that contains an agent responsible for the revocation. However, the type of the MN (trusted or untrusted) is hidden information for the CL-UR agent. Indeed, the CL-UR agent has no information on the type of MN, which can impact its revocation decision. Hence, the CL-UR agent lets the MN move first to observe its behavior and then act; it renews the MN's certificate or revokes it according to its observations. Indeed, the behavior sent by the MN can be a signal about its type.

To analyze the interactions between the CL-UR agent and a MN, and model the certificate revocation process in the IIoT network, we use Signaling game theory in which players can use the actions of their adversary to make deductions about hidden information. The Signaling game modeling the interactions between the two players, CL-UR agent and MN, is called "Certificate Revocation Game".

The remainder of this chapter is organized as follows. In section 6.2, we present a concise summary of the literature review on certificate revocation strategies used in wireless networks. The critical analysis of advantages and limitations of each strategy provides a clear vision of how to design an effective certificate revocation mechanism for IIoT networks. In section 6.3, we study a stage Certificate Revocation Game considered at an individual time slot between the CL-UR agent and a MN. In section 6.4, we look for the Bayesian Nash Equilibrium (BNE) that the stage Certificate Revocation Game can achieve in pure-strategy and in mixed-strategy respectively. In section 6.5, as the game

evolves, we develop the stage Certificate Revocation Game into a Multi-stage game in which, the CL-UR agent can adjust its strategy based on its belief updated dynamically according to the new behaviors of the monitored MN. Based on the Perfect Bayesian equilibrium (PBE) of the Multi-stage Certificate Revocation Game obtained in section 6.6, we propose an effective Certificate revocation mechanism for IIoT networks in section 6.7. We present the performance evaluation and the comparisons results in section 6.8. Finally, in section 6.9, we conclude the chapter.

6.2 Related certificate revocation strategies

In this section, we present a concise summary of the literature review on certificate revocation strategies used in wireless networks. We also discuss the advantages and limitations of each strategy.
In the literature, several works rely on trust management [20] to deal with malicious nodes. Indeed, trust management provides continuous analysis of the behavior of nodes to predict their performance over time, which improves the revocation decision process and enhances the security of the networks. To make a revocation decision, trust-based revocation mechanisms use either non-voting strategy or voting strategy.

In the non-voting strategy, each node in the network monitors the behavior of its neighbor and sends an accusation against it if suspicious behavior is detected. Indeed, only one accusation is able to launch the revocation process, which reduces the time required to revoke a certificate and the communication overhead resultant from the revocation process. Authors of [127] assumed that revocation can be much simpler and faster if it is entrusted to a single node. For this aim, [127] proposed a revocation mechanism founded on the suicide for the common good where the accusing node sacrificing itself to prove to its neighbors the sincerity of its accusations. Indeed, when a node B with a valid certificate detects an untrusted behavior issued by a node A, B puts its own identity as well as the identity of A in a signed suicide note and broadcasts the note to other nodes in the network. Each node receiving the note verifies its validity by using the signature of B, and revokes both B and A by putting them in a blacklist. The proposed mechanism is vulnerable to false accusations because the accusing nodes may be malicious. Moreover, the mechanism is effective only under the condition that the benefit to the attacker of revoking an innocent node is less than the benefit of having a malicious node placed in the network, which is not always the case in IIoT networks.

To address false accusations, several works such as [128], [129], and [133] included a certificate recovery module in their revocation mechanisms. In [133], when the Certificate authority (CA) receives an accusation, it puts the accusing node in a warning list and the accused node in a blacklist and it broadcasts both lists to other nodes in the network. Upon receiving the two lists, the leader (CH) of each cluster evaluates the trust of the accused nodes present in its cluster by using fuzzy logic. When the CH does not identify any suspicious behavior, it sends a request to the CA to recover the falsely accused nodes. However, the designation of CHs in [133] is not based on their trust level. Hence, a malicious CH may judge an accused node as innocent when it is malicious. Despite several attempts [130] to solve challenges of non-voting strategy, revocation mechanisms based on this strategy remain unsuitable for IIoT networks due to their low accuracy and low reliability.

The voting strategy revokes the certificate of a suspect node when the number of accusations against it exceeds a threshold. Authors of [131] did not rely on a single node to make revocation decisions, as any individual node is subject to misbehavior. For this purpose, the revocation in [131] was achieved through a consensus between several nodes. Indeed, each node is monitored by its neighbors at one or two hops distance. The neighbors use the result of their monitoring either to renew the certificate of the monitored node if it behaves well, or to send accusations against the monitored node if it shows untrusted behavior. When the number of accusations exceeds a threshold, the certificate of the monitored node will be revoked. Authors of [47] proposed a cooperative Trust-based Intrusion Detection System (TIDS) to detect malicious nodes in the network. In TIDS, each node periodically monitors the behavior of its one-hop neighbors and collaboratively calculates their trust values. Nodes whose trust value is below a certain threshold will be reported to the Border Router to put their identities in the list of potential malicious nodes. The list is then shared with all nodes. The voting strategy overcomes false accusations and improves the accuracy and reliability of the revocation process. However, mechanisms based on voting strategy are vulnerable to coalition attacks, when a set of nodes falsely accuse honest devices to revoke their certificates.

Despite all the attempts to improve mechanisms based on voting strategy [132], [134], [42], the voting strategy remains inappropriate for application to IIoT networks. Indeed, a malicious IIoT node may change its behavior without conducting attacks against its neighbors as explained in [111]. In this scenario, the behavior of the malicious node cannot be detected by the voting-based mechanisms, because these mechanisms initiate the revocation process

based on the neighbors' accusations. Moreover, the voting strategy is not able to instantly revoke a malicious node if there are not enough honest nodes around it, which increases the revocation time and affects the reliability of the network. These mechanisms also generate communication overhead resulting from the exchange of accusations.

Table 6.1: Non-voting strategy vs voting strategy

	Non-voting strategy	**Voting strategy**
Communication overhead	low	high
Revocation time	low	high
False accusation	very very high	low
Accuracy	very low	high (conditionally)
Reliability	very low	high (conditionally)
Unfavorable environment	highly impacted	highly impacted
Trust-related attacks	highly vulnerable	highly vulnerable

As discussed above and summarized in Table 6.1, mechanisms based either on voting or non-voting strategy suffer from several weaknesses that makes them inappropriate to be deployed in IIoT networks. By highlighting research gaps, we propose a hierarchical architecture where each CL hosts an agent that monitors the behavior of the nodes belonging to its community to manage their certificates. Indeed, the revocation process is made by CLs, nodes with very high level of trust, in order to avoid false accusations and coalition attacks.

To remove uncertainty about the type of a member node (trusted or untrusted), we model the interactions between the member node and its CL by using signaling game theory, in which players can use the actions of their adversary to make deductions about hidden information. As the game between the agent and the member IIoT node evolves, the agent can obtain its best response strategy based on its belief on the member node, updated by using the Bayesian rules. This allows the agent to make rational, accurate and fast decisions which increases the reliability of our proposed certificate revocation process.

6.3 Stage Certificate Revocation Game

The main gain of a malicious MN is when it successfully launches internal attacks and gradually causes the collapse of the IIoT network without being detected. The MN pays an energy consumption cost to accomplish the untrusted behavior. Hence, we introduce Gm and Cm to indicate the gain of

the MN when it presents untrusted behavior in the network and the energy consumption cost it has to pay for this behavior, respectively.
Sometimes a malicious MN may behave honestly in the network to disguise and whitewash its previous untrusted behavior. Thus, regardless of the type of MN (trusted or not), as long as it behaves honestly in the network, it will be rewarded for its honest behavior. On the other hand, the MN must pay a cost due to the energy consumption needed to perform an honest behavior. We introduce G_G and C_G to denote the gain of the MN when it behaves honestly in the network and the energy consumption cost that it must pay for an honest behavior, respectively.
A MN shows honest behavior if:

- It behaves according to the requirements of its profile stored in the IIoT server.
- It cooperates positively within the network by forwarding packets from other nodes, and providing correct recommendations.

The certificates of MNs have a limited validity period, they are renewed if their validity period has expired, or revoked if the conditions for renewal are not met. Hence, when the CL-UR agent successfully detects the untrusted behavior of a MN and revokes its certificate, the agent obtains the gain G_R as a reward and pays the cost C_R due to the energy consumed during the revocation process. The CL-UR agent obtains the gain G_{UP} when it renews the certificate of a MN that proves honest behavior, and obviously, it pays the cost C_{UP} due to the energy consumed during the renewal process.
It is assumed that each CL-UR agent is characterized by a detection rate, denoted by φ, and a false alarm rate, denoted by μ. These rates depend mainly on the technical and material characteristics of the agents. The loss of the CL-UR agent due to a false alarm is noted by I_F. In our game, we assume that all CL-UR agents have the same detection rate and the same false alarm rate. We gather all the notations used by the proposed Certificate Revocation Game in the table 6.2.

We define the stage Certificate Revocation Game in the strategic form, also called the normal form, by five tuple (PL, χ, A, P, Uf) where:

- PL = {MN (Sender), CL-UR agent (Receiver)} is a set composed of two players. We designate the player MN by X_S, where $X_S = 1$ if MN is malicious, $X_S = 0$ if MN is legitimate. We designate the player CL-UR agent by X_R.

Table 6.2: Notations used by the Certificate Revocation Game

G_G	Gain of MN when it behaves honestly in the network
C_G	Energy consumption cost that MN pays for the honest behavior
G_M	Gain of MN when it presents untrusted behavior in the network.
C_M	Energy consumption cost that MN pays for the untrusted behavior
G_{UP}	Gain of the CL-UR agent for renewing the certificate of a MN
C_{UP}	Energy consumption cost resulting from the renewal of the certificate of a MN
G_R	Gain of the CL-UR agent for revoking a certificate
C_R	Energy consumption cost that the CL-UR agent must pay to revoke a certificate
φ	Detection rate of the CL-UR agent
μ	False alarm rate of the CL-UR agent
I_F	Loss of the CL-UR agent due to false alarm

- $\chi = \chi_S \times \chi_R$, $\chi_S = \{X_S = 1, X_S = 0\}$ is the set of type space of the X_S, and χ_R is the set of type space of the X_R. Since the X_R in our proposed game has one type, then $\chi_R = \{X_R = CL - UR\ agent\}$.

- $A = A_S \times A_R$, is the set of possible actions (or strategies) for the players. A_S= {Untrusted behavior, Honest behavior} for the X_S and A_R= {Renew, Revoke} for the X_R.

- $P = (p, 1-p)$ is a common prior probability distribution over available types of the χ_S, i.e. $P\colon \chi_S \to [0, 1]$. p is the probability that the MN is malicious i.e. $X_S = 1$ and $(1-p)$ is the probability that the MN is legitimate i.e. $X_S = 0$.

- $Uf = (Uf_S, Uf_R)$, where $Uf_S\colon A \times \chi \to \mathbb{R}$ is the utility function for the X_S and $Uf_R\colon A \times \chi \to \mathbb{R}$ is the utility function for the X_R. Utility is the measure of each situation from the player's point of view; it is not a measure of material or monetary gain but rather a subjective measure of player satisfaction. We detail the calculation of Uf_S and Uf_R in Table. 6.3.

If the MN shows an untrusted behavior and the CL-UR agent detects it, the agent will revoke the certificate of the MN. Hence, the utility of the MN is the gain of being not revoked when it has performed untrusted behavior in the past while the CL-UR agent does not detect it, minus the loss of being revoked, and minus the cost of untrusted behavior. While the utility of the CL-UR agent is the gain of the revocation minus the cost resulting from the certificate revocation process. When the MN shows an untrusted behavior in

Table 6.3: Utility functions of the Certificate Revocation Game

χ	A_S	A_R	
		Renew	**Revoke**
$X_S = 0$	**Honest**	$Uf_S = G_G - C_G$ $Uf_R = G_{UP} - C_{UP}$	$Uf_S = G_G - C_G$ $Uf_R = -\mu I_F - C_R$
$X_S = 1$	**Honest**	$Uf_S = G_G - C_G$ $Uf_R = G_{UP} - C_{UP}$	$Uf_S = G_G - C_G$ $Uf_R = -\mu I_F - C_R$
	Untrusted	$Uf_S = G_M - C_M$ $Uf_R = -G_M - C_{UP}$	$Uf_S = (1-\varphi)G_M$ $-\varphi G_R - C_M$ $Uf_R = \varphi G_R - (1$ $-\varphi)G_M - C_R$

the IIoT network and the CL-UR agent renews its certificate, the utility of the agent is the loss of being attacked and the cost resulting from the certificate renewal process. Whereas, the utility of the MN is the gain of the untrusted behavior minus the cost of this behavior.
When the MN (malicious or legitimate) acts correctly and honestly in the network, its utility is the gain of performing honest behavior minus the cost resulting from this behavior. When the CL-UR agent renews the certificate of a node that behaves honestly, its utility is the gain of certificate renewal minus the cost resulting from the certificate renewal process. However, when the agent revokes the certificate of an honest MN, its utility is the loss of false alarm minus the cost of the certificate revocation process.

6.4 Equilibria of the stage Certificate Revocation Game

The proposed stage Certificate Revocation Game is with incomplete information because the CL-UR agent does not have any information about the type of MNs belonging to its community. Hence, as a game of incomplete information, it can reach the BNE. To analyze our game, we rely on the fundamental observation of Harsanyi [135], where we introduce a virtual player called Nature, who makes the first move by selecting the type of the player X_S as shown in the extensive form of the game illustrated in Fig. 6.1. Therefore, the game becomes complete with imperfect information.

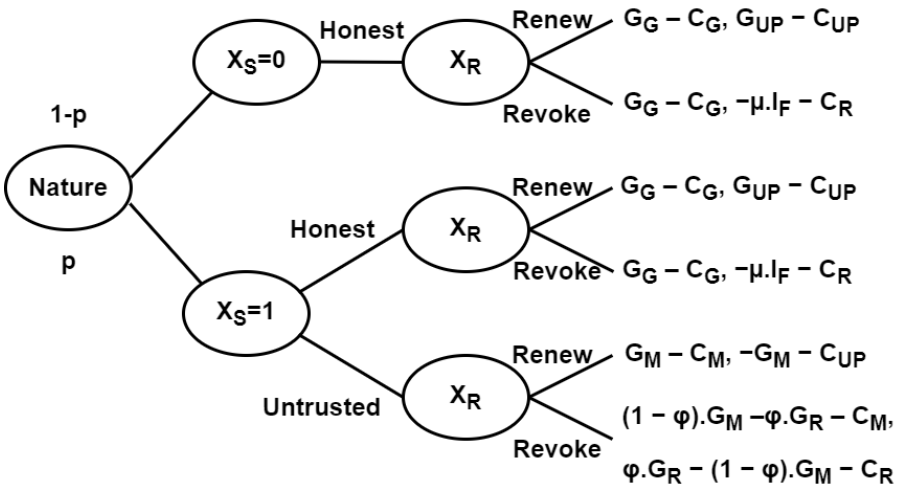


Figure 6.1: Extensive form of the stage Certificate Revocation Game

6.4.1 Pure-strategy BNE of the stage Certificate Revocation Game

- When the MN chooses the pure-strategy ($A_S(X_S = 0) = Honest\ behavior$, $A_S(X_S = 1) = Untrusted\ behavior$) which signifies that the MN always shows an untrusted behavior when it is untrusted and always shows an honest behavior when it is trusted, the expected utility of the CL-UR agent if it chooses $A_R(X_R) = Revoke$ as a strategy can be expressed as follows:

$$\begin{aligned} EU_R(Revoke) =& p(\varphi G_R - (1-\varphi)G_M - C_R)) \\ &+ (1-p)(-\mu I_F - C_R) \end{aligned} \tag{6.1}$$

The expected utility of the CL-UR agent if it chooses $A_R(X_R) = Renew$ as a strategy can be expressed as follows:

$$\begin{aligned} EU_R(Renew) =& (1-p)(G_{UP} - C_{UP}) \\ &+ p(-G_M - C_{UP}) \end{aligned} \tag{6.2}$$

If $EU_R(Renew) \leq EU_R(Revoke)$, i.e.,

$(1-p)(G_{UP} - C_{UP}) + p(-G_M - C_{UP}) \leq p(\varphi G_R - (1-\varphi)G_M - C_R)) + (1-p)(-\mu I_F - C_R)$ And,

$$p \geq \left(\frac{G_{UP} - C_{UP} + \mu I_F + C_R}{\varphi G_R + \varphi G_M + \mu I_F + G_{UP}} \right) \tag{6.3}$$

Therefore, ($A_R = Revoke$) is the dominant strategy for the CL-UR agent.

But if the CL-UR agent chooses Revoke as a strategy, ($A_S = Untrusted\ behavior$) will not be the dominant strategy for the malicious MN because:

$$(1-\varphi)G_M - \varphi G_R - C_M < G_G - C_G \tag{6.4}$$

A basic assumption of game theory is to consider that players are rational. Hence, the CL-UR agent and the MN are considered rational in our proposed game, i.e. they always aim to reach the best situation for them.

Consequently, the strategy profile $\{(A_S(X_S = 1) = Untrusted\ behavior, A_S(X_S = 0) = Honest\ behavior), A_R = Revoke\}$ is not a pure-strategy BNE.

- If $EU_R(Renew) > EU_R(Revoke)$, i.e.,

$(1-p)(G_{UP} - C_{UP}) + p(-G_M - C_{UP}) > p(\varphi G_R - (1-\varphi)G_M - C_R)) + (1-p)(-\mu I_F - C_R)$ And,

$$p < \left(\frac{G_{UP} - C_{UP} + \mu I_F + C_R}{\varphi G_R + \varphi G_M + \mu I_F + G_{UP}} \right) \tag{6.5}$$

Therefore, ($A_R = Renew$) is the dominant strategy for the CL-UR agent. Since $A_S(X_S = 1) = Untrusted\ behavior$ is the pure-strategy of the malicious MN, considered also as its dominant strategy when the CL-UR agent plays Renew because:

$$G_M - C_M > G_G - C_G \tag{6.6}$$

Then, the strategy profile $\{(A_S(X_S = 1) = Untrusted\ behavior, A_S(X_S = 0) = Honest\ behavior), (A_R = Renew)\}$ is a pure-strategy BNE.

- When the MN chooses the pure-strategy ($A_S(X_S = 1) = Honest\ behavior$, $A_S(X_S = 0) = Honest\ behavior$) which signifies that regardless of its type, it chooses to always behave honestly. Logically, the best strategy response for the CL-UR agent is to Renew, but in this case the best strategy of the malicious MN is to present untrusted behavior and not to behave honestly which is in contradiction with its pure-strategy.
Hence, the strategy profile $\{(A_S(X_S = 1) = Honest\ behavior, A_S(X_S = 0) = Honest\ behavior), (A_R = Renew)\}$ is not a pure-strategy BNE.

According to the analysis of the stage Certificate Revocation Game, we found a single pure-strategy BNE $\{A_S(X_S = 1) = Untrusted\ behavior$, $A_S(X_S = 0) = Honest\ behavior, (A_R = Renew)\}$. This pure-strategy BNE

exists when:

$$p < \left(\frac{G_{UP} - C_{UP} + \mu I_F + C_R}{\varphi G_R + \varphi G_M + \mu I_F + G_{UP}} \right) \tag{6.7}$$

However, this strategy is not practical because the CL-UR agent must always choose "Renew" as a strategy which will make the network vulnerable to untrusted nodes. Indeed, reaching only the pure-strategy BNE is not sufficient for our stage Certificate Revocation Game, it is crucial to find a mixed-strategy BNE to revoke effectively the certificates of malicious MN from the IIoT network.

6.4.2 Mixed-strategy BNE of the stage Certificate Revocation Game

Let $\Omega_S = (p_u, 1-p_u)$ and $\Omega_R = (\lambda, 1-\lambda)$ the mixed-strategies of the malicious MN and the CL-UR agent respectively. p_u designates the probability of the $X_S = 1$ playing $A_S(X_S = 1) = untrusted\ behavior$ and λ designates the probability of the CL-UR agent playing $A_R(X_R) = Revoke$.

Under the mixed-strategy Ω_R of the CL-UR agent, the expected utility of the MN resulting from $A_S(X_S = 1) = untrusted\ behavior$ as a strategy, $EU_S(Untrusted)$, is the sum of paths corresponding to this strategy in Fig. 6.1. It is calculated as follows:

$$\begin{aligned} EU_S(Untrusted) =& \lambda p((1-\varphi)G_M - \varphi G_R - C_M) \\ &+ p(1-\lambda)(G_M - C_M) \end{aligned} \tag{6.8}$$

Similarly, the expected utility of the MN resulting from $A_S(X_S = 1) = honest\ behavior$ as a strategy, $EU_S(Honest)$, is calculated as follows:

$$\begin{aligned} EU_S(Honest) =& \lambda p(G_G - C_G) + p(1-\lambda)(G_G - C_G) + \lambda(1-p)(G_G \\ &- C_G) + (1-p)(1-\lambda)(G_G - C_G) \end{aligned} \tag{6.9}$$

According to the indifference between untrusted and honest behavior under the mixed-strategy Ω_R, the expected utilities $EU_S(Honest)$ and $EU_S(Untrusted)$ are equal. Therefore, the equilibrium probability, λ^*, to play revoke is calculated as follow:

$$\lambda^* = \left(\frac{-G_G + C_G + p.G_M - p.C_M}{\varphi(G_M + G_R)} \right) \tag{6.10}$$

Under the mixed-strategy Ω_S of the MN, the expected utility of the CL-UR agent resulting from $A_R(X_R) = Revoke$ as a strategy, $EU_R(Revoke)$ is calculated as follows:

$$\begin{aligned} EU_R(Revoke) =& pp_u(\varphi G_R - (1-\varphi)G_M - C_R) \\ &+ p(1-p_u)(-\mu I_F - C_R) \\ &+ (1-p)(-\mu I_F - C_R) \end{aligned} \tag{6.11}$$

Similarly, the expected utility of the CL-UR agent resulting from $A_R(X_R) = Renew$ as a strategy, $EU_R(Renew)$ is:

$$\begin{aligned} EU_R(Renew) =& pp_u(-G_M - C_{UP}) + p(1-p_u)(G_{UP} - C_{UP}) \\ &+ (1-p)(G_{UP} - C_{UP}) \end{aligned} \tag{6.12}$$

According to the indifference between Renew and Revoke under the mixed-strategy Ω_S, the expected utilities $EU_R(Renew)$ and $EU_R(Revoke)$ are equal. Hence, the equilibrium probability, p_u^*, to play untrusted behavior is calculated as follow:

$$p_u^* = \left(\frac{-\mu I_F + C_R + G_{UP} - C_{UP}}{p(\varphi G_R + \varphi G_M + \mu I_F + G_{UP})} \right) \tag{6.13}$$

In the stage Certificate Revocation Game, there is a mixed-strategy BNE $\{\Omega_S(A_S(X_S = 1) = Untrusted\ behavior), A_S(X_S = 0) = Honest\ behavior,$ $\Omega_R(A_R = Revoke)\}$ which signifies that the malicious MN shows untrusted behavior with probability p_u^* and the legitimate MN always behaves honestly while the CL-UR agent plays Revoke with probability λ^*. This Mixed-strategy BNE exists when:

$$p \geq \left(\frac{G_{UP} - C_{UP} + \mu I_F + C_R}{\varphi G_R + \varphi G_M + \mu I_F + G_{UP}} \right) \tag{6.14}$$

As shown in 6.4.1 and 6.4.2, in the stage Certificate Revocation Game, players will choose their strategies to reach the BNE according to the value of the probability p. Since the Revocation gain and the untrusted behavior gain are very important, we expect the value of the probability p in the equation (6.7)

to be very small. We notice that as the value of p increases as shown in the equation (6.14), the players will be able to make better decisions and consequently reach the BNE in mixed-strategy.
p is the belief of the CL-UR agent on the untrusted type of the MN, the increase of this belief incites the untrusted MN to be less untrusted otherwise it will risk the revocation. The CL-UR agent needs to update its beliefs dynamically according to the real situation of the monitored MN, because determining a reasonable belief p at each individual stage is a challenging task. To address this challenge, in the next section we develop the stage Certificate Revocation Game into a multi-stage Game, in which the CL-UR agent dynamically updates its belief based on Bayesian rules.

6.5 Multi-stage dynamic Certificate Revocation Game

As the game evolves, the stage Certificate Revocation Game will be played successively and repeatedly over time, at each continuous time slot $\{t_j|j \in \{1,2,...,t\}\}$, where $(t \in Z^+)$, and during the network monitoring process. Hence, the Certificate Revocation Game becomes a multi-stage game, where the players observe the results of each stage game before the next one starts. This enables the CL-UR agent to predict the type of the MN and condition its future strategies on previous results. The multi-stage certificate revocation Game is also defined by five tuple (PL, χ, A, P, Uf) where:

- PL, A, χ, and Uf are defined as for the Stage Certificate Revocation Game described in the previous section. We assume that the utilities of both players are the same in all stage games.
- $P = (p(X_S = 1|h(t_j)), 1 - p(X_S = 1|h(t_j))$ where $p(X_S = 1|h(t_j))$ is the probability that the MN is malicious considering the history of its behavior $h(t_j)$. $p(X_S = 1|h(t_j))$ is updated by the posterior belief of the CL-UR agent, $p(X_S = 1|a_S(t_j), h(t_j))$, calculated by the equation (6.15) at the end of the t_jth stage game. The process of updating the belief of the CL-UR agent at the t_jth stage game can be led from the $(t_{j-1})th$ stage game by using Bayes rules [136].

$$p(X_S = 1|A_S(t_j), h(t_j)) = \left(\frac{p(X_S = 1|h(t_j))p(A_S(t_j)|X_S = 1, h(t_j))}{\sum_{X'_S \epsilon \chi_S} p(X'_S|h(t_j)).p(A_S(t_j)|X'_S, h(t_j))} \right) \tag{6.15}$$

Where $A_S(t_j)$ is the behavior of the MN at the t_jth stage game, and $p(A_S(t_j)|X_S, h(t_j))$ is the probability that MN presents the behavior $A_S(t_j)$ at the t_jth stage game under the history $h(t_j)$. The different probabilities $p(A_S(t_j)|X_S, h(t_j))$ can be calculated as follows:

$$p(A_S(t_j) = Untrusted|X_S = 1, h(t_j)) = \varphi.p_u + \mu(1 - p_u) \quad (6.16)$$

$$p(A_S(t_j) = Honest|X_S = 1, h(t_j)) = (1 - \varphi).p_u + (1 - \mu)(1 - p_u) \quad (6.17)$$

$$p(A_S(t_j) = Untrusted|X_S = 0, h(t_j)) = \mu \quad (6.18)$$

$$p(A_S(t_j) = Honest|X_S = 0, h(t_j)) = 1 - \mu \quad (6.19)$$

We consider the effect of the false negative rate, $1-\varphi$, and the true negative rate, $1-\mu$, respectively, when calculating the different probabilities $p(A_S(t_j)|X_S, h(t_j))$.

The CL-UR agent evaluates the probability p_u that the monitored MN presents an untrusted behavior as a follows:

$$p_u = 1 - (p_c.p_d.p_I) \quad (6.20)$$

p_c is the probability that the monitored MN cooperates in the network, p_d is the probability that the monitored MN is honest and p_I is the probability that the monitored MN has a good reputation in the network. The calculation of these three probabilities is detailed in chapter 5.

In the proposed multi-stage Certificate Revocation Game, players do not always adopt the same strategy at each stage. Indeed, players condition their future behaviors on previous results and current beliefs to produce the most utility and choose the best response strategy as the game evolves. Thus, we characterize our proposed multi-stage game by the PBE which provides each player with a complete system of beliefs about the type of its opponent in order to choose its best response strategy.

Before searching for the PBE of the multi-stage Certificate Revocation Game, we have to prove first that the proposed game satisfies the following Bayesian conditions [136]:

- **C1**: The beliefs are updated by the Bayesian rule.
- **C2**: The posterior beliefs are independent, and the different types of the same player must have the identical beliefs.

- **C3**: The posterior belief is coherent with a common joint distribution [137] in χ.
- **C4**: The players signal just what they know and in no case they signal what they don't know.

Lemma. The proposed certificate revocation game satisfies the Bayesian conditions.

Proof. **C1** is verified; the belief of the CL-UR agent on MNs is updated using the Bayesian rule as mentioned in the equation (6.15). **C2** is verified since the CL-UR agent in our game has only one type, it will always have the same belief on a MNs. **C3** is verified as the certificate revocation game consists of only 2 opponents, no other player can influence the belief of the CL-UR agent on the MNs. **C4** is verified because the signal of MN is defined only by its behavior, $p(A'_S(t_j)|X_S = 1, h(t_j)) = p(A_S(t_j)|X_S = 1, h(t_j))$ when $A'_S(t_j) = A_S(t_j)$. □

6.6 The PBE of the Certificate Revocation Game in mixed-strategy

Theorem. The proposed multi-stage Certificate Revocation Game has a mixed-strategy PBE.

Proof. Let $\Omega_{Sj} = (p_{uj}, 1-p_{uj})$ and $\Omega_{Rj} = (\lambda_j, 1-\lambda_j)$ the mixed-strategies of the MN and the CL-UR agent respectively at the t_jth stage game. p_{uj} designates the probability of the MN playing $A_{Sj}(X_S = 1)$ = untrusted behavior at the t_jth stage game and λ_j designates the probability of the CL-UR agent playing $A_{Rj}(X_R)$ = Revoke at the t_jth stage game.

The mixed-strategy PBE can be achieved in the multi-stage Certificate revocation game if the CL-UR agent and the MN play respectively with the equilibrium strategy profile $\Omega^*_{Sj} = (p^*_{uj}, 1 - p^*_{uj})$ and $\Omega^*_{Rj} = (\lambda^*_j, 1 - \lambda^*_j)$ at the t_jth stage game. The main purpose is to calculate the different equilibrium probabilities p^*_{uj} and λ^*_j that define the best response behavior of each player at the t_jth stage game.

Under the mixed-strategy Ω_{Rj} of the CL-UR agent, the expected utility, $EU_S(Untrusted)$, of the MN resulting from the choice of $A_{Sj}(X_S = 1)$ = *untrusted behavior* as a strategy at the t_jth stage game is calculated as follows:

$$\begin{aligned} EU_S(Untrusted) =& \lambda_j p(X_S = 1|h(t_j))((1-\varphi)G_M - \varphi G_R \\ & - C_M) + p(X_S = 1|h(t_j))(1-\lambda_j) \\ & (G_M - C_M) \end{aligned} \tag{6.21}$$

Similarly, the expected utility, $EU_S(Honest)$, of the MN resulting from $A_{Sj}(X_S = 1) = honest\ behavior$ as a strategy at the $t_j th$ stage game is calculated as follows:

$$\begin{aligned} EU_S(Honest) =& \lambda_j p(X_S = 1|h(t_j))(G_G - C_G) \\ & + p(X_S = 1|h(t_j))(1-\lambda_j)(G_G - C_G) \\ & + \lambda_j(1 - p(X_S = 1|h(t_j)))(G_G - C_G) \\ & + (1 - p(X_S = 1|h(t_j)))(1-\lambda_j)(G_G - \\ & C_G) \end{aligned} \tag{6.22}$$

According to the indifference of untrusted and honest behavior under the mixed-strategy Ω_{Rj}, the expected utilities $EU_S(Honest)$ and $EU_S(Untrusted)$ are equal. Therefore, the equilibrium probability, λ_j^*, to play revoke at the $t_j th$ stage game is calculated as follow:

$$\lambda_j^* = \left(\frac{-G_G + C_G + p(X_S = 1|h(t_j))G_M - p(X_S = 1|h(t_j))C_M}{p(X_S = 1|h(t_j))(\varphi G_M + \varphi G_R)} \right) \tag{6.23}$$

Under the mixed-strategy Ω_{Sj} of the MN, the expected utility $EU_R(Revoke)$ of the CL-UR agent resulting from the choice of $A_{Rj}(X_R) = Revoke$ as a strategy at the $t_j th$ stage game is:

$$\begin{aligned} EU_R(Revoke) =& p_{uj} p(X_S = 1|h(t_j))(\varphi G_R - (1-\varphi)G_M - C_R) \\ & + p(X_S = 1|h(t_j))(1 - p_{uj})(-\mu I_F - C_R) \\ & + (1 - p(X_S = 1|h(t_j)))(-\mu I_F - C_R) \end{aligned} \tag{6.24}$$

Similarly, the expected utility, $EU_R(Renew)$, of the CL-UR agent resulting from the choice of $A_{Rj}(X_R) = Renew$ as a strategy at the $t_j th$ stage game is calculated as follows:

$$EU_R(Renew) = p_{uj}p(X_S = 1|h(t_j))(-G_M - C_{UP}) + p(X_S = 1|h(t_j))(1 - p_{uj}) (G_{UP} - C_{UP}) + (1 - p(X_S = 1|h(t_j)))(G_{UP} - C_{UP}) \tag{6.25}$$

According to the indifference of Revoke and Renew under the mixed-strategy Ω_{Sj}, the expected utilities $EU_R(Renew)$ and $EU_R(Revoke)$ are equal. Hence, the equilibrium probability, p^*_{uj}, to play untrusted behavior at the t_jth stage game is calculated as follow:

$$p^*_{uj} = \left(\frac{-\mu I_F + C_R + G_{UP} - C_{UP}}{p(X_S = 1|h(t_j))(\varphi G_R + \varphi G_M + \mu I_F + G_{UP})} \right) \tag{6.26}$$

□

6.7 The proposed certificate revocation mechanism based on the PBE of the multi-stage Game

We propose an effective certificate revocation mechanism and its corresponding algorithm based on the PBE of the multi-stage game obtained in section 6.6. In Fig. 6.2, we detail in a modular way the proposed mechanism as well as the interactions between the CL-UR agent and the MN. Initially, the CL-UR agent performs the monitoring process in which it calculates the trust probability pu of the MN in order to define the type of its behavior, trusted or untrusted. Afterwards, to make a revocation decision, the agent enters into a game with the MN. As the game evolves, the belief of the CL-UR agent on the MN evolves as well. Based on this belief, the CL-UR agent can choose its best response strategy at the t_jth stage game against the MN, according to the probability λ^*_j. At the end of each stage game, the CL-UR agent evaluates its posterior belief on the MN in order to update its prior belief for the next stage game. To illustrate this, the Algorithm 5 describes the whole certificate revocation process.

Indeed, the CL-UR agent initiates its monitoring process according to lines 2-4. Thereafter, it gets the game parameters (lines 5-7), and the prior belief on the MN (line 8) from the IIoT server. Based on the results of the monitoring process, the prior belief and the game parameters, the CL-UR agent

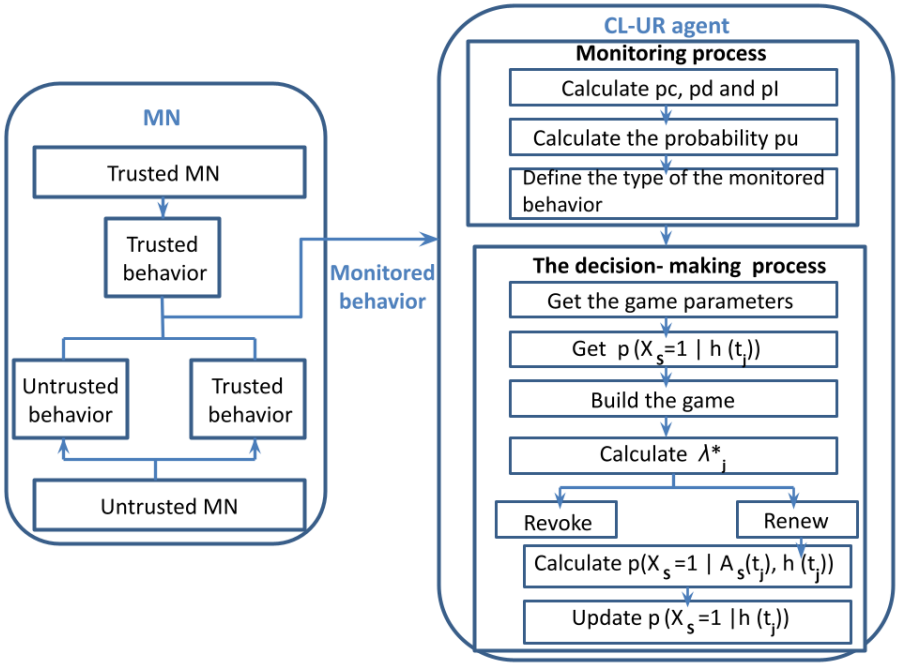


Figure 6.2: The certificate revocation mechanism

defines its strategy ("renew certificate" or "revoke") according to the equilibrium probability λ_j^* (lines 8-10). If the strategy chosen by the CL-UR agent at the $t_j th$ stage game is to renew the certificate of the MN, the agent updates its prior belief on MN (line 19) by evaluating its posterior belief (line 18) for subsequent use in the next stage game (line 20).

6.8 Performance evaluation of the certificate revocation mechanism

In this section, we evaluate the performance of our proposed certificate revocation mechanism and we compare the results obtained, in the presence of a variable number of malicious nodes, with those of the voting and non-voting mechanisms. For this purpose, we use the InstantContiki 2.7 platform. The different simulation parameters used in this chapter are the same as those used in chapter 5, listed in Table. 5.1. We set the parameters of the Certificate Revocation Game as follows: $G_G = 15$, $C_G = 5$, $G_{UP} = 15$, $C_{UP} = 10$, $G_R = 150$, $C_R = 20$, $G_M = 200$, $C_M = 20$, $I_F = 15$. Also, we assume that the initial prior belief of the CL-UR agent on all MNs is equal to 0.5.

Fig. 6.3 evaluates the convergence speed of the posterior belief, $p(X_S = 1|A_S(t_j), h(t_j))$, of the CL-UR agent on three MN nodes:

- In Fig. 6.3a, we consider a malicious node MN_1, with p_u=0.9, that

Algorithm 5 Certificate revocation algorithm

Begin
1. **WHILE** t
2. Monitor the behavior of the MN;
3. Calculate the probability pu from the probabilities p_c, p_d and p_I;
4. Define the type of the monitored behavior;
5. **IF** the game is not existed **THEN**
6. Get the game parameters : G_G, C_G, G_M, C_M, μ, φ, C_R, G_{UP}, C_{UP}, G_R and I_F, from the IIoT server.
7. **ENDIF**
8. Get the prior probability $p(X_S = 1|h(t_j))$ on the MN;
9. Calculate λ_j^*;
10. Choose the best strategy against the MN according to the probability λ_j^*;
11. **IF** Choose Revoke **THEN**
12. Revoke the $MN's$ certificate;
14. **ELSE IF** Choose Renew **AND** time-out-certificate is expired **THEN**
15. Renew the $MN's$ certificate;
16. **ELSE** keep the previous certificate;
17. **ENDIF**
18. Calculate $p(X_S = 1|A_S(t_j), h(t_j))$;
19. Update $p(X_S = 1|h(t_j))$ with $p(X_S = 1|A_S(t_j), h(t_j))$;
20. Store $p(X_S = 1|h(t_j))$ for the next stage game;
21. **END WHILE**

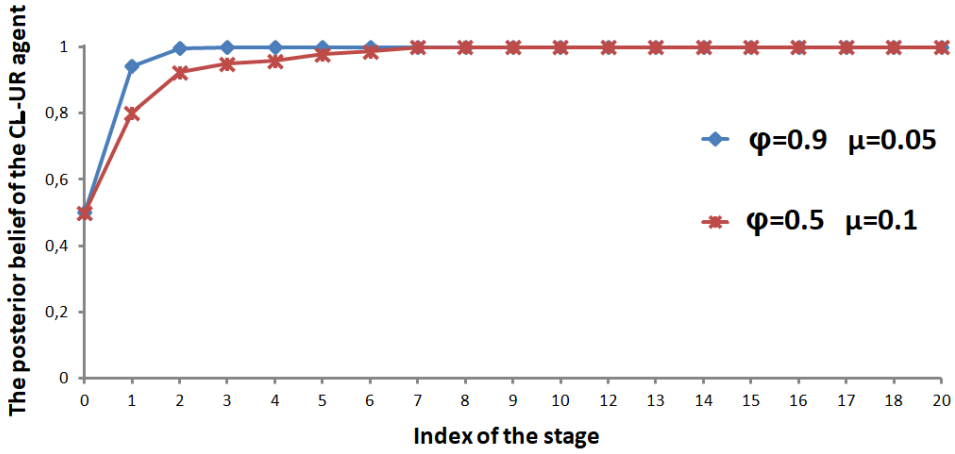
End

presents only untrusted behavior.

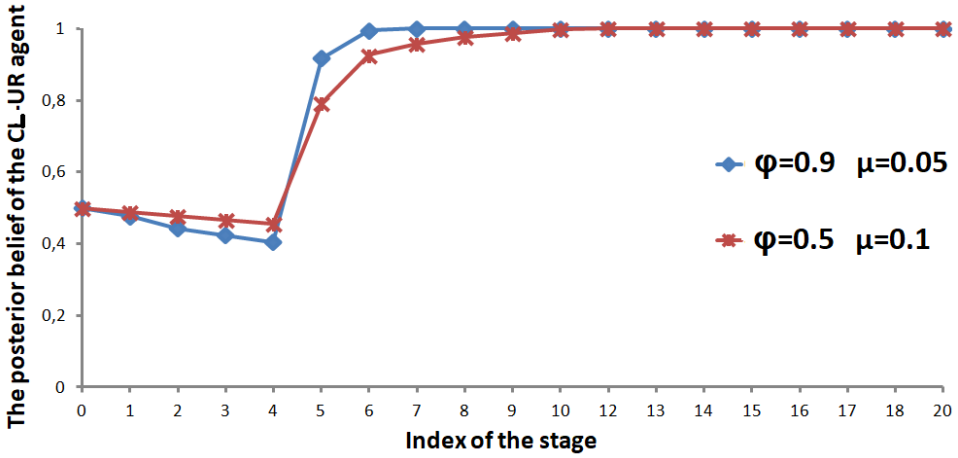
- In Fig. 6.3b, we consider a node MN_2 that presents honest behavior until the 4 th stage game with $p_u = 0.1$ and then it decides to change its behavior and become untrusted with p_u= 0.9.

- In Fig. 6.3c, we consider a node MN_3 that performs an on-off attack. MN_3 shows an honest behavior with $p_u = 0.1$ during the first 5 stages of the game, then it changes its honest behavior into untrusted behavior with $p_u = 0.9$ during the next 5 stages of the game followed by honest behavior during 5 stages of the game and so on.

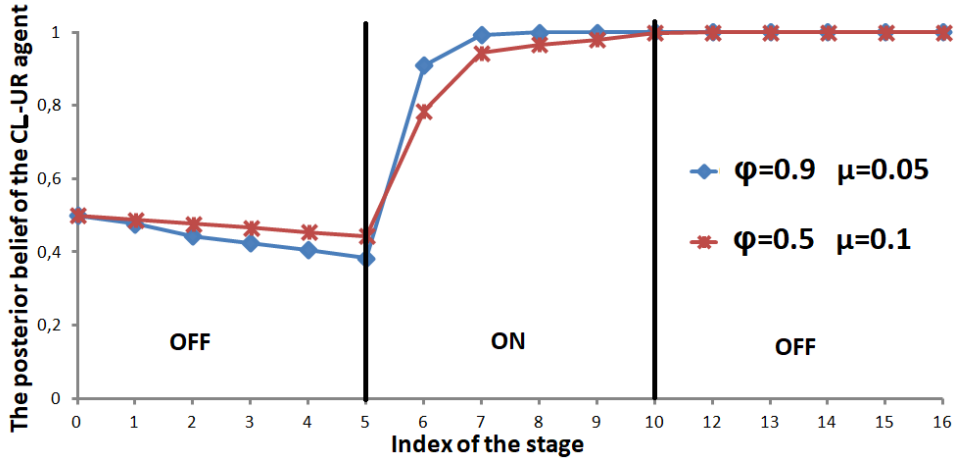
As shown in Fig. 6.3a, the posterior belief allowing the CL-UR agent to decide if the MN_1 is untrusted converges quickly to 1, especially when the detection rate is high and the false alarm rate is low (φ=0.9, μ=0.05). This demonstrates the speed of our proposed model to detect and revoke malicious nodes from the network. Our model is also sensitive to behavior changes as shown in Fig. 6.3b and Fig. 6.3c, it promptly and accurately detects the behavior changes of nodes MN_2 and MN_3 at the 4 th and 5 th stage game respectively. Indeed, the posterior belief of the CL-UR agent converges quickly to 1 when nodes MN_2 and MN_3 begin to behave unreliably. We can see in Fig. 6.3c that even when the MN_3 returns to its previous honest behavior to disguise its untrusted behavior, the posterior belief of the CL-UR agent on it



(a) MN_1



(b) MN_2



(c) MN_3

Figure 6.3: The convergence speed of the $p(X_S = 1|A_S(t_j), h(t_j))$ according to φ and μ

does not decrease and remains fixed at 1. This leads to confirm that our model

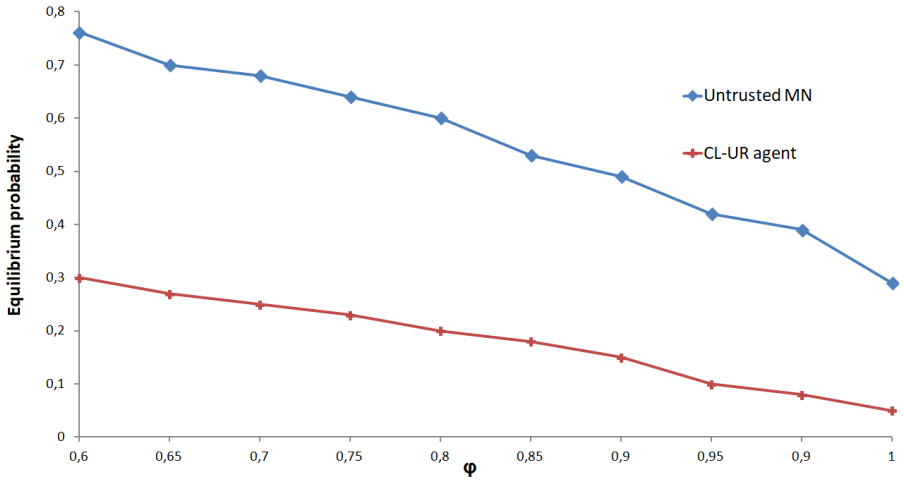


Figure 6.4: Change trend of p^*_{uj} and λ^*_j according to φ

deals with on-off attacks.

With a lower detection rate and a higher false alarm rate (φ=0.5, μ=0.1), the convergence speed of the posterior belief becomes slower. This is due to the fact that the CL-UR agent is not sufficiently sensitive in such a case. The CL-UR agent can improve its decision making by improving its detection rate and reducing its false alarm rate.

We assume that when φ= 0.6, the equilibrium probability of the MN choosing untrusted behavior as a strategy is 0.76 and the CL-UR agent choosing Revoke as a strategy is 0.3.

In Fig. 6.4, we study the change trend of these equilibrium probabilities when the value of φ ranges from 0.6 to 1. According to Fig. 6.4, we notice that the increase of φ involves the decrease of the equilibrium probabilities p^*_{uj} and λ^*_j. With the increase of the detection rate, φ, the MN will no longer have interest to perform untrusted behaviors in the network because it will be detected and revoked, which explains the decrease of its equilibrium probability p^*_{uj}. When the malicious MN behaves honestly in the network and does not perform untrusted behavior, the posterior belief of the CL-UR on this node decreases which explains the decrease of the equilibrium probability λ^*_j. Hence, the CL-UR agent should always increase the rate φ to revoke malicious nodes and thus improve the performance of the IIoT network.

In Fig. 6.5, we compare the revocation rate of our proposed mechanism with that of the voting and non-voting revocation mechanisms, for different percentage of malicious nodes. As shown in Fig. 6.5, with the increase of malicious nodes, the revocation rate decreases significantly in the voting mechanism compared to our mechanism. Indeed, in the voting mechanism, the increase of

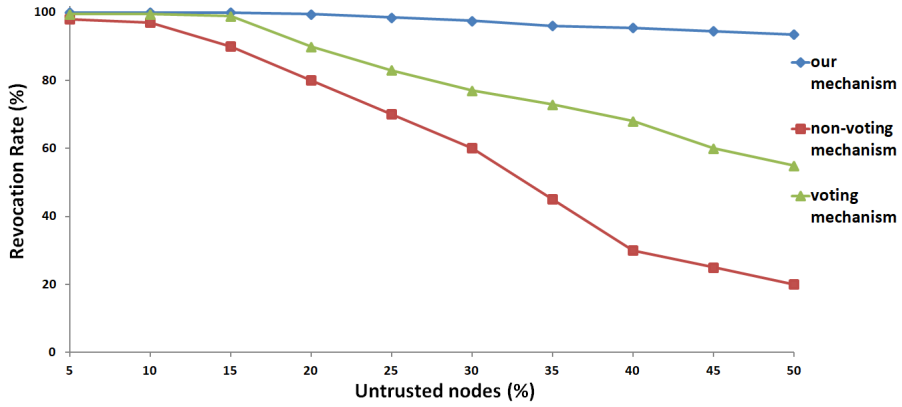


Figure 6.5: Revocation rate vs malicious nodes

malicious nodes increases the risk of coalition attacks. Malicious nodes form coalitions and do not vote against each other to protect themselves from revocation. The revocation rate decreases further in the non-voting revocation mechanism because the increase of malicious nodes in the network increases the false accusations against honest nodes. Malicious nodes can send false accusations against honest nodes to isolate them from the network, therefore there would not be enough honest nodes in the network to make accusations. Our proposed mechanism outperforms related mechanisms despite the increase of the malicious nodes in the network. This is due to the fact that our mechanism relies on the trusted CLs to make revocation decisions and not on normal nodes in the network that are susceptible to be untrusted.

Another crucial parameter for evaluating and validating the performance of revocation mechanisms is the revocation time. It is calculated as the average time required to revoke the certificate of a malicious node. For this purpose, in Fig. 6.6, we evaluate the revocation time of our proposed mechanism when the percentage of malicious nodes increases. We also compare the obtained results with those of voting and non-voting revocation mechanisms.

According to Fig. 6.6, we notice that non-voting mechanism is the fastest when the percentage of malicious nodes is low. This is because revoking a node's certificate in this mechanism requires only one accusation. Afterwards, the revocation time increases with the increase of malicious nodes that, thanks to Bad-mouthing attacks, they revoke the honest nodes responsible for the revocation. Voting mechanism needs more time to do revocations even if the percentage of malicious nodes does not exceed 30%. Indeed, to revoke the certificate of a malicious node, it must have a necessary number of votes against it. The revocation time becomes huger when the percentage of malicious nodes increases in the network. This is explained by the fact that voting mechanism

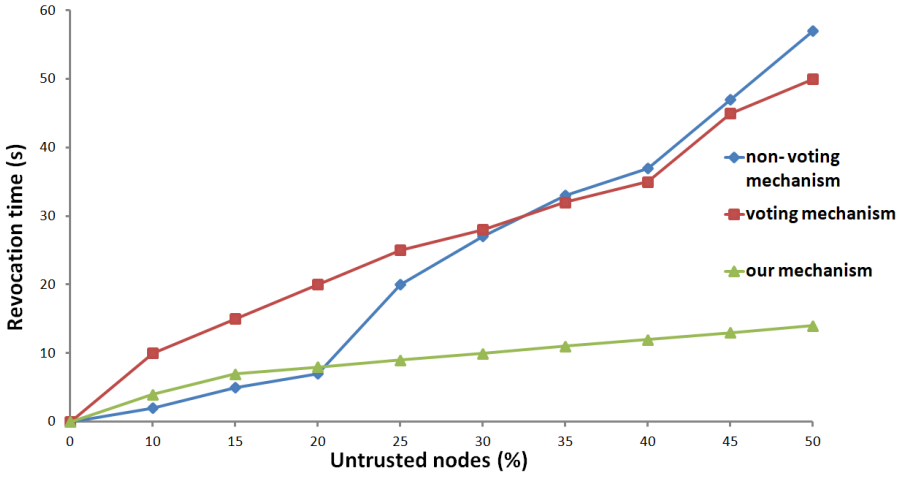


Figure 6.6: Revocation time vs malicious nodes

is not able to identify and revoke a malicious node instantly if there are not enough honest nodes around it, which affects the reliability of the network.

6.9 Conclusion

The proposed multi-stage Certificate Revocation Game allows the CL-UR agent to predict its strategy and make rational decisions based on the behavior of MNs in the network. Based on the PBE of the multi-stage game, the CL-UR agent can accurately revoke the certificate of malicious nodes to enhance the security of the IIoT network.

When a certificate is revoked, it is important to inform other nodes in the network to ensure that they do not establish a connection with its owner. However, distributing the revocation information while considering the constraints and requirements of IIoT networks is a challenging task. This is the purpose of the next chapter.

Chapter 7

Trust-based Certificate Management for IIoT networks

Contents

7.1 Introduction 124

7.2 Related work 125

7.3 Proposed architecture 127

7.4 SLC-based certificate verification scheme 129

7.4.1 SLC as an alternative to the conventional certificate verification schemes 130

7.4.2 Workflow of our Certificate verification scheme 132

7.5 Performance Evaluation 132

7.5.1 Performance evaluation of the SLC management process 133

7.5.2 Performance evaluation of the SLC-based certificate verification scheme 136

7.6 Security Analysis 139

7.7 Conclusion 141

7.1 Introduction

Currently, the role of IIoT is not only limited to sending to the cloud data collected with sensors located on industrial equipment, but much more than that. In fact, IIoT may applies artificial intelligence algorithms on the data collected during the monitoring process to reach a higher maturity layer allowing the IIoT devices to communicate [1] and collaborate [143] with each other in all stages of production, in real time and without human interventions. The automation and autonomy of IIoT devices increase significantly the risk of communication with malicious devices. Thus, IIoT devices must be informed immediately when the certificate [138] of a device is revoked from the network. Otherwise, they will risk establishing a communication with a malicious device that will pretend to be trusted. The information about revoked certificates allows IIoT devices to verify the revocation status of their peers' certificates before establishing a communication with them. However, the distribution of this information in the IIoT network in a resource efficient manner and without latency is a difficult task.

Conventional techniques for distributing information about revoked certificates, also known as certificate verification schemes, are the Certificate Revocation List (CRL), the Online Certificate Status Protocol (OCSP), and the Bloom filter. These schemes are not appropriate to be applicable in IIoT networks because they are subject to some issues like communication overhead, storage overhead and latency. For this purpose, in this chapter, we propose a new efficient certificate verification scheme based on Short-Lived Certificates (SLCs), and adapted to the requirements of IIoT networks. With the shortened validity of SLCs, the user would not have to worry about verifying the revocation status of a certificate by using conventional schemes, but rather rely on the expiration date indicated on the certificate itself. This avoids the issues related to the use of conventional schemes and supports real-time communication in IIoT environments by eliminating the latency resulting from the verification process.

The remainder of this chapter is organized as follows. In section 7.2, we review the existing certificate verification systems in the literature, we also provide a critical analysis of why these schemes are not suitable for deployment in IIoT networks. In section 7.3, we detail the improvements we apply to the hierarchical architecture proposed in chapter 5 to support the management of SLCs in IIoT networks. Based on the research gaps and open challenges discussed

[1] Data Captured by IoT Connections to Top 1.6 Zettabytes in 2020, As Analytics Evolve from Cloud to Edge

in section 7.2, in section 7.4, we propose a new efficient certificate verification scheme based on SLCs. The proposed scheme is suitable for the IIoT network requirements. We present the performance evaluation and comparison results in section 7.5. In section 7.6, we provide a security analysis of our proposed certificate verification scheme against a set of potential security threats and vulnerabilities. Finally, in section 7.7, we conclude the chapter.

7.2 Related work

In this section, we provide a critical analysis of existing certificate verification schemes in the literature, their advantages, limitations and deployability in IIoT networks.

The well known certificate verification schemes in the literature are those based on the CRL. The CRL operates as a blacklist where the certificate authority (CA) periodically publishes certificates that are no longer trustworthy. Thus, each node must download this list and check if it contains the serial number of its peer's certificate before establishing a communication with it. The CRL-based schemes prevent communications with malicious nodes, however, they require huge processing and consume the storage of nodes because the size of the CRL increases as the number of revoked certificates in the network increases. Authors of [146] proposed a distributed CRL management scheme based on distributed hash trees. [146] enhanced the distribution and storage cost of conventional CRL-based schemes by sharing the burden of CRL storage among all nodes. If a node wants to check the revocation status of a certificate, first, it calculates the hash of the certificate's identity as a key. Afterwards, it uses this key to find the node in the network that contains the certificate status. This proposal is vulnerable to trust attacks because the requested node may indicate that a certificate is revoked when it is not, or the reverse. Authors of [147] proposed a semi decentralized public key management scheme for IoT systems, in which nodes decide whether to look for the revocation information locally at the edge or refer to the CA hosted in the cloud. The decision of each node is made intelligently by estimating the cost of each approach. In the first approach, each node uses its trust table to contact trusted nodes that can help it to make verification. If the estimations are incorrect, i.e. the revocation status of the certificate is not found among the CRL of nodes listed in the trust table, the node must pay an additional cost of verification through CA. Therefore, it is necessary to make the right estimation to avoid excess costs. In [147], trust tables are static, they do not consider the dynamic behavior of

nodes where even trusted nodes can become untrusted and selfish over time.

To solve the resource consumption issues of CRL-based schemes, researchers developed the OCSP protocol [148]. Hence, to verify the revocation status of a certificate, the node must send a request to an OCSP server and wait for the response. This creates latency and communication overhead. To enhance the performance of OCSP, authors of [149] proposed the OCSP stapling protocol that allows each node to ask its peers during the handshake phase to send it valid OCSP responses with their certificates. However, this protocol does not guarantee that the peer node will always transmit a valid OCSP response. Authors of [150] proposed a certificate validation scheme for Mobile Ad-hoc Networks called Ad-hoc Distributed OCSP (ADOPT). ADOPT adjusts the validity period of the OCSP response according to the trust level of the certificate owner. The proposed scheme uses the caching to provide certificate status information (CSI) even in the off-line states to the nodes. However, the CSI refreshing method used by [150] to reduce the incoherence of CSI causes a significant overhead to the network.

To compress the size of the CRLs and solve the limitations of the OCSP, several works such as [145], [151] and [152] proposed certificate verification schemes based on Bloom filter [153]. Bloom filter is a probabilistic data structure that allows to verify the presence of an element in a set. The limitation of Bloom filter is that the number of false positives is proportional to the number of elements in the structure. Authors of [158] relied on the advantages of blockchain to propose a decentralized certificates revocation management and status verification system. In [158], the CA combines in a data structure called Revocation Status Information (RSI) the identities of revoked certificates and the Bloom filter containing the revoked certificates. The certificate of each entity in the network contains a field indicating to which RSI its state will belong in case of its revocation. RSI structures are recorded by the CA in a public blockchain to be available to entities. During the handshake phase, the node receives not only the certificate of its peer but also the RSI allowing it to perform the verification. Thus, the node first ensures that it has the correct RSI that contains the revocation information of its peer, then it extracts the Bloom filter and verifies the belonging of the certificate to the filter. A negative response ensures that the certificate has not been revoked. However, when the response is positive, the node performs an additional investigation to ensure that the response is not a false positive. For this purpose, the node requests its peer to send it all the revocation information issued by the CA. Such a process generates additional communication costs and increases the latency of

verification.

Several works ([139], [140], [141], [142], and [143]) relied on SLCs as an alternative to conventional certificate verification schemes. However, SLCs can convert the certificate management process into a bottleneck because they present performance and organizational challenges. They require sophisticated automation and frequent reissuance, making their use in IIoT networks constrained despite their support for automation. To address the challenges of SLCs, authors of [140] shared the burden of SLCs management among a set of decentralized CAs. The work in [140] used SLCs to address service integrity and metadata protection, but not for authentication between distributed nodes. Authors of [143] used SLCs to support authentication between distributed devices. They rely on the strength of fog computing to make their solution scalable and meet the challenges related to the use of SLCs. However, the certificate validity period in [143] is considered the same for all devices of the network, which can create a large workload if all devices synchronously request a new certificate at the same time.

Based on the weaknesses of the existing certificate verification schemes discussed above, we propose a new efficient SLC-based certificate verification scheme suitable for IIoT network requirements. The proposed solution addresses challenges due to the use of SLCs and supports automation in IIoT environments. We also introduce the notion of trust to determine the validity period of SLCs. Indeed, the validity period of each SLC is proportional to the trust level of its owner so that only the SLCs of less trusted nodes will be updated frequently. To the best of our knowledge, our work is the first to integrate the notion of trust in the calculation of the validity period of SLCs. This makes a good trade-off between certificate life and overheads resulting from certificate renewal process, while keeping a high security level.

7.3 Proposed architecture

In this section, we detail the improvements we apply to the hierarchical architecture proposed in chapter 5 to support and enhance the SLCs management in IIoT networks.
A single network entity "CA" cannot manage alone the certificate of all IIoT devices in the network, as it is not able to accurately monitor the behavior of these devices. In an IIoT network, it is necessary to detect malicious devices promptly and accurately during the revocation phase. This was the purpose of the chapter 6 where we proposed an effective certificate revocation mecha-

nism for IIoT networks. It is also important to quickly inform other devices of revoked certificates to avoid establishing a communication with their owners, while considering the constraints and requirements of the IIoT network.This is the purpose of this chapter.
For this purpose, we use the hierarchical architecture proposed in chapter 5 and illustrated in Fig. 5.1. We remind the reader that the hierarchical architecture organize the IIoT network as a set of clusters called industrial communities. Communities are dynamically formed on the basis of the industrial relationships between IIoT devices. Each community consists of a trusted and powerful CL and a set of MNs. The CL-UR agent, hosted in the CL, revokes the certificate of malicious MNs, renews those of legitimate nodes and shares the revocation result with the rest of the network. Indeed, the CL acts as a distributed CA. The hierarchical architecture makes our proposed verification scheme scalable and prevents bottleneck due to frequent reissuance of SLCs by sharing the burden of certificate management among all CLs.

To achieve continuous monitoring, the CL evaluates frequently the behavior and the trust of MNs belonging to its community. Hence, after each period of time, the CL records in a database hosted in the cloud the IDs of revoked MNs in its community as well as the behaviors of the non-revoked MNs to use them as a history for its next monitoring. These records must be shared with all other CLs in the network. Consequently, if a MN changes community, the new CL that has no information about the behavior of this MN can rely on the information shared by the old CL. Hence, the new CL provides the MN with a valid certificate whose validity period is proportional to its behavior, or refuses the integration of the MN to the community if it was already revoked by its old CL. This prevents several attacks such as the whitewashing attack that occurs when a MN leaves its community and joins another once its untrusted behavior is detected. To share the monitoring and revocation information with other CLs in the network, we propose to use a consortium blockchain [156], where after each period of time, the CL writes a transaction as described in Fig. 7.1:

Each transaction encapsulates the following information:

- ID_{CL}: The unique identifier of the CL.
- Pb_{CL}: The public key of the CL.
- $Uniform\ resource\ identifier$ (URI): points in the database the location where the CL has transcribed the last monitoring and revocation data related to its community.

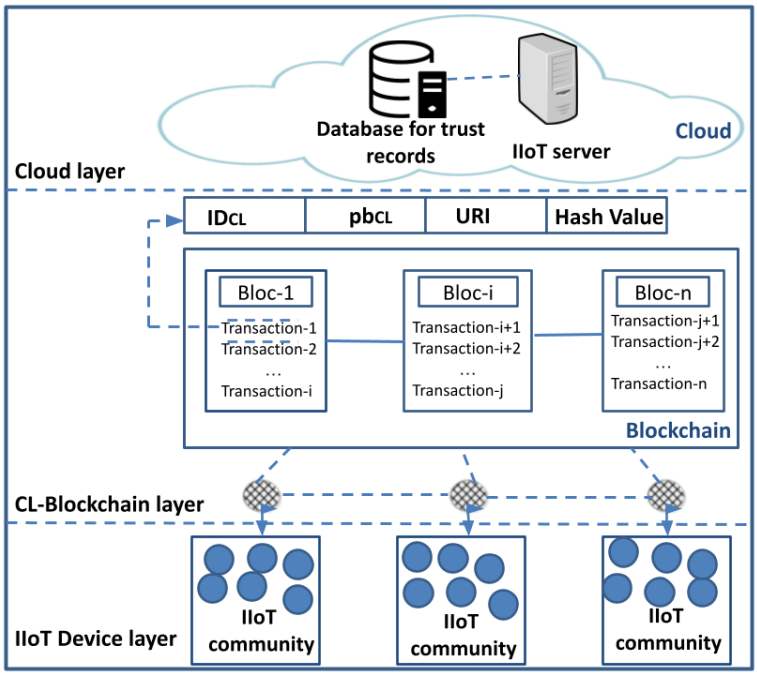


Figure 7.1: Transaction written by the CL in the blockchain after each monitoring or revocation.

- *Hash Value*: is the hash of the file containing the latest updated monitoring and revocation data related to the community. Since this file is stored outside the blockchain, in the database, it could be potentially modified by attackers. The Hash Value allows to verify the authenticity of the file.

We use the blockchain to secure exchanges between communities. Indeed, the blockchain allows a node to check the validity of certificates sent by nodes belonging to other communities before establishing communication with them, as detailed in section 7.4. The blockchain also keeps track of the behavior of nodes, even when they change communities. Indeed, CL_j access the latest updated monitoring and revocation data of the community "$C_{j'}$" by using the URI mentioned by the leader of $C_{j'}$ in the blockchain. We use the URI to avoid storing monitoring data at CLs and overloading the blockchain, hence, all records are made in the database hosted in the cloud. The IIoT server is the entity responsible for validating blocks in the blockchain.

7.4 SLC-based certificate verification scheme

Based on the research gaps and open challenges discussed in section 7.2, in this section we propose an efficient certificate verification scheme for IIoT networks, based on SLC.

7.4.1 SLC as an alternative to the conventional certificate verification schemes

IIoT devices are led to communicate and collaborate with each other at all stages of production, in real time and without human intervention. To prevent communication with malicious devices, each device must possess the information about the revoked certificates in the network to verify the certificate revocation status of its peer. IIoT networks are heterogeneous, thus, the revocation verification process in these environments should consider the processing power, storage, battery lifetime and bandwidth of resource-constrained devices such as sensors and must not waste the resources of powerful devices such as connected robots and machines. The revocation verification process must also be done without latency. Hence, to address these challenges, we propose a new efficient certificate verification scheme based on SLCs.

Indeed, SLC is similar to an ordinary certificate; it can be deployed and chained in the same way as the existing X.509 certificates, except that its validity period is a short span of time which can be few hours or few days. Despite the advantages of SLCs and their support for automation in IIoT networks, they still present performance and organizational challenges that make their use constrained. SLCs can transform the certificate distribution process into a bottleneck, as a high number of certificates must be issued each period of time. To enable our certificate verification scheme to meet these challenges and take full advantage of SLCs, we rely on the architecture described in section 7.3. The proposed architecture shares the burden of SLCs management among all CLs, instead of it being handled by a single entity in the network. It also brings the certificate verification process closer to MNs, which improves the real-time security services for the IIoT network.

SLCs allow to avoid the limitations of conventional certificate verification schemes. They eliminate storage, communication and processing overheads at MNs during the certificate verification process. Indeed, MNs don't need to store revocation lists which size increases with the number of revoked certificates in the network, they don't have to worry about the false positives and they don't have to contact a third party entity to check the revocation status of their peers' certificates. Further, with the shortened validity period of the SLCs, the certificates of malicious MNs would expire before their status could be verified by conventional schemes and before a major attack could be properly conducted. This reduces the risk of accepting the certificate of a malicious node before it expires. Indeed, MNs would not have to worry about

using conventional schemes to verify the certificate revocation status of their peers, but rather rely on the expiration date indicated on the certificate itself. This enable real-time communications by reducing the latency at the time of communication establishment since MNs do not have to check the revocation status of their peers' certificates.

In several works as [143], the certificate validity period V_p of SLCs is considered the same for all devices of the network, which can create a large workload if all devices synchronously request a new certificate at a given time. The process of updating SLCs with a large V_p value does not cause a high bandwidth consumption. However, a large value of V_p increases the risk of communicating with malicious nodes that are revoked but their certificate is still valid. This will give the malicious MNs more time to make attacks in the network even after their revocation. A short V_p enhances the security of the network by reducing the risk of accepting the certificates of malicious MNs before they expire, but it causes network congestion and bandwidth consumption due to the intensive certificate renewal process. In conclusion, a large V_p reduces the bandwidth consumption but at the expense of the IIoT network security. The network security is better when the value of V_p is short, however, this increases the consumption of the network bandwidth. To have a trade-off between bandwidth consumption and security, we consider that the validity period V_p of each SLC is proportional to the behavior and trust level of its owner. Hence, CL assigns a larger V_p to the legitimate MNs and a shorter V_p to the less trusted ones as follows:

$$V_p = (1 - p(X_S = 1|h(t_j))).Vp_{th} \tag{7.1}$$

Where: Vp_{th} is the maximum validity period of a SLC. The authors of [139] suggest that this period is 4 days, which corresponds to the average caching time of an OCSP response.

Dynamic V_p reduces bandwidth consumption, since only the SLC of less trusted nodes will be updated frequently. To revoke a malicious MN, the CL simply stops renewing its SLC. As a result, this MN will be unable to participate in future network activities because its old certificate must have expired, or will expire soon. Even in the case of a compromised CL, there will be no need to revoke the stolen certificates, since they have a short validity period.

7.4.2 Workflow of our Certificate verification scheme

As depicted in the flow chart in Fig. 7.2, when a MN wants to start communication with any other node "D" (MN or CL) in the network, first, it checks that the SLC of node D, received during the handshake phase is not expired. If the SLC is no longer valid, no communication will be established because D has already been revoked. In the case where D is the CL of the MN, the communication between the two nodes will be established directly without any verification process, because the MN knows in advance that its CL is a legitimate node. Otherwise, MN checks if the SLC of D is signed by the same CL as it, if it is the case, it means that the two MNs are in the same community. Thus, the communication between them can start immediately without checking the certificate revocation status. Each community is a trust area, i.e. the communications between nodes belonging to the same community are done directly without using certificate verification schemes as long as their V_p is not expired. This improves real-time security services and supports automation within IIoT environments. If the SLC of "D" is signed by another entity than the CL of the MN, which means that the two nodes "D" and MN do not belong to the same community, then the MN must check the certificate revocation status of D as described in flow chart in Fig. 7.3 before establishing communication with it.

When MN wants to establish a communication with a node belonging to another community or another domain whose SLC provider is unknown, the MN sends a request to the CL of its community to verify the legitimacy of the SLC provider. The CL checks if the ID of the SLC provider exists in the blockchain and corresponds to the identity of a CL that periodically publishes transactions in the blockchain. If the ID exists, the CL sends to the MN a response containing the provider's public key downloaded from the blockchain. MN uses the public key of the provider to verify the integrity of its peer's certificate. However, if the ID of the SLC provider is unknown and does not exist in the blockchain, the CL warns the MN to not establish a communication with the owner of this SLC.

7.5 Performance Evaluation

To evaluate the performance of our proposed certificate verification scheme, we use the simulation parameters listed in Table. 7.1.

Table 7.1: Simulation parameters of the certificate verification scheme

Simulation tool	contiki/cooja 2.7
Mote type	Tmote Sky
Nodes Distribution	Random
Certificate Size	1054 Bytes
Certificate's serial number size	20 bytes
CA's signature size	700 bytes
The median OCSP response time	291 ms
Size of OCSP	292 bytes
Average communication delay between IoT and CA	100 ms
Average Time of verification for each certificate	1ms
Size of Bloom filter	350 bytes
RSI structure	520 bytes
Chosen probability of a false positive	0.1

7.5.1 Performance evaluation of the SLC management process

In this part, we conduct a comparative study between the management of SLCs using our proposed hierarchical architecture described in section 7.3, the management of SLCs using Fog Computing architecture proposed in FONICA [143] and the management of SLCs using the Certificate Life-cycle Management (CLM) edge server proposed in [157]. In this comparison study, the percentage of energy saving and the central processing unit (CPU) usage saving over the entities responsible for SLCs management are evaluated for each architecture. If these architectures are not used, all SLCs will be managed by the traditional centralized architecture, where a single entity manages the SLCs for the entire network. Therefore, we normalize the saved energy and CPU usage against the centralized architecture consumption.

To save the energy consumption due to the management of SLCs in the centralized architecture, FONICA distribute the burden of SLCs management among a set of fog nodes, CLM distribute the burden among a set of CLM nodes and our proposed architecture distribute the burden among a set of CL nodes. In Fig. 7.4, we investigate the impact of the network density on the percentage of energy saving over the CL nodes during the SLCs management process. According to the results of Fig. 7.4, the percentage of energy saving over CLs in our architecture is much more significant than that saved over Fog nodes in FONICA and CLM nodes in [157], even when the density of the network increases. This is due to two main reasons: The first reason is that, unlike FONICA and CLM, the validity period of each SLC in our scheme is

proportional to the trust level of its owner. Thus, only the SLC of less trusted nodes will be renewed and issued frequently. This reduces the workload on the CLs and allows them to conserve their energy. The second reason is that the number of CLs in our architecture is proportional to the density of the IIoT network. As explained in chapter 5, when the network evolves, the number of CLs increases to absorb the burden related to the SLCs management, which explains the stability of the energy saving corresponding to our scheme.

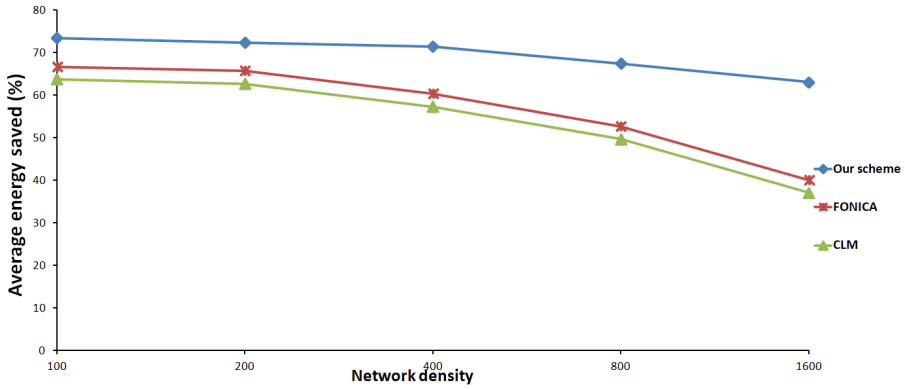


Figure 7.4: The impact of network density on the energy saving

Unlike the centralized architecture, CLs in our architecture, fog nodes in FONICA and CLM nodes in [157] do not receive requests for SLCs from all nodes in the network but just from the set of nodes they are supposed to manage. This allows them to save their CPU usage compared to the centralized architecture as shown in Fig. 7.5.

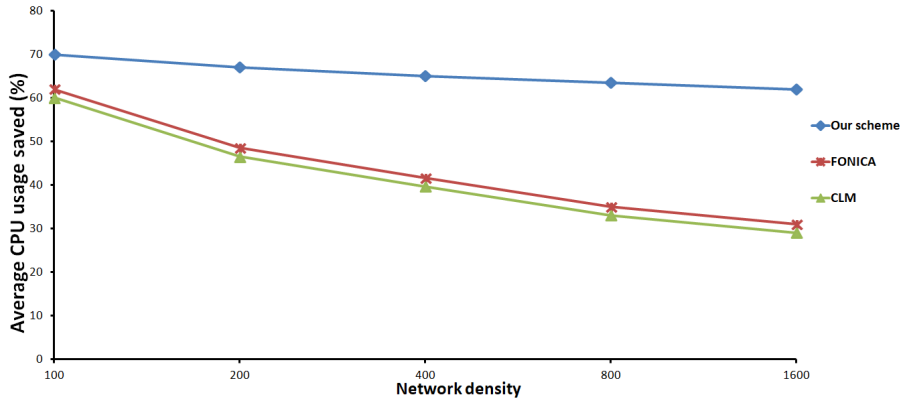


Figure 7.5: The impact of network density on the CPU usage saving

In FONICA and CLM, all nodes synchronously request a SLC at the same time which causes a substantial stress to the CPU of fog and CLM nodes. The increase of the number of nodes in the network increases the demand for SLCs and consequently decreases the percentage of CPU usage saving in

these schemes. In contrast, the percentage of CPU usage saving over CL nodes in our architecture is more important than that of FONICA and CLM. This is explained by the fact that in our scheme only the SLCs of less trusted nodes are renewed frequently and not all the SLCs in the network, which decreases the pressure on the CPU of the CLs. As explained above, in our hierarchical architecture, the number of CLs is proportional to the network density. This reduces the load on the existing CLs, which explains that even with the increase of the network density, the percentage of CPU usage saving remains above 60%.

In this part, we also evaluate the impact of the choice of SLC validity period on the risk of accepting a revoked certificate before it expires. For this purpose, in Fig. 7.6 we evaluate the risk of a MN accepting a revoked certificate before it expires, according to the increasing percentage of untrusted nodes in the network.

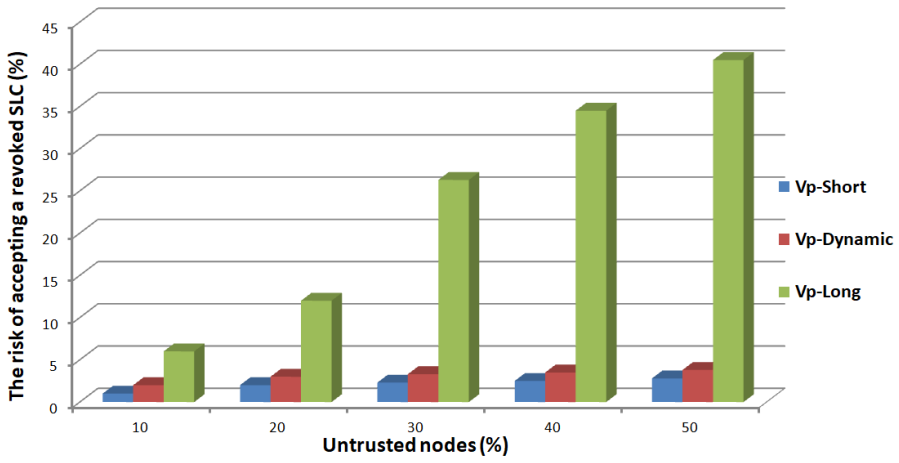


Figure 7.6: The choice of SLC validity period VS the risk of accepting a revoked certificate before it expires

We notice from Fig. 7.6 that the longer the validity period of SLCs, the higher the risk of communicating with malicious nodes whose certificate is revoked but not yet expired. This risk decreases when the validity period of SLCs is set to a shorter duration. However, SLCs with a short validity period cause network congestion and bandwidth consumption due to the intensive certificate renewal process. To this aim, in our proposed scheme the validity period of each SLC is dynamic, it is proportional to the trust level of its owner according to equation (7.1). From Fig. 7.6, the dynamic validity period gives good results even with a large percentage of malicious nodes in the network. These results are almost the same as the one obtained by $V_p - Short$, without having to frequently update the SLCs of all the nodes in the network. Only

the SLCs of less trusted nodes will be updated frequently.

7.5.2 Performance evaluation of the SLC-based certificate verification scheme

In this part, we study the time needed to obtain the revocation information as well as the resulting storage and communication overhead to achieve this aim in our proposed scheme. We also compare the obtained results with those of the related certificate verification schemes. Indeed, the time needed to obtain revocation information also called the verification time is responsible for the latency of communications between IIoT nodes. Furthermore, the resulting storage and communication overhead must be the most optimal to minimize the resource consumption of constrained devices.

- Verification Time :

Fig. 7.7 shows the results obtained with our verification scheme regarding the average time required for a MN to verify the revocation status of its peer in two cases: 1) the peer belongs to the same community as it, and 2) the peer belongs to another community than it. We notice that in the first case, the time needed for the MN to do the verification is 0 ms. Indeed, in this case, the MN relies only on the expiration date of the peer's certificate and no further verification is required. However, when the peer belongs to another community, the MN must check the reliability of the certificate issuer from its CL by following the process described in Fig. 7.3.

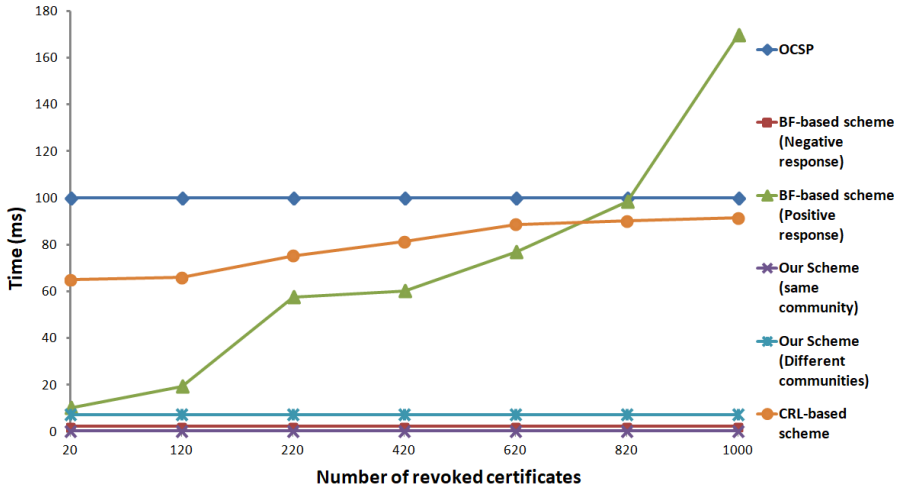


Figure 7.7: Time needed to verify the revocation status of a certificate

To allow IIoT nodes to have the majority of verification at 0 ms, we can gather in the same community the nodes that tend to communicate often with

each other. The Verification Time in our scheme does not increase with the variation of the number of revoked certificates.

Fig. 7.7 shows that the verification process in our scheme is faster than that of the OCSP and the CRL-based scheme proposed in [147]. The verification time in the CRL-based scheme [147] increases with the increase of the number of revoked certificates. Indeed, each node in [147] approaches the trusted edge nodes to verify whether a certificate is in their CRLs. If the certificate revocation status is not found among the CRLs of the edge nodes, the verification must be done through the CA hosted in the cloud. This process incurs an additional cost and increases the verification time. In Fig. 7.7, we also compare our scheme with the work [158] which proposes a verification scheme based on Bloom filter. According to the obtained results, we notice that the Bloom filter-based scheme [158] (BF-based scheme) gives good results when the filter provides a negative response. However, when the filter provides a positive response, additional verification for false positives must be performed.

- Storage and communication overhead

In Fig. 7.8, we compare the storage consumption of our proposed scheme with that of the OCSP, the CRL-based scheme [147] and BF-based scheme [158], when the number of revoked certificate ranges from 20 to 100.

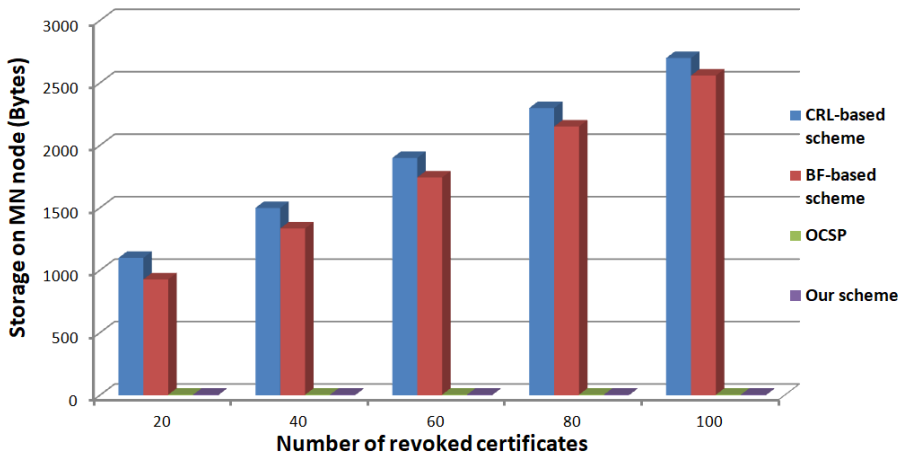


Figure 7.8: Storage consumption comparison under a varying number of revoked certificates

Fig. 7.8 shows that the storage consumption needed for a MN to verify the revocation state of a certificate in the CRL-based scheme and in the BF-based scheme is proportional to the number of revoked certificates. The CRL-based scheme is the scheme that consumes the most storage resources because it is based on CRLs. Hence, as the number of revoked certificates increases, the

MN must store more records in its memory to perform the verification. On another hand, OCSP and our scheme do not consume the storage resources of IIoT nodes, even when the number of revoked certificates increases. Indeed, in our schemes there is no need to maintain records at MNs. To perform verification, nodes simply refer to the information indicated in the peer's certificate.

In Fig. 7.9, we measure the communication overhead representing the data amount needed to exchange in order to ensure the verification task. As shown in Fig. 7.9, the Bf-based scheme requires the most communication overhead than the other schemes. This is due to the 12 % of cases where the filter provides a positive response. Indeed, positive responses involve additional verification using RSI structures whose the size is proportional to the number of revoked certificates. The communication overhead in the CRL-based scheme results from the communications with the trusted edge nodes or with the CA.

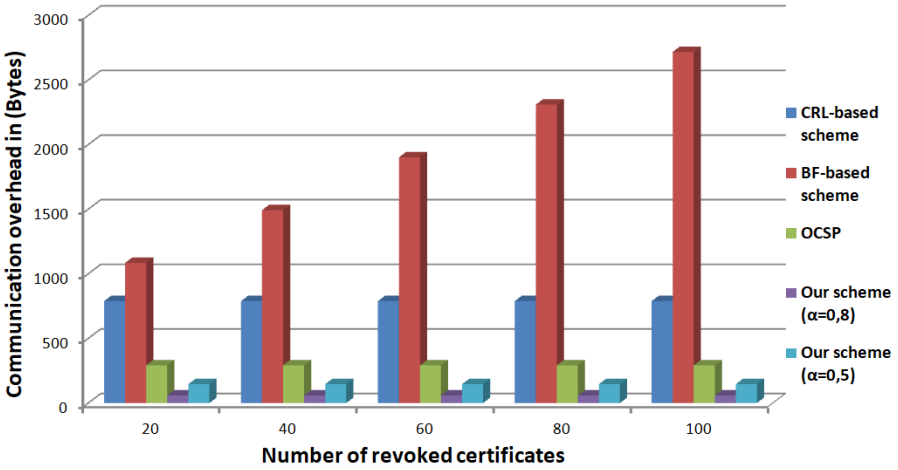


Figure 7.9: Communication overhead comparison under a varying number of revoked certificates

In our proposed scheme, the verification process performed inside the same community does not produce any communication overhead, since MNs do not need any third party to ensure the reliability of a certificate. Indeed, $\alpha = 0.8$ means that 80 % of the $MN's$ communications are performed inside the community, so the verification process for 80 % of communications is eliminated. All that remains is the verification process resulting from the 20% of communications made outside the community. The resulting communication overhead in our scheme is due solely to the verification process of the certificates belonging to different communities as described in Fig. 7.3. Hence, in order to decrease the communication overhead in our proposed scheme, we need to act on the community establishment method detailed in section 5.4.2.
In the hierarchical architecture, communities are formed based on the indus-

trial relationships between IIoT nodes. Indeed, one of these industrial relationships is the coworker relationship that gathers in the same community the IIoT nodes that tend to collaborate together. By privileging this relationship in the community establishment method, we will have in the same community the nodes that often collaborate together. This will increase the number of communications inside the community, decrease the number of communications outside the community and therefore decrease the communication overhead related to the certificate verification process. The communication overhead in our proposed scheme does not change with the variation of the number of revoked certificates, it is affected by the total number of certificates requiring verification by the CL. Indeed, the communication overhead produced by each node becomes larger as the number of verification increases.

To prove the efficiency of our proposed scheme, Fig. 7.10 compares the overall communication and storage overhead in the different verification schemes when the number of revoked certificates varies from 20 to 100 and the number of verification varies from 0 to 10. It can be clearly depicted from Fig. 7.10d that our proposed scheme is much efficient in terms of storage consumption and communication overhead in comparison with the OCSP (Fig. 7.10a), the CRL-based scheme (Fig. 7.10b) and the BF-based scheme (Fig. 7.10c), even when the number of revoked certificates and the number of verification increase. This is because our scheme does not require storage operations or communication overhead when performing verification within communities. The only overhead comes from the verification performed outside the communities. As explained before, thanks to the involvement of the industrial relationships between IIoT nodes in the community establishment method, even the resulting communication overhead of external verification can be further reduced. Therefore, considering the above discussion, our proposed certificate verification scheme is the most appropriate for IIoT networks.

7.6 Security Analysis

In this section, we provide a security analysis of our proposed certificate verification scheme against a set of potential security threats and vulnerabilities.

Compromised CA: an attacker can compromise the CA and access its private key to sign its own certificate as well as the certificates of other malicious nodes. Upon the detection of this attack in our proposed scheme, the compromised CL will be isolated from the network immediately and replaced by another CL.

In the other verification schemes, the compromised CA must find all stolen certificates to revoke them, which is not always obvious. In our proposed scheme, it is not needed to search for stolen certificates to revoke them since they have a short validity period and after expiration they will be useless.

Denial-of-Service (DoS) attack: is the most significant threat to the online verification schemes like OCSP, where an attacker or group of attackers sends multiple requests at the same time to overwhelm the resources of the OCSP server and make it unavailable. In our proposed scheme, MNs do not need the CL to do verification inside the community. However, our scheme is vulnerable to DOS attacks during external verification where the MN needs to contact its CL. To solve this problem, the MN can simply join another reachable CL to perform the verification. In our scheme, MN will always be able to change community in case its CL is unreachable, which will allow it to have a backup system to verify the certificate revocation status of its peers. In other verification schemes like CRL and OCSP there is no backup system; disabling the CA or the OCSP server could cause a major network disruption.

Man in The Middle (MiTM) attack: this attack can be deployed by intercepting the verification requests and responding with a fraudulent response, signed with random data before the legitimate response arrives. Indeed, this attack is popular for the OCSP protocol, where the attacker intercepts the communications between the nodes and the OCSP server. Our scheme excludes OCSP responses, which prevents MiTM attacks especially in the verification inside communities. However, the outside verification are still vulnerable to MiTM attacks. Problems related to MiTM can solved by Elliptic Curve Digital Signature Algorithm (ECDSA) validations. For OCSP, nodes have to do ECDSA validations for all their certificate status verification, which will lead to a consumption of their resources. ECDSA validations in our scheme are not needed for all verification as OCSP, but just for verification outside the community. Thanks to the involvement of the industrial relationships in the community establishment method, the majority of communications occur between nodes belonging to the same community, which does not require verification and minimizes the risks of MiTM.

Trust related attacks: some certificate verification schemes are vulnerable to trust related attacks; for example the work in [146] proposes to share the CRL file between all the nodes of the network instead of each node containing the whole file. Hence, before a node starts communication with its peer, first,

it sends a request to the node having the status of the required certificate. However, a malicious node can confirm that a requested certificate is not revoked, which is completely false. This allows the malicious node to perform more attacks in the network. Our certificate verification scheme is not vulnerable to Trust related attacks because only the legitimate CLs can decide if a certificate is valid or not.

Whitewashing attacks: Occur when a node changes its community each time the CL detects its untrusted behavior. To deal with this behavior, the CL in our proposed scheme stops renewing the SLC of the detected malicious node to revoke it from the network. The CL also registers this revocation in the blockchain. Hence, when the malicious node aims to join another community, the new CL asks it to send the ID of its old CL to search in the blockchain for the URI pointing to the file containing the revocation information of the old community. If the certificate of the node is not revoked, the CL uses the trust evaluation of the old CL to define the validity period of the node's certificate according to equation (7.1).

Replay Attacks: an attacker can steal the private key of an IIoT node and at the same time request the certificate status of that node from the Validation Authority (VA). Therefore, the attacker will have all the means to impersonate the IIoT node even if its certificate is revoked. Each time a node wants to verify the revocation status of the victim node's certificate, the attacker replays the previously obtained VA response. OCSP and the proposed work in [149] are vulnerable to replay attacks, while our proposed verification scheme can cope with this kind of attacks by using SLCs. Indeed, once the validity period of a SLC expires, the CL does not renew it if its owner is untrusted or its private key is declared stolen. Hence, the attacker will not be able to use the stolen certificate after its expiration. It will also not be able to replay the VA response because the other nodes in the network will consider the SLC invalid since it is not renewed.

7.7 Conclusion

In this chapter, we based on trust management and SLCs to propose a certificate verification scheme suitable for IIoT network requirements. Our experiments showed the efficiency of our proposed certificate verification scheme compared to conventional ones, including centralized SLCs, CRL, Bloom filter and OCSP. We also compared our scheme with works that have proposed

improvements to conventional verification schemes to prove that our scheme outperforms them.

In the last section of this chapter, we provide a security analysis of our proposed certificate verification scheme against a set of potential security threats and vulnerabilities. In our future work, we will propose simulations to validate the resilience of our certificate verification scheme to attacks and threats discussed in the Section 7.6 of this chapter.

In this chapter, we studied the time required to obtain the revocation information as well as the resulting storage and communication overhead to achieve this aim. The purpose was to prove the efficiency of our proposed certificate verification scheme. However, we have not evaluated the cost to renew an expired certificate of a legitimate MN. This is foreseen for the future works.

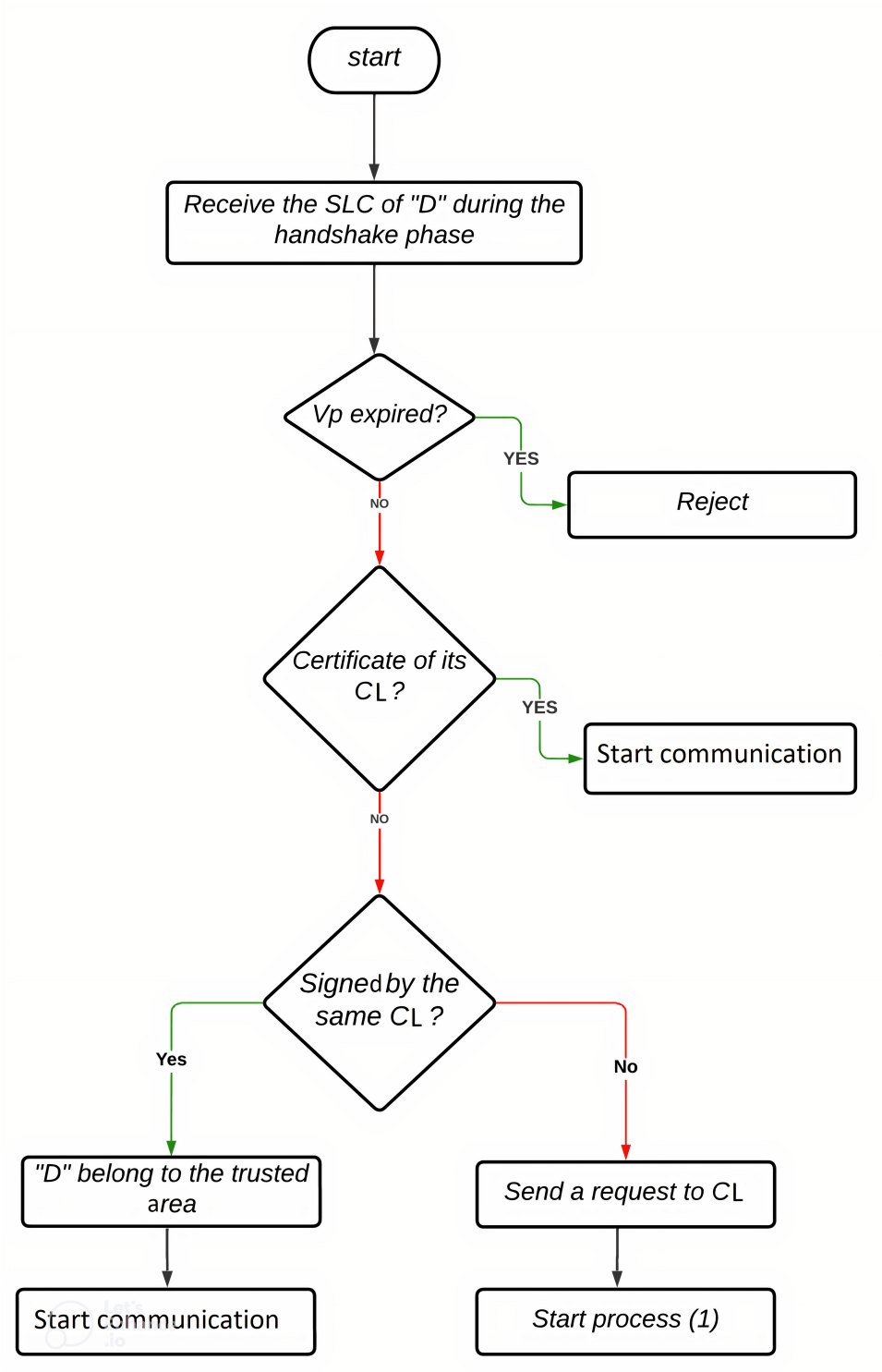


Figure 7.2: Workflow of the proposed certificate verification scheme

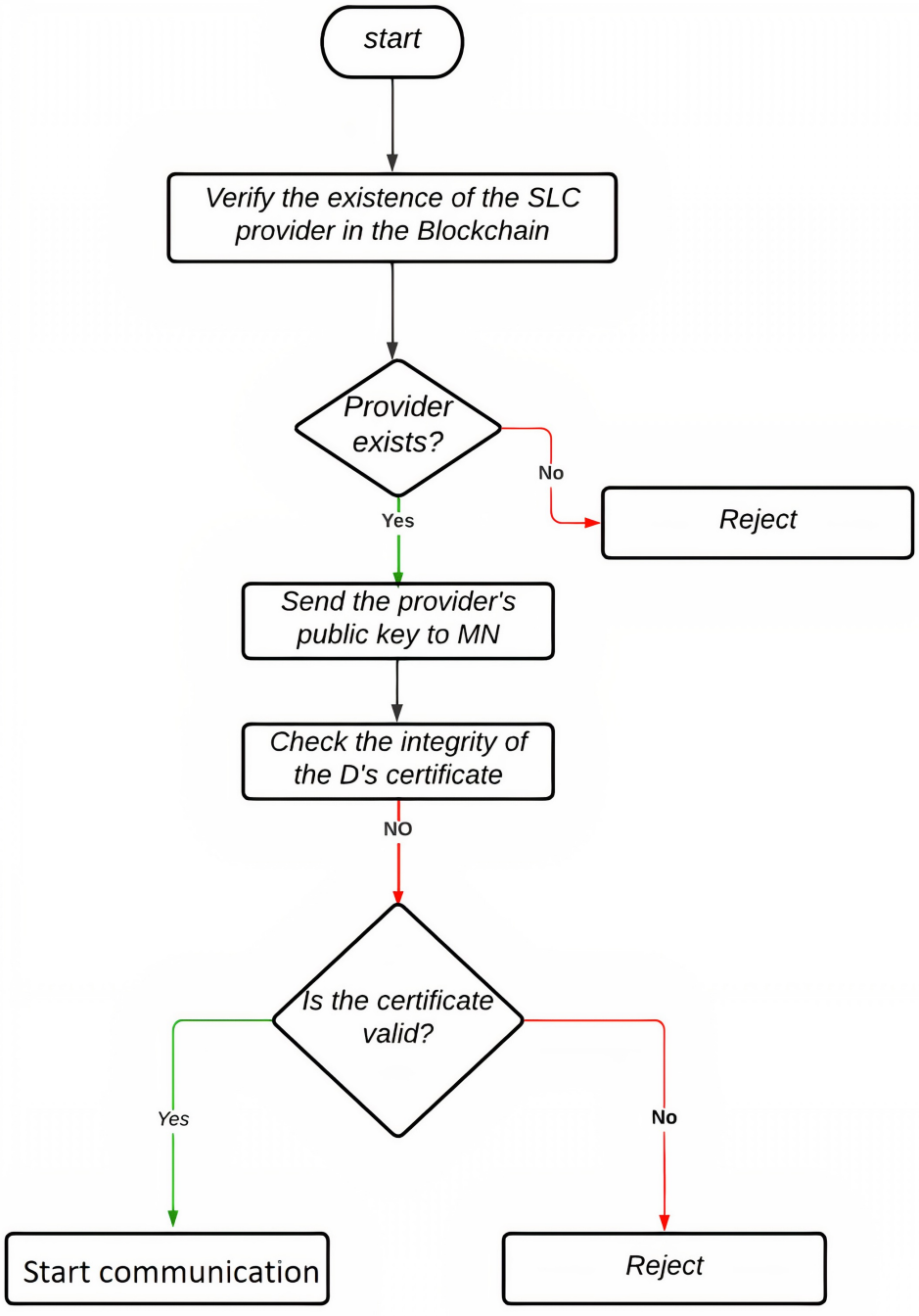
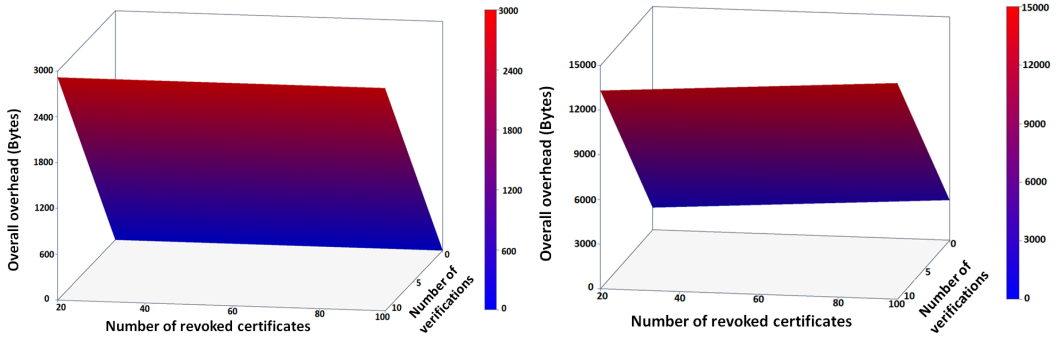
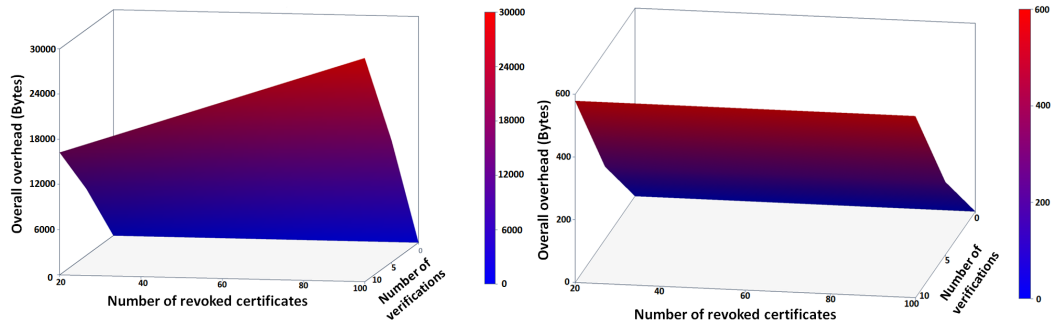


Figure 7.3: Workflow of the process(1)



(a) The overall overhead of OCSP

(b) The overall overhead of the CRL-based scheme



(c) The overall overhead of the BF-based scheme

(d) The overall overhead of our scheme

Figure 7.10: The overall communication and storage overhead according to the number of revoked certificates and the number of verification

Chapter 8

Conclusion and perspectives

8.1 Conclusion

In this thesis we were interested in trust management in IoT and IIoT networks in order to strengthen the security and improve the performance of these networks.

As trust is multidisciplinary concept, in a first step, we gave a detailed introduction of trust in wireless networks based on its definition in several disciplines including sociology, economics, philosophy, psychology, and autonomic computing and human-machine interactions. We also presented the different modules, components and related attacks that must be considered when designing an effective trust management model.

The effectiveness of the trust management in terms of security and confidentiality can be achieved by employing the appropriate trust modeling methods. However, there is no single and standard methodology for modeling and evaluating trust. In the literature, each work models and evaluates trust according to its own definition and perception of trust, also according to the needs of its application. Therefore, to give the reader a clearer vision on trust modeling in wireless networks, we provided an overview of the literature on the most used methods for modeling trust in these networks.

Afterwards, we proposed a distributed analytical trust management model for IoT networks where each node monitors the behavior of its neighbors and assigns them a local trust metric. Depending on the result of the monitoring process, the trust metric of the monitored node may increase, decrease, remain unchanged or goes down to zero. Indeed, we modeled the state changes of the trust metric by using a discrete-time Markov chain with $M + 1$ states. Each state corresponds to a trust level. The number of transitions to achieve the

high level of trust as well as the trust interval in the proposed model can be adapted according to the monitoring process and network characteristics. This makes our trust management model flexible and adaptable

Subsequently, we oriented our research towards the trust management in IIoT networks where the security challenges are even greater. Indeed, IIoT devices manage sensitive data related to company trade secrets, which makes them a very attractive target for security threats that aim at compromising the reliability of the IIoT network. For this purpose, we proposed a trust management model for IIoT networks by using a new concept called industrial relationship between devices. To the best of our knowledge, at the time we started our research on trust management in IIoT, there was no proposed trust management model specifically for IIoT networks that considers the requirements and constraints of these networks.

In the proposed hierarchical architecture, to remove uncertainty about the type of a member node (legitimate or malicious), we modeled the interaction between the member node and its detection agent as a signaling game. Indeed, the detection agent uses the behavior of nodes in the network as a signal about their type. As the game between the agent and the member IIoT node evolves, the agent can obtain its best response strategy based on its belief on the member node. This allows the agent to make rational, accurate and fast decisions which increases the reliability of our proposed certificate revocation process.

When a node is revoked, all other nodes in the IIoT network must be notified to avoid any connection with it. This process must be done quickly, in a resource-efficient manner, and without latency. For this purpose, we proposed a new efficient SLC-based certificate verification scheme suitable for IIoT network requirements. The proposed solution addresses challenges due to the use of SLCs and supports automation in IIoT environments. We also introduced the notion of trust to determine the validity period of SLCs. Indeed, the validity period of each SLC is proportional to the trust level of its owner so that only the SLCs of less trusted nodes will be updated frequently. To the best of our knowledge, our work is the first to integrate the notion of trust in the calculation of the validity period of SLCs. This makes a good trade-off between certificate life and overheads resulting from certificate renewal process, while keeping a high security level.

8.2 Perspectives

The work done in this thesis opens many perspectives. About short-term perspectives, we intend to deploy our trust management model for IIoT network, our certificate revocation mechanism and our certificate verification scheme in a real IIoT environment. The objective is to perform a testbed and compare the results with those obtained by the contiki/cooja simulator.

In section 7.6 of chapter 7, we provided a security analysis of our proposed certificate verification scheme against a set of potential security threats and vulnerabilities such as DOS attack, MiTM attack, whitewashing attacks, etc. In our future work, we intend to deepen this section. Indeed, we will consider simulations challenging the proposed certificate verification scheme with attack scenarios to prove its robustness.

In our future work, we will further improve our proposed analytical trust management model for IoT networks. We will also propose network simulations to prove that our trust management model addresses the main requirements and constraints of IoT networks, such as scalability, resource-constrained devices, energy consumption, etc.

This thesis also opens many long term perspectives. One of the main aims of six generation (6G) is highly intelligent, fully autonomous, and ultra-dense heterogeneous Internet of Everything (IoE) system. The decentralized and dynamic nature of devices in IoE systems makes the implementation of a uniform security system very difficult, and raises security as one of the main concerns of the 6G network. Indeed, intelligent IoE devices require intelligent 6G security.

Trust management can be a potential method to ensure reliable, real-time communications in 6G wireless networks. However, how to improve the 6G security intelligence by integrating Artificial intelligence (AI) into the trust management system and ensure reliable end-to-end communication in the IoE system remains a challenging task that requires further investigation.

Due to the dynamic characteristics of 6G wireless networks, it is difficult to enrich the training dataset, which affects the effectiveness of traditional AI techniques. Finding a promising deep learning method is a very important issue that is still open and needs more investigation, because the lack of training data reduces the accuracy of trust assessment and decrease the reliability of 6G networks.

The development of 6G and artificial intelligence has accelerated the proliferation of robotic technology in the popular sphere. Currently, robots are not

only deployed in the industrial area, but they are increasingly present in public spaces, where they interact directly with humans. The main question is: are humans ready to accept these service robots at their side? This acceptance depends on the level of trust that humans place in robots. Trust in human-robot interactions is a very important issue that needs further research, as there is a growing need to understand and measure human trust in robots. Therefore, determining the factors that influence this trust is not an obvious task.

Which dimension to choose to measure the trust of humans in robots? Is it the QoS dimension that defines the capabilities of the robot, or the social dimension used in social networks and that defines the willingness of the user to be vulnerable?

In our thesis, the interactions between IIoT devices led us to define a new industrial dimension of trust. Will the understanding of human-robot interactions also lead researchers to the definition of a new dimension of trust?

List of Figures

1.1 Applications of IoT 2

2.1 The concept of trust defined by the other disciplines can be applied to modeling trust in communications and networking [7] 10
2.2 Trust management modules. 14
2.3 Percentage of efforts made in different trust management applications 21

4.1 State transition diagram 54
4.2 The honesty related to the direct interactions 55
4.3 The honesty related to the indirect interactions 57
4.4 $p_d = 0.80$, $p_I = 0.95$ 61
4.5 $p_d = 0.95$, $p_I = 0.95$ 62
4.6 $p_d = 0.8$, $p_I = 0.6$ 63
4.7 $p_d = 0.95$, $p_I = 0.95$ 64

5.1 Proposed H-IIoT architecture 73
5.2 The topology of the traditional centralized IIoT architecture . . 85
5.3 The topology of the proposed H-IIoT architecture 86
5.4 The average power consumption in the traditional IIoT architecture 87
5.5 The average power consumption in the proposed H-IIoT architecture 87
5.6 Average energy consumption of nodes with the evolution of the network population and the increase of κ 88
5.7 Average energy consumption of trust management entities, with the evolution of the network population and the increase of κ . . 89
5.8 Average number of lost monitoring packets, with the evolution of the network population and the increase of κ 90
5.9 Tm of a Good node with Pm ranging from 20% to 50% 91
5.10 Tm of a good node, Pm=20% 92
5.11 Tm of a good node, Pm=50% 92
5.12 Behavior changes of a non-critical node, Pm=30% 93
5.13 Behavior changes of a critical node, Pm=30% 93
5.14 Behavior changes of a critical node, $\kappa = 0.8$, $\kappa = 0.5$, $\kappa = 0.1$ . . 94
5.15 Coalition attack: Bad-mouthing attack 95
5.16 Coalition attacks: Ballot-stuffing attack 96

6.1 Extensive form of the stage Certificate Revocation Game 107
6.2 The certificate revocation mechanism 116

6.3 The convergence speed of the $p(X_S = 1|A_S(t_j), h(t_j))$ according to φ and μ . . . 118
6.4 Change trend of p^*_{uj} and λ^*_j according to φ . . . 119
6.5 Revocation rate vs malicious nodes . . . 120
6.6 Revocation time vs malicious nodes . . . 121

7.1 Transaction written by the CL in the blockchain after each monitoring or revocation. . . . 129
7.4 The impact of network density on the energy saving . . . 134
7.5 The impact of network density on the CPU usage saving . . . 134
7.6 The choice of SLC validity period VS the risk of accepting a revoked certificate before it expires . . . 135
7.7 Time needed to verify the revocation status of a certificate . . . 136
7.8 Storage consumption comparison under a varying number of revoked certificates . . . 137
7.9 Communication overhead comparison under a varying number of revoked certificates . . . 138
7.2 Workflow of the proposed certificate verification scheme . . . 143
7.3 Workflow of the process(1) . . . 144
7.10 The overall communication and storage overhead according to the number of revoked certificates and the number of verification . . . 145

List of Tables

3.1 Comparison between the Trust Modelling methods 43

5.1 Simulation parameters 84

6.1 Non-voting strategy vs voting strategy 103
6.2 Notations used by the Certificate Revocation Game 105
6.3 Utility functions of the Certificate Revocation Game 106

7.1 Simulation parameters of the certificate verification scheme . . . 133

Bibliography

Personal publications

[1] Chaimae Boudagdigue, Abderrahim Benslimane, Abdellatif Kobbane, and Mouna Elmachkour. "A Distributed Advanced Analytical Trust Model for IoT". In: *IEEE International Conference on Communications (ICC)*. 2018, pp. 1–6. DOI: 10.1109/ICC.2018.8422726.

[2] Chaimaa Boudagdigue, Abderrahim Benslimane, Abdellatif Kobbane, and Jiajia Liu. "Trust Management in Industrial Internet of Things". In: *IEEE Transactions on Information Forensics and Security,* vol. 15, (2020), pp. 3667–3682. DOI: 10.1109/TIFS.2020.2997179.

[3] Chaimaa Boudagdigue, Abderrahim Benslimane, and Abdellatif Kobbane. "Cluster-based certificate revocation in industrial loT networks using Signaling game". In: *IEEE Global Communications Conference (GLOBECOM)*. 2020, pp. 1–6. DOI: 10.1109/GLOBECOM42002.2020.9322497.

[4] Chaimaa Boudagdigue, Abderrahim Benslimane, Abdellatif Kobbane, and Jiajia Liu. "Trust-based Certificate Management for industrial IoT networks". In: *Accepted for publication in IEEE Internet of Things Journal,* (2023).

References

[5] Wade Trappe, Richard Howard, and Robert S. Moore. "Low-Energy Security: Limits and Opportunities in the Internet of Things". In: *IEEE Security Privacy,* vol. 13, no. 1, (2015), pp. 14–21. DOI: 10.1109/MSP.2015.7.

[6] Anthea Mayzaud, Remi Badonnel, and Isabelle Chrisment. "A Taxonomy of Attacks in RPL-based Internet of Things". In: *International Journal of Network Security,* vol. 18, no. 3, (2016), pp. 459–473. DOI: 10.6633/IJNS.201605.18(3).07.

[7] Jin-Hee Cho, Ananthram Swami, and Ing-Ray Chen. "A Survey on Trust Management for Mobile Ad Hoc Networks". In: *IEEE Communications Surveys Tutorials,* vol. 13, no. 4, (2011), pp. 562–583. DOI: 10.1109/SURV.2011.092110.00088.

[8] Hassan Jameel, Le Hung, Umar Kalim, Ali Sajjad, and Sungyoung Lee. "A trust model for ubiquitous systems based on vectors of trust values". In: vol. 2005. Jan. 2006, pp. 6. ISBN: 0-7695-2489-3. DOI: 10.1109/ISM.2005.22.

[9] Russell Hardin. *Trust and Trustworthiness*. New York, NY: Russell Sage Foundation, 2002, 256 pages.

[10] Jason D'Cruz. "Humble Trust". In: *Philosophical Studies,* vol. 176, no.4, (2019), pp. 933–953. DOI: 10.1007/s11098-018-1220-6.

[11] Soroush Aalibagi, Hamidreza Mahyar, Ali Movaghar, and H. Eugene Stanley. "A Matrix Factorization Model for Hellinger-Based Trust Management in Social Internet of Things". In: *IEEE Transactions on Dependable and Secure Computing* vol. 19.4 (2022), pp. 2274–2285. DOI: 10.1109/TDSC.2021.3052953.

[12] Jia Guo, Ing-Ray Chen, and Jeffrey Tsai. "A Survey of Trust Computation Models for Service Management in Internet of Things Systems". In: *Computer Communications,* vol. 97, (Oct. 2016), pp. 1–14. DOI: 10.1016/j.comcom.2016.10.012.

[13] Govindaraj Ramya, Govindaraj Priya, Chowdhury Subrata, Dohyeun Kim, Duc Tan Tran, and Anh Ngoc Le. "A Review on Various Applications of Reputation Based Trust Management A Review on Various Applications of Reputation Based Trust Management". In: *International Journal of Interactive Mobile Technologies (iJIM),* vol. 15, no. 10, (May 2021), pp. 87–102. DOI: 10.3991/ijim.v15i10.21645.

[14] Zhaojun Lu, Gang Qu, and Zhenglin Liu. "A Survey on Recent Advances in Vehicular Network Security, Trust, and Privacy". In: *IEEE Transactions on Intelligent Transportation Systems,* vol. 20, no. 2, (2019), pp. 760–776. DOI: 10.1109/TITS.2018.2818888.

[15] Avani Sharma, Emmanuel Pilli, Arka Mazumdar, and Poonam Gera. "Towards trustworthy Internet of Things: A survey on Trust Management applications and schemes". In: *Computer Communications,* vol. 160, (June 2020), pp. 475–493. DOI: 10.1016/j.comcom.2020.06.030.

[16] Zeinab Movahedi, Zahra Hosseini, Fahimeh Bayan, and Guy Pujolle. "Trust-Distortion Resistant Trust Management Frameworks on Mobile Ad Hoc Networks: A Survey". In: *IEEE Communications Surveys Tutorials,* vol. 18, no. 2, (Oct. 2015), pp. 1287–1309. DOI: 10.1109/COMST.2015.2496147.

[17] Sarah Ali Siddiqui, Adnan Mahmood, Quan Sheng, Hajime Suzuki, and Wei Ni. "Trust in Vehicles Towards Context-aware Trust and Attack Resistance for the Internet of Vehicles". In: *TechRxiv,* (May 2022). DOI: 10.36227/techrxiv.19657665.v1.

[18] Michele Nitti, Roberto Girau, and Luigi Atzori. "Trustworthiness Management in the Social Internet of Things". In: *IEEE Transactions on Knowledge and Data Engineering,* vol. 26, no. 5, (2014), pp. 1253–1266. DOI: 10.1109/TKDE.2013.105.

[19] Michele Nitti, Roberto Girau, Luigi Atzori, Antonio Iera, and Giacomo Morabito. "A subjective model for trustworthiness evaluation in the social Internet of Things". In: *IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. 2012, pp. 18–23. DOI: 10.1109/PIMRC.2012.6362662.

[20] Ikram Ud Din, Mohsen Guizani, Byung-Seo Kim, Suhaidi Hassan, and Khurram Khan. "Trust Management Techniques for the Internet of Things: A Survey". In: *IEEE Access* vol. 7, (Nov. 2018), pp. 29763–29787. DOI: 10.1109/ACCESS.2018.2880838.

[21] Shenyun Che, Renjian Feng, Xuan Liang, and Xiao Wang. "A lightweight trust management based on Bayesian and Entropy for wireless sensor networks". In: *Security and Communication Networks,* vol. 8, (Jan. 2015), pp. 168–175. DOI: 10.1002/sec.969.

[22] Aljawharah Alnasser and Hongjian Sun. "Global Roaming Trust-based Model for V2X Communications". In: *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2019, pp. 1–6. DOI: 10.1109/INFOCOMWKSHPS47286.2019.9093753.

[23] Xiaoxiong Zhong, Renhao Lu, Li Li, Xinghan Wang, and Yanbin Zheng. "DSOR: A Traffic-Differentiated Secure opportunistic Routing with Game Theoretic Approach in MANETs". In: *IEEE Symposium on Computers and Communications (ISCC)*. 2019, pp. 1–6. DOI: 10.1109/ISCC47284.2019.8969650.

[24] Nabil Djedjig, Djamel Tandjaoui, Faiza Medjek, and Imed Romdhani. "New trust metric for the RPL routing protocol". In: *8th International Conference on Information and Communication Systems (ICICS)*. 2017, pp. 328–335. DOI: 10.1109/IACS.2017.7921993.

[25] Fatemehsadat Mirsadeghi, Marjan Kuchaki Rafsanjani, and Brij B Gupta. "A trust infrastructure based authentication method for clustered vehicular ad hoc networks". In: *Peer-to-Peer Networking and Applications,* vol. 14, no. 4, (July 2021), pp. 2537–2553. DOI: 10.1007/s12083-020-01010-4.

[26] Dajun Zhang, F.Richard Yu, Ruizhe Yang, and Helen Tang. "A deep reinforcement learning-based trust management scheme for software-defined vehicular networks". In: *8th ACM symposium on design and analysis of intelligent vehicular networks and applications*. 2018, pp. 1–7.

[27] Luigi Atzori, Antonio Iera, Giacomo Morabito, and Michele Nitti. "The Social Internet of Things (SIoT) – When social networks meet the Internet of Things: Concept, architecture and network characterization". In: *Computer Networks,* vol. 56, (Nov. 2012), pp. 3594–3608. DOI: 10.1016/j.comnet.2012.07.010.

[28] Aparna A. Junnarkar, Y. P. Singh, and Vivek S. Deshpande. "SQMAA: Security, QoS and Mobility Aware ACO Based Opportunistic Routing Protocol for MANET". In: *4th International Conference for Convergence in Technology (I2CT)*. 2018, pp. 1–6. DOI: 10.1109/I2CT42659.2018.9058022.

[29] Mohammad Nikravan, Ali Movaghar, and Mehdi Hosseinzadeh. "A lightweight signcryption scheme for defense against fragment duplication attack in the 6LoWPAN networks". In: *Peer-to-Peer Networking and Applications,* vol. 12, (Jan. 2019), pp. 1–18. DOI: 10.1007/s12083-018-0659-8.

[30] M Hema Kumar, V Mohanraj, Y Suresh, J Senthilkumar, and G Nagalalli. "Trust aware localized routing and class based dynamic block chain encryption scheme for improved security in WSN". In: *Journal of Ambient Intelligence and Humanized Computing,* vol. 12, no. 5, (2021), pp. 5287–5295. DOI: 10.1007/s12652-020-02007-w.

[31] David Airehrour and Sayan Ray. "SecTrust-RPL: A secure trust-aware RPL routing protocol for Internet of Things". In: *Future Generation Computer Systems,* vol. 93, (Mar. 2018), pp. 1248–1278. DOI: 10.1016/j.future.2018.03.021.

[32] Charles E Perkins, Elizabeth M Royer, Samir R Das, and Mahesh K Marina. "Performance comparison of two on-demand routing protocols for ad hoc networks". In: *IEEE Personal communications,* vol.8, no. 1, (2001), pp. 16–28. DOI: 10.1109/98.904895.

[33] Muhammad Khalid Riaz, Fan Yangyu, and Imran Akhtar. "A Multidimensional Trust Inference Model for the Mobile Ad-Hoc Networks". In: *28th Wireless and Optical Communications Conference (WOCC).* 2019, pp. 1–5. DOI: 10.1109/WOCC.2019.8770587.

[34] Ruo Jun Cai, Xue Jun Li, and Peter Han Joo Chong. "An Evolutionary Self-Cooperative Trust Scheme Against Routing Disruptions in MANETs". In: *IEEE Transactions on Mobile Computing,* vol. 18, no. 1, (2019), pp. 42–55. DOI: 10.1109/TMC.2018.2828814.

[35] Neeraj Arya, Upendra Singh, and Sushma Singh. "Detecting and avoiding of worm hole attack and collaborative blackhole attack on MANET using trusted AODV routing algorithm". In: *International Conference on Computer, Communication and Control (IC4).* 2015, pp. 1–5. DOI: 10.1109/IC4.2015.7375649.

[36] D. Gayathri and S. Janaki Raman. "Pltrust AODV: Physical logical factor estimated trust embedded AODV for optimised routing in Manets". In: *4th International Conference on Advanced Computing and Communication Systems (ICACCS).* 2017, pp. 1–5. DOI: 10.1109/ICACCS.2017.8014647.

[37] Seyyed Yasser Hashemi and Fereidoon Shams Aliee. "Fuzzy, Dynamic and Trust Based Routing Protocol for IoT". In: *Journal of Network and Systems Management,* vol. 28, no. 4, (Oct. 2020), pp. 1248–1278. DOI: 10.1007/s10922-020-09535-y.

[38] Ing-Ray Chen, Jia Guo, and Fenye Bao. "Trust management for service composition in SOA-based IoT systems". In: *IEEE Wireless Communications and Networking Conference, WCNC.* Nov. 2014, pp. 3444–3449. DOI: 10.1109/WCNC.2014.6953138.

[39] Hannan Xiao, Nitin Sidhu, and Bruce Christianson. "Guarantor and reputation based trust model for Social Internet of Things". In: *International Wireless Communications and Mobile Computing Conference (IWCMC)*. 2015, pp. 600–605. DOI: 10.1109/IWCMC.2015.7289151.

[40] Fenye Bao and Ing-Ray Chen. "Trust management for the internet of things and its application to service composition". In: June 2012, pp. 1–6. ISBN: 978-1-4673-1238-7. DOI: 10.1109/WoWMoM.2012.6263792.

[41] Upul Jayasinghe, Gyu Myoung Lee, Tai-Won Um, and Qi Shi. "Machine Learning Based Trust Computational Model for IoT Services". In: *IEEE Transactions on Sustainable Computing,* vol. 4, no. 1, (2019), pp. 39–52. DOI: 10.1109/TSUSC.2018.2839623.

[42] Narani Aravinthan and K. Geetha. "Certificate Revocation Scheme Based on Weighted Voting Game and Rational Secure Multiparty Computing". In: *ICTACT Journal on Communication Technology,* vol. 08, (Mar. 2017), pp. 1453–1460. DOI: 10.21917/ijct.2017.0215.

[43] Dong Chen, Guiran Chang, Dawei Sun, Jiajia Li, Jie Jia, and Xingwei Wang. "TRM-IoT: A Trust Management Model Based on Fuzzy Reputation for Internet of Things". In: *Computer Science and Information Systems,* vol. 8, no. 4, (Oct. 2011), pp. 1207–1228. DOI: 10.2298/CSIS110303056C.

[44] Giancarlo Fortino, Fabrizio Messina, Domenico Rosaci, Giuseppe M. L. Sarné, and Claudio Savaglio. "A Trust-Based Team Formation Framework for Mobile Intelligence in Smart Factories". In: *IEEE Transactions on Industrial Informatics,* vol. 16, no. 9 (2020), pp. 6133–6142. DOI: 10.1109/TII.2020.2963910.

[45] Sarah Ali Siddiqui, Adnan Mahmood, Quan Sheng, Hajime Suzuki, and Wei Ni. "A Survey of Trust Management in the Internet of Vehicles". In: *Electronics,* vol. 10, no. 18, (Sept. 2021), pp. 2079–9292. DOI: 10.3390/electronics10182223.

[46] Furong Wang, Chen Huang, Jing Zhao, and Chunming Rong. "IDMTM: A Novel Intrusion Detection Mechanism Based on Trust Model for Ad Hoc Networks". In: *22nd International Conference on Advanced Information Networking and Applications (AINA)*. 2008, pp. 978–984. DOI: 10.1109/AINA.2008.124.

[47] Faiza Medjek, Djamel Tandjaoui, Imed Romdhani, and Nabil Djedjig. "A Trust-Based Intrusion Detection System for Mobile RPL Based Networks". In: *IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. 2017, pp. 735–742. DOI: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.113.

[48] Guntur Dharma Putra, Volkan Dedeoglu, Salil S. Kanhere, and Raja Jurdak. "Trust Management in Decentralized IoT Access Control System". In: *IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. 2020, pp. 1–9. DOI: 10.1109/ICBC48266.2020.9169481.

[49] Parikshit Mahalle, Pravin Thakre, Neeli Prasad, and Ramjee Prasad. "A fuzzy approach to trust based access control in internet of things". In: *Wireless VITAE 2013*. 2013, pp. 1–5. DOI: 10.1109/VITAE.2013.6617083.

[50] Hamed Hellaoui, Abdelmadjid Bouabdallah, and Mouloud Koudil. "TAS-IoT: Trust-Based Adaptive Security in the IoT". In: *2016 IEEE 41st Conference on Local Computer Networks (LCN)*. 2016, pp. 599–602. DOI: 10.1109/LCN.2016.101.

[51] Sofiane Dahmane, Chaker Abdelaziz Kerrache, Nasreddine Lagraa, and Pascal Lorenz. "WeiSTARS: A weighted trust-aware relay selection scheme for VANET". In: *2017 IEEE International Conference on Communications (ICC)*. 2017, pp. 1–6. DOI: 10.1109/ICC.2017.7996451.

[52] Upul Jayasinghe, Abayomi Otebolaku, Tai-Won Um, and Gyu Myoung Lee. "Data centric trust evaluation and prediction framework for IOT". In: *ITU Kaleidoscope: Challenges for a Data-Driven Society (ITUK)*. 2017, pp. 1–7. DOI: 10.23919/ITU-WT.2017.8246999.

[53] Weili Han, Yun Gu, Yin Zhang, and Lirong Zheng. "Data driven quantitative trust model for the Internet of Agricultural Things". In: *International Conference on the Internet of Things (IOT)*. 2014, pp. 31–36. DOI: 10.1109/IOT.2014.7030111.

[54] Amira Kchaou, Ryma Abassi, and Sihem Guemara. "Towards the performance evaluation of a clustering and trust based security mechanism for VANET". In: *15th International Conference on Availability, Reliability and Security*. 2020, pp. 1–6. DOI: 10.1145/3407023.3407071.

[55] Sarah Oubabas, Rachida Aoudjit, Joel Rodrigues, and Said Talbi. "Secure and Stable Vehicular Ad Hoc Network Clustering Algorithm based on Hybrid Mobility Similarities and Trust Management Scheme". In: *Vehicular Communications,* vol. 13, (Aug. 2018), pp. 128–138. DOI: 10.1016/j.vehcom.2018.08.001.

[56] Jingpei Wang, Zhenyong Zhang, and Mufeng Wang. "A Trust Management Method Against Abnormal Behavior of Industrial Control Networks Under Active Defense Architecture". In: *IEEE Transactions on Network and Service Management,* vol. 19, no. 3, (2022), pp. 2549–2572. DOI: 10.1109/TNSM.2022.3173398.

[57] Xin Kang and Yongdong Wu. "A Trust-based Pollution Attack Prevention Scheme in Peer-to-Peer Streaming Networks". In: *Computer Networks,* vol. 72, (July 2014), pp. 62–73. DOI: 10.1016/j.comnet.2014.07.012.

[58] Jie Zhang and Robin Cohen. "Evaluating the trustworthiness of advice about seller agents in e-marketplaces: A personalized approach". In: *Electronic Commerce Research and Applications,* vol. 7, (Sept. 2008), pp. 330–340. DOI: 10.1016/j.elerap.2008.03.001.

[59] Trung Huynh, Nicholas Jennings, and Nigel Shadbolt. "Developing an Integrated Trust and Reputation Model for Open Multi-Agent Systems". In: *Autonomous Agents and Multi-Agent Systems,* vol. 13, (Sept. 2006), pp. 119–154. DOI: 10.1007/s10458-005-6825-4.

[60] Kiyana Zolfaghar and Abdollah Aghaie. "Evolution of trust networks in social web applications using supervised learning". In: *Procedia Computer Science,* vol. 3, (Dec. 2011), pp. 833–839. DOI: 10.1016/j.procs.2010.12.137.

[61] Xin Wang, Ying Wang, and Hongbin Sun. "Exploring the Combination of Dempster-Shafer Theory and Neural Network for Predicting Trust and Distrust". In: *Computational Intelligence and Neuroscience,* vol. 2016, (Jan. 2016), pp. 1–12. DOI: 10.1155/2016/5403105.

[62] Xu Chen, Yuyu Yuan, Lilei Lu, and Jincui Yang. "A Multidimensional Trust Evaluation Framework for Online Social Networks Based on Machine Learning". In: *IEEE Access,* vol. 7, (2019), pp. 175499–175513. DOI: 10.1109/ACCESS.2019.2957779.

[63] Sarhad Arisdakessian, Omar Abdel Wahab, Azzam Mourad, Hadi Otrok, and Mohsen Guizani. "A Survey on IoT Intrusion Detection: Federated Learning, Game Theory, Social Psychology and Explainable AI as Future Directions". In: *IEEE Internet of Things Journal* (2022), pp. 1–1. DOI: 10.1109/JIOT.2022.3203249.

[64] Sang H. Chin. "On application of game theory for understanding trust in networks". In: *International Symposium on Collaborative Technologies and Systems*. 2009, pp. 106–110. DOI: 10.1109/CTS.2009.5067469.

[65] Christian Esposito, Oscar Tamburis, Xin Su, and Chang Choi. "Robust Decentralised Trust Management for the Internet of Things by Using Game Theory". In: *Information Processing and Management,* vol. 57, no. 6, (June 2020), p. 102308. DOI: 10.1016/j.ipm.2020.102308.

[66] Mohammad Mahdi Azadjalal, Parham Moradi, and Alireza Abdollahpouri. "Application of game theory techniques for improving trust based recommender systems in social networks". In: *4th International Conference on Computer and Knowledge Engineering (ICCKE)*. 2014, pp. 261–266. DOI: 10.1109/ICCKE.2014.6993436.

[67] Rinki Rani, Sushil Kumar, and Upasana Dohare. "Trust Evaluation for Light Weight Security in Sensor Enabled Internet of Things: Game Theory Oriented Approach". In: *IEEE Internet of Things Journal,* vol. 6, no. 5, (2019), pp. 8421–8432. DOI: 10.1109/JIOT.2019.2917763.

[68] Yan Lindsay Sun, Wei Yu, Zhu Han, and K.J. Ray Liu. "Information theoretic framework of trust modeling and evaluation for ad hoc networks". In: *IEEE Journal on Selected Areas in Communications,* vol. 24, no. 2, (2006), pp. 305–317. DOI: 10.1109/JSAC.2005.861389.

[69] Yan Sun, Wei Yu, Zhu Han, and K.J. Ray Liu. "Trust modeling and evaluation in ad hoc networks". In: *IEEE Global Telecommunications Conference (GLOBECOM)*. Vol. 3. 2005, p. 6. DOI: 10.1109/GLOCOM.2005.1577971.

[70] Shuhao Jiang, Jincheng Ding, and Liyi Zhang. "A Personalized Recommendation Algorithm Based on Weighted Information Entropy and Particle Swarm Optimization". In: *Mobile Information Systems,* vol. 2021, (Dec. 2021), pp. 1–9. DOI: 10.1155/2021/3209140.

[71] Yin Xueqiang and Shining Li. "Trust evaluation model with entropy-based weight assignment for malicious node's detection in wireless sensor networks". In: *EURASIP Journal on Wireless Communications and Networking,* vol. 2019, no. 1, (Aug. 2019), pp. 1–10. DOI: 10.1186/s13638-019-1524-z.

[72] Yonghua Gong and Lei Chen. "A Trust Model for Evaluating Trust and Centrality Based on Entropy Weight Method in Social Commerce". In: Aug. 2020, pp. 40–50. ISBN: 978-981-15-7529-7. DOI: 10.1007/978-981-15-7530-3_4.

[73] Hannah Lim Jing Ting, Xin Kang, Tieyan Li, Haiguang Wang, and Cheng-Kang Chu. "On the Trust and Trust Modeling for the Future Fully-Connected Digital World: A Comprehensive Study". In: *IEEE Access,* vol. 9, (2021), pp. 106743–106783. DOI: 10.1109/ACCESS.2021.3100767.

[74] Agus Kurniawan and Marcel Kyas. "A trust model-based Bayesian decision theory in large scale Internet of Things". In: *IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP).* 2015, pp. 1–5. DOI: 10.1109/ISSNIP.2015.7106964.

[75] A. Ben Abdessalem, Nikolaos Dervilis, D. Wagg, and Keith Worden. "An efficient likelihood-free Bayesian computation for model selection and parameter estimation applied to structural dynamics". In: *Structural Health Monitoring, Photogrammetry & DIC.* 2018, pp. 141–151.

[76] Saurabh Ganeriwal, Laura Balzano, and Mani Srivastava. "Reputation-based framework for high integrity sensor networks". In: *ACM Transactions on Sensor Networks (TOSN),* vol. 4, no. 3, (Jan. 2003), pp. 1–37. DOI: 10.1145/1029102.1029115.

[77] L. Mui, Mojdeh Mohtashemi, C. Ang, Peter Szolovits, and A. Halberstadt. "Bayesian Ratings in Distributed Systems: Theories, Models, and Simulations". In: (Nov. 2022).

[78] Vijender Busi Reddy, Sarma Venkataraman, and Atul Negi. "Communication and Data Trust for Wireless Sensor Networks Using D–S Theory". In: *IEEE Sensors Journal,* vol. 17, no. 12, (2017), pp. 3921–3929. DOI: 10.1109/JSEN.2017.2699561.

[79] Mohammad Karami and Mohammad Fathian. "A robust trust establishment framework using Dempster-Shafer theory for MANETs". In: *International Conference for Internet Technology and Secured Transactions, (ICITST).* 2009, pp. 1–7. DOI: 10.1109/ICITST.2009.5402628.

[80] Ing-Ray Chen, Fenye Bao, and Jia Guo. "Trust-Based Service Management for Social Internet of Things Systems". In: *IEEE Transactions on Dependable and Secure Computing,* vol. 13, no. 6, (2016), pp. 684–696. DOI: 10.1109/TDSC.2015.2420552.

[81] Fan Chung, Alexander Tsiatas, and Wensong Xu. "Dirichlet PageRank and Trust-Based Ranking Algorithms". In: *International Workshop on Algorithms and Models for the Web-Graph*. 2011, pp. 103–114. DOI: 10.1007/978-3-642-21286-4_9.

[82] Yali Gao, Xiaoyong Li, Jirui Li, Yunquan Gao, and Philip S. Yu. "InfoTrust: A Multi-Criteria and Adaptive Trustworthiness Calculation Mechanism for Information Sources". In: *IEEE Access,* vol. 7, (2019), pp. 13999 –14012. DOI: 10.1109/ACCESS.2019.2893657.

[83] Chung-Wei Hang and Munindar P. Singh. "Trust-based recommendation based on graph similarity". In: *The 13th International Workshop on Trust in Agent Societies (TRUST). Toronto, Canada*. Vol. 82. 2010.

[84] Jiliang Tang, Huiji Gao, and Huan Liu. "mTrust: Discerning Multi-Faceted Trust in a Connected World". In: *The 5th ACM International Conference on Web Search and Data Mining (WSDM 2012)*. 2012, pp. 93–102. DOI: 10.1145/2124295.2124309.

[85] Mansooreh Ezhei and Behrouz Tork Ladani. "GTrust: a group based trust model". In: *ISC Int. J. Inf. Secur.,* vol. 5, no. 2, (2013), pp. 155–170.

[86] Tomislav Duricic, Emanuel Lacic, Dominik Kowald, and Elisabeth Lex. "Trust-based collaborative filtering: tackling the cold start problem using regular equivalence". In: *The 12th ACM Conference on Recommender Systems*. 2018, pp. 446–450. DOI: 10.1145/3240323.3240404.

[87] Frank E Walter, Stefano Battiston, and Frank Schweitzer. "Personalised and dynamic trust in social networks". In: *The third ACM conference on Recommender systems*. 2009, pp. 197–204. DOI: 10.1145/1639714.1639747.

[88] George Theodorakopoulos and John S Baras. "On trust models and trust evaluation metrics for ad hoc networks". In: *IEEE Journal on Selected Areas in Communications,* vol. 24, no. 2, (2006), pp. 318–328. DOI: 10.1109/JSAC.2005.861390.

[89] Li Yu, Jing-ru Li, and Zu-hao Liu. "Semiring Trust Model Based on Adaptive Forgetting Scheme". In: *Journal of Electronics Information Technology,* vol. 33, no. 1, (Feb. 2011), pp. 175–179. DOI: 10.3724/SP.J.1146.2010.00221.

[90] Kiran Somasundaram and John Baras. "Path Optimization and Trusted Routing in MANET: An Interplay between Ordered Semirings". In: *Communications in Computer and Information Science*. 2011, pp. 88–98. DOI: 10.1007/978-3-642-17878-8_10.

[91] Wenjia Li, Anupam Joshi, and Tim Finin. "SAT: an SVM-based automated trust management system for Mobile Ad-hoc Networks". In: *Military Communications Conference (MILCOM 2011).* 2011, pp. 1102–1107. DOI: 10.1109/MILCOM.2011.6127446.

[92] Nirav J.Patel and Rutvij Jhaveri. "Detecting Packet Dropping Misbehaving Nodes using Support Vector Machine (SVM) in MANET". In: *International Journal of Computer Applications,* vol. 122, (July 2015), pp. 26–32. DOI: 10.5120/21689-4794.

[93] Naeem Shahabi Sani and Ferial Najian. "A New Strategy in Trust-Based Recommender System using K-Means Clustering". In: *International Journal of Advanced Computer Science and Applications,* vol. 8, no. 9, (Jan. 2017), 152–156. DOI: 10.14569/IJACSA.2017.080922.

[94] David Airehrour, Jairo Gutierrez, and Sayan Kumar Ray. "A Lightweight Trust Design for IoT Routing". In: *IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress.* 2016, pp. 552–557. DOI: 10.1109/DASC-PICom-DataCom-CyberSciTec.2016.105.

[95] Fayez Alqahtani, Zafer Al-Makhadmeh, Amr Tolba, and Omar Said. "TBM: A trust-based monitoring security scheme to improve the service authentication in the Internet of Things communications". In: *Computer Communications,* 150, (Jan. 2020), pp. 216–225. DOI: 10.1016/j.comcom.2019.11.030.

[96] Meena Kowshalya and Valarmathi ml. "Trust Management for Reliable Decision Making among Social Objects in the Social Internet of Things". In: *IET Networks,* vol. 6, no. 4, (Apr. 2017), pp. 75–80. DOI: 10.1049/iet-net.2017.0021.

[97] Luigi Atzori, Antonio Iera, and Giacomo Morabito. "SIoT: Giving a Social Structure to the Internet of Things". In: *IEEE Communications Letters,* vol. 15, no. 11, (2011), pp. 1193–1195. DOI: 10.1109/LCOMM.2011.090911.111340.

[98] Weidong Fang, Ming Xu, Chunsheng Zhu, Weili Han, Wuxiong Zhang, and Joel J. P. C. Rodrigues. "FETMS: Fast and Efficient Trust Management Scheme for Information-Centric Networking in Internet of Things". In: *IEEE Access,* vol. 7, (2019), pp. 13476–13485. DOI: 10.1109/ACCESS.2019.2892712.

[99] Tahani Gazdar, Abderrezak Rachedi, Abderrahim Benslimane, and Abdelfettah Belghith. "A distributed advanced analytical trust model for VANETs". In: *IEEE Global Communications Conference (GLOBECOM).* 2012, pp. 201–206. DOI: 10.1109/GLOCOM.2012.6503113.

[100] Abderrezak Rachedi and Abderrahim Benslimane. "Toward a cross-layer monitoring process for mobile ad hoc networks". In: *Security and communication networks,* vol. 2, (July 2009), pp. 351–36. DOI: 10.1002/sec.72.

[101] Aravind Iyer, Arzad Kherani, Ashwin Rao, and Aditya Karnik. "Secure V2V communications: Performance impact of computational overheads". In: *IEEE INFOCOM Workshops 2008*. 2008, pp. 1–6. DOI: 10.1109/INFOCOM.2008.4544660.

[102] G.D. Hachtel, Enrico Macii, Abelardo Pardo, and Fabio Somenzi. "Markovian analysis of large finite state machines". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 15, no. 12, (1996), pp. 1479–1493. DOI: 10.1109/43.552081.

[103] Erik Hofmann and Marco Rüsch. "Industry 4.0 and the current status as well as future prospects on logistics". In: *Computers in Industry,* vol. 89, (Aug. 2017), pp. 23–34. DOI: 10.1016/j.compind.2017.04.002.

[104] Guo-Jian Cheng, Li-Ting Liu, Xin-Jian Qiang, and Ye Liu. "Industry 4.0 Development and Application of Intelligent Manufacturing". In: *International Conference on Information System and Artificial Intelligence (ISAI)*. 2016, pp. 407–410. DOI: 10.1109/ISAI.2016.0092.

[105] ATCC Finance. "Industry 4.0 Challenges and solutions for the digital transformation and use of exponential technologies". In: *Finance, Audit Tax Consulting Corporate: Zurich, Swiss* (2015), pp. 1–12.

[106] Soumya Kanti Datta and Christian Bonnet. "Securing IoT Platforms". In: *IEEE International Conference on Consumer Electronics (ICCE)*. 2019, pp. 1–2. DOI: 10.1109/ICCE.2019.8661966.

[107] Florence D. Hudson, Phillip A. Laplante, and Ben Amaba. "Enabling Trust and Security: TIPPSS for IoT". In: *IT Professional,* vol. 20, no. 2, (2018), pp. 15–18. DOI: 10.1109/MITP.2018.021921646.

[108] Arman Pouraghily, Md Nazmul Islam, Sandip Kundu, and Tilman Wolf. "Poster Abstract: Privacy in Blockchain-Enabled IoT Devices". In: *IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*. 2018, pp. 292–293. DOI: 10.1109/IoTDI.2018.00045.

[109] Chao Li and Balaji Palanisamy. "Privacy in Internet of Things: From Principles to Technologies". In: *IEEE Internet of Things Journal,* vol. 6, no. 1, (2019), pp. 488–505. DOI: 10.1109/JIOT.2018.2864168.

[110] Abbas M. Hassan and Ali Ismail Awad. "Urban Transition in the Era of the Internet of Things: Social Implications and Privacy Challenges". In: *IEEE Access,* vol. 6, (2018), pp. 36428–36440. DOI: 10.1109/ACCESS.2018.2838339.

[111] David P. Fidler. "Was Stuxnet an Act of War? Decoding a Cyberattack". In: *IEEE Security Privacy,* vol. 9, no. 4, (2011), pp. 56–59. DOI: 10.1109/MSP.2011.96.

[112] Fenye Bao and Ing-Ray Chen. "Dynamic trust management for internet of things applications". In: *Self-IoT'12 - Proceedings of the 2012 International Workshop on Self-Aware Internet of Things, Co-located with ICAC'12*. Sept. 2012, pp. 1–6. DOI: 10.1145/2378023.2378025.

[113] Nabil Djedjig, Djamel Tandjaoui, and Faiza Medjek. "Trust-based RPL for the Internet of Things". In: *IEEE Symposium on Computers and Communication (ISCC)*. 2015, pp. 962–967. DOI: 10.1109/ISCC.2015.7405638.

[114] Panagiotis Karkazis, Helen C. Leligou, Lambros Sarakis, Theodore Zahariadis, Panagiotis Trakadas, Terpsichori H. Velivassaki, and Christos Capsalis. "Design of primary and composite routing metrics for RPL-compliant Wireless Sensor Networks". In: *International Conference on Telecommunications and Multimedia (TEMU)*. 2012, pp. 13–18. DOI: 10.1109/TEMU.2012.6294705.

[115] Panagiotis Karkazis, Ioannis Papaefstathiou, Lambros Sarakis, Theodore Zahariadis, Terpsichori-Helen Velivassaki, and Dimitrios Bargiotas. "Evaluation of RPL with a transmission count-efficient and trust-aware routing metric". In: *IEEE International Conference on Communications (ICC)*. 2014, pp. 550–556. DOI: 10.1109/ICC.2014.6883376.

[116] Ing-Ray Chen, Jia Guo, and Fenye Bao. "Trust Management for SOA-Based IoT and Its Application to Service Composition". In: *IEEE Transactions on Services Computing,* vol. 9, no. 3, (2016), pp. 482–495. DOI: 10.1109/TSC.2014.2365797.

[117] Hamid Al-Hamadi and Ing Ray Chen. "Trust-Based Decision Making for Health IoT Systems". In: *IEEE Internet of Things Journal,* vol. 4, no. 5, (2017), pp. 1408–1419. DOI: 10.1109/JIOT.2017.2736446.

[118] Antonio L. Maia Neto, Yuri L. Pereira, Artur L. F. Souza, Italo Cunha, and Leonardo B. Oliveira. "Demo Abstract: Attributed-Based Authentication and Access Control for IoT Home Devices". In: *17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 2018, pp. 112–113. DOI: 10.1109/IPSN.2018.00019.

[119] Yosra Saied, Alexis Olivereau, Djamal Zeghlache, and Maryline Laurent. "Trust management system design for the Internet of Things: A context-aware and multi-service approach". In: *Computers Security,* vol.39, (Nov. 2013), pp. 351–365. DOI: 10.1016/j.cose.2013.09.001.

[120] Hugh Boyes, Bil Hallaq, Joe Cunningham, and Tim Watson. "The industrial internet of things (IIoT): An analysis framework". In: *Computers in industry,* vol.101, (2018), pp. 1–12.

[121] Mohammad Alshehri, Farookh Hussain, and Omar Hussain. "Clustering-Driven Intelligent Trust Management Methodology for the Internet of Things (CITM-IoT)". In: *Mobile Networks and Applications,* vol. 23, (June 2018), pp. 419–431. DOI: 10.1007/s11036-018-1017-z.

[122] Oumaima Ben Abderrahim, Mohamed Houcine Elhdhili, and Leila Saidane. "TMCoI-SIOT: A trust management system based on communities of interest for the social Internet of Things". In: *13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. 2017, pp. 747–752. DOI: 10.1109/IWCMC.2017.7986378.

[123] Wafa Abdelghani, Corinne Zayani, Ikram Amous, and Florence Sedes. "Trust Management in Social Internet of Things: A Survey". In: *Social Media: The Good, the Bad, and the Ugly.* Vol. vol. 9844. Sept. 2016, pp. 430–441. DOI: 10.1007/978-3-319-45234-0_39.

[124] Fenye Bao, Ing-Ray Chen, and Jia Guo. "Scalable, adaptive and survivable trust management for community of interest based Internet of Things systems". In: *IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS).* Vol. vol. 2013, pp. 1–7. DOI: 10.1109/ISADS.2013.6513398.

[125] Asma Lahbib, Khalifa Toumi, Anis Laouiti, Alexandre Laube, and Steven Martin. "Blockchain based trust management mechanism for IoT". In: *IEEE Wireless Communications and Networking Conference (WCNC).* 2019, pp. 1–8. DOI: 10.1109/WCNC.2019.8885994.

[126] Axel Moinet, Benoit Darties, and Jean-Luc Baril. "Blockchain based trust authentication for decentralized sensor networks". In: *ArXiv,* vol. abs/1706.01730, (June 2017).

[127] Jolyon Clulow and Tyler Moore. "Suicide for the common good: a new strategy for credential revocation in self-organizing systems". In: *ACM SIGOPS Operating Systems Review,* vol. 40, (July 2006), pp. 18–21. DOI: 10.1145/1151374.1151381.

[128] Dipti S Sawant and E Jayanthi. "Cluster-based certificate revocation in mobile ad-hoc network using fuzzy logic". In: *International Journal of Computer Engineering and Applications,* vol. 9, no. 7, (2015), pp. 2321–3469.

[129] Kyul Park, Hiroki Nishiyama, Nirwan Ansari, and Nei Kato. "Certificate Revocation to Cope with False Accusations in Mobile Ad Hoc Networks". In: *2010 IEEE 71st Vehicular Technology Conference.* 2010, pp. 1–5. DOI: 10.1109/VETECS.2010.5494213.

[130] E. Jayanthi and Mohammed Ali Hussain. "A Novel Approach Certificate Revocation in MANET using Fuzzy logic". In: *Indonesian Journal of Electrical Engineering and Computer Science,* vol. 10, (May 2018), pp. 654–663. DOI: 10.11591/ijeecs.v10.i2.pp654-663.

[131] Haiyun Luo, Jiejun Kong, P. Zerfos, Songwu Lu, and Lixia Zhang. "URSA: ubiquitous and robust access control for mobile ad hoc networks". In: *IEEE/ACM Transactions on Networking,* vol. 12, no. 6, (2004), pp. 1049–1063. DOI: 10.1109/TNET.2004.838598.

[132] Sungwook Kim. "Effective certificate revocation scheme based on weighted voting game approach". In: *IET Information Security,* vol. 10, (Oct. 2015), pp. 180–187. DOI: 10.1049/iet-ifs.2015.0047.

[133] Santhana .R, Golden Julie, Y. Robinson, Raghvendra Kumar, Thong Pham, and Le Son. "Enhanced Certificate Revocation Scheme with Justification Facility in Mobile Ad-hoc Networks". In: *Computers Security,* vol. 97, (July 2020), pp. 101962. DOI: 10.1016/j.cose.2020.101962.

[134] H.L. Bhavyashree, C.R. Nagarathna, Anusha Preetham, and R. Priyanka. "Modified Cluster Based Certificate Blocking of Misbehaving Node In Manets". In: *1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing Communication Engineering (ICATIECE)*. 2019, pp. 155–161. DOI: 10.1109/ICATIECE45860.2019.9063622.

[135] Roger B Myerson. "Comments on "Games with Incomplete Information Played by 'Bayesian'Players, I–III Harsanyi's Games with Incoplete Information"". In: *Management Science,* vol.50,12_supplement (2004), pp. 1818–1824.

[136] Drew Fudenberg and Jean Tirole. *Game theory*. MIT press, 1991.

[137] Shigen Shen, Yuanjie Li, Hongyun Xu, and Qiying Cao. "Signaling game based strategy of intrusion detection in wireless sensor networks". In: *Computers Mathematics with Applications,* vol. 62, (Sept. 2011), pp. 2404–2416. DOI: 10.1016/j.camwa.2011.07.027.

[138] Daniel Díaz-Sánchez, Andrés Marín-Lopez, Florina Almenárez Mendoza, Patricia Arias Cabarcos, and R. Simon Sherratt. "TLS/PKI Challenges and Certificate Pinning Techniques for IoT and M2M Secure Communications". In: *IEEE Communications Surveys Tutorials,* vol. 21, no. 4, (2019), pp. 3502–3531. DOI: 10.1109/COMST.2019.2914453.

[139] Emin Topalovic, Brennan Saeta, Lin-Shung Huang, Collin Jackson, Dan Boneh, and Stanford. "Towards Short-Lived Certificates". In: (Nov. 2022), 1–9.

[140] Marc-Oliver Pahl and Lorenzo Donini. "Giving IoT Services an Identity and Changeable Attributes". In: *IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. 2019, pp. 455–461.

[141] Seunghwan Ju and Hee-Suk Seo. "Certificate Management Scheme for IoT Services". In: *The Mattingley Publishing Co., Inc.,* vol. 83, (Apr. 2020), pp. 4186–4194. DOI: 10.13140/RG.2.2.18478.66880.

[142] Yung-Kao Hsu and S. Seymour. "Intranet security framework based on short-lived certificates". In: *IEEE 6th Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*. 1997, pp. 228–234. DOI: 10.1109/ENABL.1997.630819.

[143] Shahid Mahmood, Moneeb Gohar, Jin-Ghoo Choi, Seok-Joo Koh, Hani Alquhayz, and Murad Khan. "Digital Certificate Verification Scheme for Smart Grid using Fog Computing (FONICA)". In: *Sustainability,* vol. 13, (Feb. 2021), pp. 2549. DOI: 10.3390/su13052549.

[144] Joel Höglund, Samuel Lindemer, Martin Furuhed, and Shahid Raza. "PKI4IoT: Towards Public Key Infrastructure for the Internet of Things". In: *Computers Security,* vol. 89, (Nov. 2019), pp. 101658. DOI: 10.1016/j.cose.2019.101658.

[145] Li Duan, Yong Li, and Lijun Liao. "Flexible certificate revocation list for efficient authentication in IoT". In: *The 8th International Conference on the Internet of Things*. Oct. 2018, pp. 1–8. DOI: 10.1145/3277593.3277595.

[146] Mumin Cebe and Kemal Akkaya. "Efficient Certificate Revocation Management Schemes for IoT-based Advanced Metering Infrastructures in Smart Cities". In: *Ad Hoc Networks,* vol. 92, (Oct. 2018), pp. 1570–8705. DOI: 10.1016/j.adhoc.2018.10.027.

[147] Mohammad Sayad Haghighi, Maryam Ebrahimi, Sahil Garg, and Alireza Jolfaei. "Intelligent Trust-Based Public-Key Management for IoT by Linking Edge Devices in a Fog Architecture". In: *IEEE Internet of Things Journal,* vol. 8, no. 16, (2021), pp. 12716–12723. DOI: 10.1109/JIOT.2020.3027536.

[148] Stefan Santesson, Michael Myers, Rich Ankney, Ambarish Malpani, Slava Galperin, and Carlisle Adams. *"X. 509 internet public key infrastructure online certificate status protocol-OCSP". No. rfc6960. Internet Engineering Task Force (IETF).* Tech. rep. 2013.

[149] Eric Rescorla. *The transport layer security (TLS) protocol version 1.3.* Tech. rep. 2018. DOI: 10.17487/RFC8446.

[150] Konstantinos Papapanagiotou, Giannis Marias, and P Georgiadis. "Revising centralized certificate validation standards for mobile and wireless communications". In: *Computer Standards Interfaces,* vol. 32, (Oct. 2010), pp. 281–287. DOI: 10.1016/j.csi.2009.07.001.

[151] Hongyu Jin and Panos Papadimitratos. "Bloom filter based certificate validation for VANET: poster". In: *The 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks.* July 2017, pp. 273–274. DOI: 10.1145/3098243.3106020.

[152] Khaled Rabieh, Mohamed M.E.A. Mahmoud, Kemal Akkaya, and Samet Tonyali. "Scalable Certificate Revocation Schemes for Smart Grid AMI Networks Using Bloom Filters". In: *IEEE Transactions on Dependable and Secure Computing,* vol. 14, no. 4, (2017), pp. 420–432. DOI: 10.1109/TDSC.2015.2467385.

[153] Haoyu Song, Sarang Dharmapurikar, Jonathan Turner, and John Lockwood. "Fast Hash Table Lookup Using Extended Bloom Filter: An Aid to Network Processing". In: *ACM SIGCOMM Computer Communication Review,* vol. 35, (Oct. 2005), pp. 181–192. DOI: 10.1145/1090191.1080114.

[154] Kemal Akkaya, Kalid Rabeh, Mohamed Mahmoud, and Samet Tonyalı. "Efficient generation and distribution of CRLs for IEEE 802.11s-based Smart Grid AMI networks". In: *IEEE International Conference on Smart Grid Communications, SmartGridComm 2014.* Nov. 2014, pp. 982– 988. DOI: 10.1109/SmartGridComm.2014.7007776.

[155] Mohammad Masdari, Sam Jabbehdari, and J. Bagherzadeh. "Improving OCSP-Based Certificate Validations in Wireless Ad Hoc Networks". In: *Wireless Personal Communications,* vol. 82, (Dec. 2014), pp. 377–400. DOI: 10.1007/s11277-014-2213-1.

[156] Nikolay Teslya and Igor Ryabchikov. "Blockchain-based platform architecture for industrial IoT". In: *21st Conference of Open Innovations Association (FRUCT)*. 2017, pp. 321–329. DOI: 10.23919/FRUCT.2017.8250199.

[157] Jasone Astorga, Marc Barcelo, Aitor Urbieta, and Eduardo Jacob. "How to Survive Identity Management in the Industry 4.0 Era". In: *IEEE Access,* vol. 9, (2021), pp. 93137–93151. DOI: 10.1109/ACCESS.2021.3092203.

[158] Badis Hammi, Ahmed Serhrouchni, Sherali Zeadally, and Adja Elloh Yves Christian. "A Blockchain-based Certificate Revocation Management and Status Verification System". In: *Computers Security,* vol. 104, (Jan. 2021), p. 102209. DOI: 10.1016/j.cose.2021.102209.

[159] Tim Winter, Pascal Thubert, Anders Brandt, Jonathan Hui, Richard Kelsey, Philip Levis, Kris Pister, Rene Struik, Jean-Philippe Vasseur, and Roger Alexander. *RPL: IPv6 routing protocol for low-power and lossy networks*. Tech. rep. 2012.