UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Trust Management in Service-Oriented Internet of Things (SO-IoT)

THÈSE

présentée et soutenue publiquement le July 5, 2024

pour l'obtention du

Doctorat de l'Université de Lorraine

(Docteur en informatique)

par

Runbo Su

Composition du jury

Président :	Abdelmadjid Bouabdalla, UTC (Université de technologie de Compiègne)
Rapporteurs :	Valeria Loscrì, Inria Lille-Nord Est Abdelmadjid Bouabdalla, UTC
Examinateurs :	Claudia-Lavinia Ignat, Inria-Nancy Grand Est Anna Maria Mandalari, University College London
Invités :	Ye-Qiong Song, Loria, Université de Lorraine Riahi Arbia, Cylab, CISS, Royal Military Academy, Belgium
Encadrants :	Enrico Natalizio, Loria; TII (Technology Innovation Institute), UAE Pascal Moyal, IECL, Université de Lorraine

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503
Institut Élie Cartan de Lorraine — UMR 7502

Inria cnrs Loria Laboratoire lorrain de recherche en informatique et ses applications Institut ÉLIE CARTAN

Acknowledgments

First and foremost, I would like to express my gratitude to my respected supervisors, Enrico Natalizio and Pascal Moyal, for their unwavering support, invaluable experience, and consistent encouragement, which have formed the foundation of my research. I am eternally grateful for the countless hours they dedicated to assisting me in overcoming difficulties and realizing the full potential of my work.

To all of the co-authors of the papers, Arbia, Ye-Qiong, Amaury, and Yujun, The collective effort and synergy we shared have significantly improved the quality of my work. I would also like to thank the anonymous reviewers who took the time to read and provide constructive remarks on my work.

My heartfelt gratitude goes to Christine, Vincent, Laurent, Sylvain, Virginie, Jean-Louis, Emmanuel, and Ludovic, as well as other peer members at the SIMBIOT team of Loria Lab, Théo, Debashisha, Jean-Baptiste, Thomas, Virgile, Norhane, Maxime, Diego, Solayman, Ikram, Idriss. Our team's collaborative environment and intellectual exchanges have enriched my research experience, and I am thankful for all the discussions and feedback that have made my PhD journey motivating and enjoyable. I would also like to address my particular thanks to the member of the monitoring committee, Isabelle Chrisment.

Specifically, I want to acknowledge your countless support and belief in me for all my family members, especially my parents and my wife Xinyue. Your patience, understanding, and sacrifices have been the bedrock upon which I built my PhD pursuits. Your love and encouragement have sustained me through the challenges.

All in all, thanks to everyone who has helped me directly or indirectly while earning my doctoral degree. You all have made it possible for me to achieve this goal.

Contents

List of Figures

List of Tables

List of Publications

[1] **Runbo Su**, Arbia Riahi Sfar, Enrico Natalizio, Pascal Moyal, and Ye-Qiong Song. "PDTM: Phase-based dynamic trust management for Internet of things." In 2021 International Conference on Computer Communications and Networks (ICCCN), pp. 1-7. IEEE, 2021.

[2] **Runbo Su**, Arbia Riahi Sfar, Enrico Natalizio, Pascal Moyal, and Ye-Qiong Song. "Ensuring Trustworthiness in IoIT/AIoT: A Phase-Based Approach." IEEE Internet of Things Magazine 5, no. 2 (2022): 84-88.

[3] **Runbo Su**, Yujun Jin, Ye-Qiong Song. "A lightweight cooperative trust model for IoV." Poster Paper In 2023 International Conference on Computer Communications and Networks (ICCCN).

[4] **Runbo Su**, Arbia Riahi Sfar, Enrico Natalizio, Pascal Moyal, Ye-Qiong Song. "A Game Theoretical Model addressing Misbehavior in Crowdsourcing IoT." In 20th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON 2023), pp. 195–203, IEEE, 2023.

[5] **Runbo Su**, Amaury Saint-Jore. "A Role-based Trust Model assessing IoA services: First Results on Real MAS Implementation by using ROS 2." The International Conference on Embedded Wireless Systems and Networks (EWSN 2023), pp. 293-296, ACM, 2024.

[6] **Runbo Su**, Yujun Jin, Ye-Qiong Song. "Assessing trustworthiness of V2X messages: a cooperative trust model against CAM- and CPM-based Ghost Vehicles in IoV" To Appear In Proceedings of 10th Vehicle Technology and Intelligent Transport Systems (VEHITS 2024), pp. 276–283.

[7] **Runbo Su**, Arbia Riahi Sfar, and Pascal Moyal. "Game theoretical analysis of strategy changes and influence factors in Crowdsourcing IoT systems." To Appear In Proceedings of International Conference on Distributed Computing In Smart Systems and the Internet of Things (Dcoss-Iot 2024).

[8] **Runbo Su**, Arbia Riahi Sfar, Enrico Natalizio, Pascal Moyal, Amaury Saint-jore, and Ye-Qiong Song. "Assessing intra- and inter-community trustworthiness in IoT: a role-based attack-resilient dynamic trust management model." Internet of Things Elsevier, 26, p.101213.

Chapter 1

Introduction

1.1 Trust: From Social Science to IoT Security

1.1.1 Trust in Social Science

Social scientists have long regarded trust as a fundamental element for the functioning and sustainability of human society, which plays a crucial role in various social interactions, relationships, and institutions. As stated by authors in [9], the Trust process contains (a) a trustor, (b) a trustee, and (c) a trust object, more precisely, *"A trusts B with respect to issue x', where A is the trustor (the actor placing trust), B is the trustee (the target of trust), and x is the trust object (the domain or activity in which trust is placed)."* In 1988, the author in [10] defined Trust as the trustor's subjective probability about whether the trustee will perform a particular action that benefits the trustor. Later, in 2002, the authors in [11] analyzed the Trust concept in Economics: *"To say 'A trusts B' means that A expects B will not exploit a vulnerability A has created for himself by taking action."* Furthermore, in Social Psychology, e.g., the work in [12], a degree of subjective belief towards the behavior of a particular entity is described as Trust.

As illustrated in Fig. 1.1, authors defined *"each contributes a unique perceptual perspective from which to consider the trustee, while the set provides a solid and parsimonious foundation for the empirical study of trust for another party".*

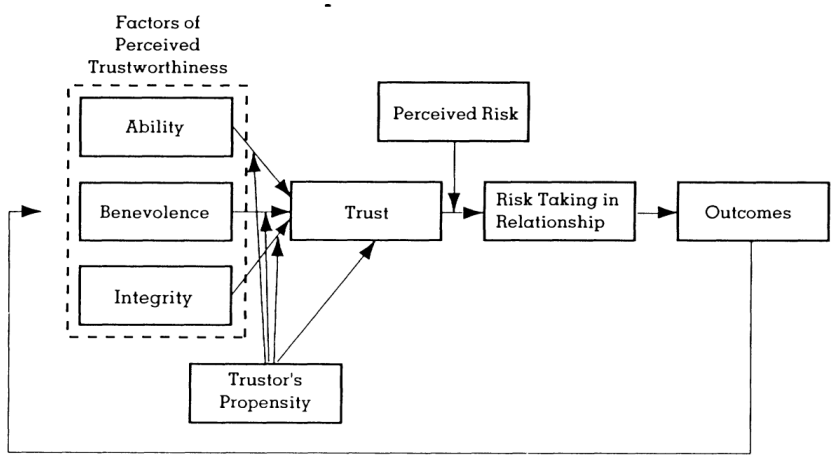


Figure 1.1: Trust model in Social Science [13]

Based on the above studies, in social science, trust is generally defined as the trustor's psychological state, which places the expectation of positive outcomes in the trustee's action despite the trustee not being under the trustor's control. Besides, it should also be noted that several synonyms associated with Trust are often employed as alternatives: confidence, belief, credibility, trustworthiness, etc. Understanding trust in social science provides insights into the dynamics of relationships and structures of trustors and trustees, as well as the mechanisms that facilitate their cooperation and collaboration. Researchers across disciplines, including computer science, explore trust to comprehensively employ the utilities of trust underlying secure systems composed of reliable trustors and trustees.

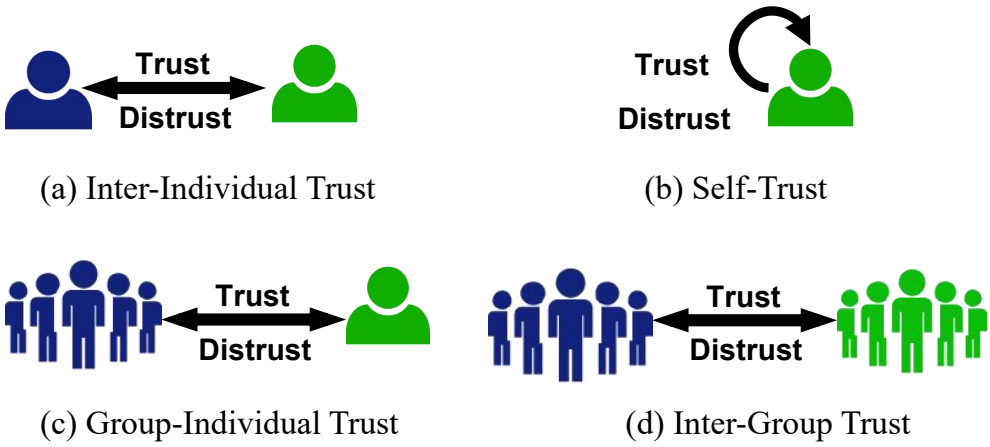


Figure 1.2: Trust in Social Science according to 'action correlators'.

Fig. 1.2 demonstrates that Trust in Social Science is a multifaceted concept encompassing various dimensions that define interpersonal and collective relationships. Inter-individual trust refers to the confidence individuals place in one another, forming the foundation of personal relationships. Self-trust involves an individual's reliance on their own judgments and abilities. Inter-Group trust delves into the dynamics of trust between different social groups, fostering cooperation and collaboration. Group-Individual trust explores the reciprocal relationship between an individual and a larger social entity, such as a community or organization. Each type of trust

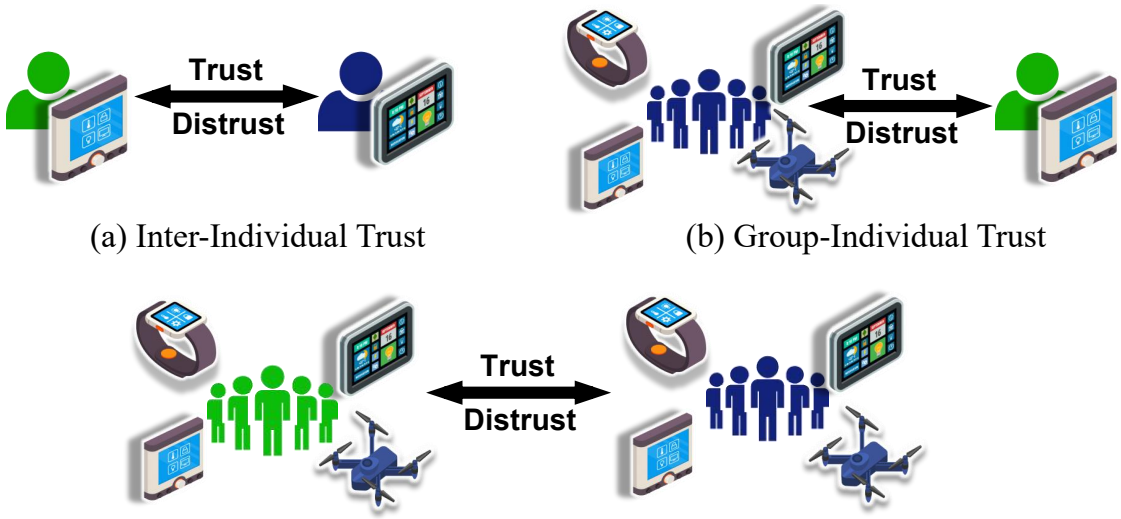
plays a pivotal role in shaping social interactions, influencing cooperation, and establishing the fabric of societal bonds. Understanding and analyzing these dimensions of trust contribute to a more comprehensive comprehension of human behavior, societal structures, and the mechanisms that underpin functional communities. Trust, in its diverse forms, serves as a linchpin for the cohesion and functionality of social systems, making it a significant focus in the domain of Social Science [14].

1.1.2 Trust in IoT Security

Since 'Industry 4.0' was first proposed in Germany [15], the interest and need in using IoT technologies are rapidly growing. So far, several countries have announced their national Industry 4.0 plan, such as 'Made in China 2025' [16], released in 2015. IoT enables smart devices (thereafter referred to as "nodes") to be connected and operated over a network. In such a way, physical processing can be visualized by using numerical descriptions, where diverse services can be associated, classified, and assessed, also meaning that IoT applications focus on offering an automated and efficient environment where a massive number of nodes can collaboratively assist in service provision and evaluation. Despite these investments and developments, security issues encountered in IoT remain challenging. Till now, more than 90% of companies are vulnerable to cyber-attacks, according to new research from Positive Technologies [17]. The motivations of such attacks are diverse, including financial gain, espionage, and even criminal goals for creating disruptions and casualties. For this reason, a mechanism that monitors the behaviors of IoT nodes is needed to secure the IoT system and to prevent untrustworthy or undesired activities from compromised nodes. Therefore, IoT has a specific demand for service evaluation to ensure security due to the fact that it encourages the entire network to involve connected devices in participating in complex services. Preventing the negative effects caused by misbehaving nodes or malicious attacks on services is an essential task for IoT security. As discussed before, Trust was originally utilized in Social Sciences to measure the degree of interpersonal relationships. In the context of Computer Science, notably IoT, Trust becomes no longer limited to person-to-person but extends to objects (nodes), users, and communities because of the integration of devices and systems, i.e., hardware and software sides.

As can be seen in Fig. 1.3, Trust in IoT security involves various dimensions crucial for ensuring the reliability and integrity of interconnected devices and systems [18]. Inter-Individual trust pertains to the confidence established between individual entities within the IoT network, emphasizing the need for secure and trustworthy communication. Inter-Group trust extends this concept to relationships between different entities or groups of devices, fostering collaboration and dependability across diverse components. Group-Individual trust explores the dynamic between a singular device or entity and the larger IoT ecosystem, emphasizing the need for individual components to operate securely within the collective network. These types of trust are foundational in addressing security challenges, as trust is paramount for effective communication, data sharing, and collaborative decision-making within the IoT environment. Comprehensive understanding and implementation of these trust dimensions are essential for fortifying the security posture of IoT systems, ensuring that devices can interact and exchange information in a trustworthy and secure manner. Trust in IoT security, in its various forms, plays a pivotal role in establishing the foundation for resilient and secure interconnected systems.

The concept of Trust Management (TM) in Computer Science was first introduced in 1996 [19], where authors clarified that *"Trust management provides a unified approach for specifying and interpreting security policies, credentials, and relationships."* TM has emerged as one of the



(a) Inter-Individual Trust

(b) Group-Individual Trust

(c) Inter-Group Trust

Figure 1.3: Trust in IoT security

most promising ways to ensure the continued performance of a system that can be relied upon. The significance of TM has been evaluated in relation to IoT (Internet of Things) along a variety of dimensions, including reliable service management, access control, and strategy optimization. TM contributes to a reduction in the amount of uncertainty and risk regarding various IoT services [20, 21, 22].

Trust, which was first developed in the field of Social Science, has particularly developed into an essential instrument in IoT security. Employing trust in IoT aims to eliminate security concerns, assuring the dependable operation of IoT devices and enabling safe communication and cooperation. With the increasing prevalence of IoT devices and infrastructures in daily life, the importance of trust in terms of security is significantly growing.

1.2 Trust Management (TM) in Service-Oriented IoT (SO-IoT)

1.2.1 Trust Management Process

Before taking a closer look at TM models in IoT systems, it is necessary to identify the basic characteristics of Trust. Until the end of the section, we fix $T_{a,b}^{\mathrm{tn}}(x)$ to describe how much the trustor a trusts the trustee b at the moment tn with respect to trust object x.We illustrate the basic characteristics of trust in Fig. 1.4, in which a, b, and c are different nodes in IoT and are dedicated to being trustors or trustees accordingly. It is important to note that the inequality signs in the figure do not mean the trust cannot hold the same value mathematically but differs from each other in terms of trust nature.

- (i)**Independency**: the trustors hold their own subjective opinion towards the trustee, meaning that c may not trust as much as a trusts b.
- (ii)**Asymmetricity**: when the trustor and the trustee switch their roles in terms of trust assessment, how a trusts b is not based on how b trusts a, and vice versa.

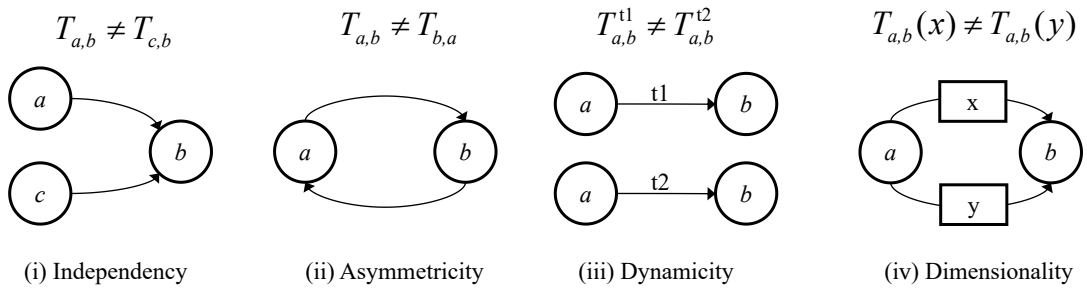


Figure 1.4: Four basic characteristics of Trust

- (iii)**Dynamicity**: trust between a and b can be updated by new experience; thus, the trust varies at different moments.

- (iv)**Dimensionality**: the trust from a to b may consist of several dimensions, i.e., not only one trust object. a may trusts b more on x issue rather than it does on y.

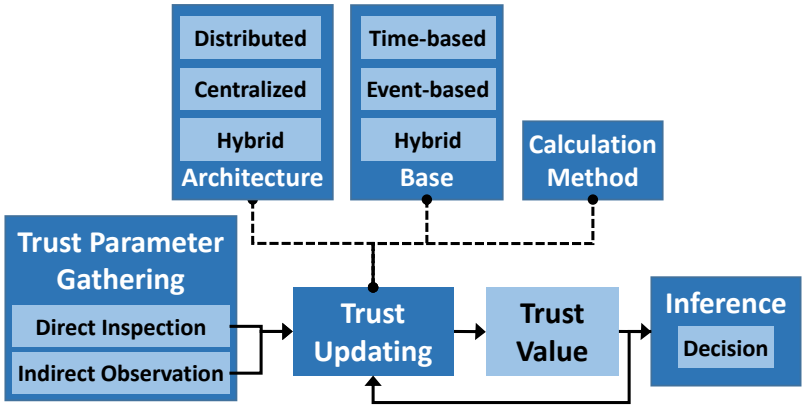


Figure 1.5: General TM process

As illustrated in Fig.1.5, the process of existing TM models has mainly three components: trust parameters gathering, trust updating, and trust inference. Firstly, the TM gathers relevant parameters through direct inspection (e.g., performance in their communication) and/or indirect observation (e.g., recommendations given by other nodes). The choice of trust parameters depends on concrete scenarios, e.g., nodes' nature and requirements of IoT services. Next, the degree of trustworthiness, namely the trust value, is calculated by a predetermined mathematical method in trust updating. The trust updating scheme also relies on two dimensions: (1) TM architecture, namely, distributed, centralized, and hybrid; (2) TM base, namely time-based, event-based, and hybrid. For TM architecture: the centralized model conducts TM by a single node, nodes are self-organizing in a distributed model, and the hybrid combines both. For TM base: the time-based model runs TM periodically; the event-based is activated when a specific event occurs, and again, the hybrid mixes both. After that, the trust value is computed, and TM infers a conclusion based on this value to decide if the node can be trusted. If the node is regarded as non-malicious, the inference result will be treated as its historical record to help update the next trust.

1.2.2 Service-Oriented Architecture (SOA)-based TM

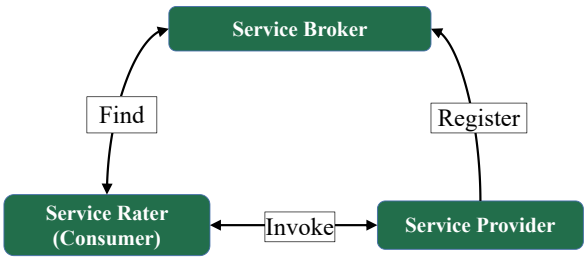


Figure 1.6: SOA

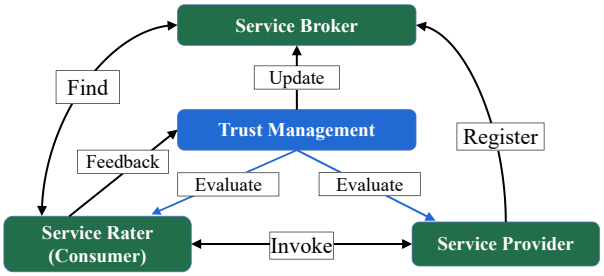


Figure 1.7: SOA-based TM

In service-oriented IoT [23, 24], intelligent IoT nodes can participate collaboratively in complex IoT services on the basis of the architecture illustrated in Fig.1.6. As can be seen, this architecture is composed of three fundamental elements: service broker, service consumer, and service provider (SP). The SP publishes its services in the repository of the service broker, and then the service consumer discovers and finally invokes the services. To summarize, three crucial activities in SOA are service registration, service composition, and service provision [25].

By integrating the TM into the SOA [26], Fig. 1.7 shows an SOA-based TM model for IoT: The service consumer becomes service rater (SR) when sending its feedback to the TM entity after service provision, and then both service rater and service provider will be evaluated by the TM. Next, TM can assist service brokers with decision-making through the results of the node classification and trust score values, e.g., checking available service types and removing malfunctioning nodes or malicious attackers. Finally, information related to services will be updated by the service broker. Indeed, the metrics are complex for service evaluation, e.g., a service provider performing poorly in the provision of service X may be outstanding at service Y. Thus, securing a service-based IoT requires the TM to interact with aforementioned service activities to assign to nodes and services the accurate trust values.

1.3 Scope of the Thesis

This thesis focuses on applying TM to improve IoT security and attempts to give answers to the following questions.

Q1: What are the threats, vulnerabilities, and limitations the current TM models are facing?

Q2: What are the challenges and benefits in the presence of the trust manager, and how can intra- and inter-community trust be assessed accordingly?

Q3: How can IoT devices be involved in a cooperative environment, how does the phase-based dynamic approach ensure trustworthiness in such a scenario, and what special difficulties are introduced by the implementation using real-world IoT devices?

Q4: Can nodes return to the original IoT community or switch to a new community, and how can trust assessment aid each community's access control to increase security since the entry of nodes?

Q5: Is it possible that IoT nodes assess trust in a self-organizing manner without a trust manager? If so, how to output optimal action according to the inferred trust level of the opponent node through a learning process? Can such a learning process encourage cooperation between the evaluator node and the opponent? Does it introduce new challenges to trust assessment?

Q6: Can trust be utilized in IoV security? What standardized protocols support V2X communication? What kinds of misbehavior in IoV can be considered from the perspective of trust?

To give the response to **Q1**, Section 1.5 reviews the related attacks, strategic malicious behaviors, and the current limitations due to TM architectures.

1.4 Contributions and Outlines of the Thesis

In **Chapter 2**, a role-based attack-resilient trust management (TM) model for community-driven IoT is proposed at two different levels to treat Group-Individual and Inter-Group Trust. First, the intra-community TM enables the IoT nodes within the same community (group) to be monitored dynamically based on their service roles, namely service provider (SP) and service rater (SR). Second, the inter-community TM examines the trust between different communities in terms of cooperativeness. The proposed model has been simulated and implemented under various attacks on service. The numerical results show the effectiveness in evaluating both intra- and inter-community trustworthiness, contributing thus to increasing the security and reliability of IoT. This chapter provides answers to **Q2**, **Q3**, and **Q4**.

After discussing the Group-related Trust, **Chapter 3** is going to focus on Inter-Individual Trust by proposing a Stochastic Bayesian Game (SBG) to address the Byzantine Altruistic Rational (BAR) based misbehavior, where workers' behavioral types can be deduced reasonably, and the requestor can perform optimal actions accordingly by taking the long-term gain into consideration. To validate and evaluate the performance of the proposed model, we simulate various scenarios and compare them with other approaches. The numerical results show the effectiveness and feasibility of our proposed solution. The answer to **Q5** is presented in this chapter.

As Chapters 2 and 3 both contribute on trust in SO-IoT, **Chapter 4** analyzes Trust into IoV, which focuses on evaluating the impact of V2X messages in terms of trust. By integrating CPS (Collective Perception Service) in the Veins simulator, we aim to develop a trust assessment model in IoV against several types of CAM- and CPM-based GV to increase security. The simulation results provide a preliminary analysis of the feasibility of the proposed model and show the effectiveness in terms of assessing V2X messages' trust. The answer to **Q6** is presented in this chapter.

And finally, we conclude this thesis in **Chapter 5** and present future work directions. The overall organization of this thesis is illustrated in Fig. 1.8.

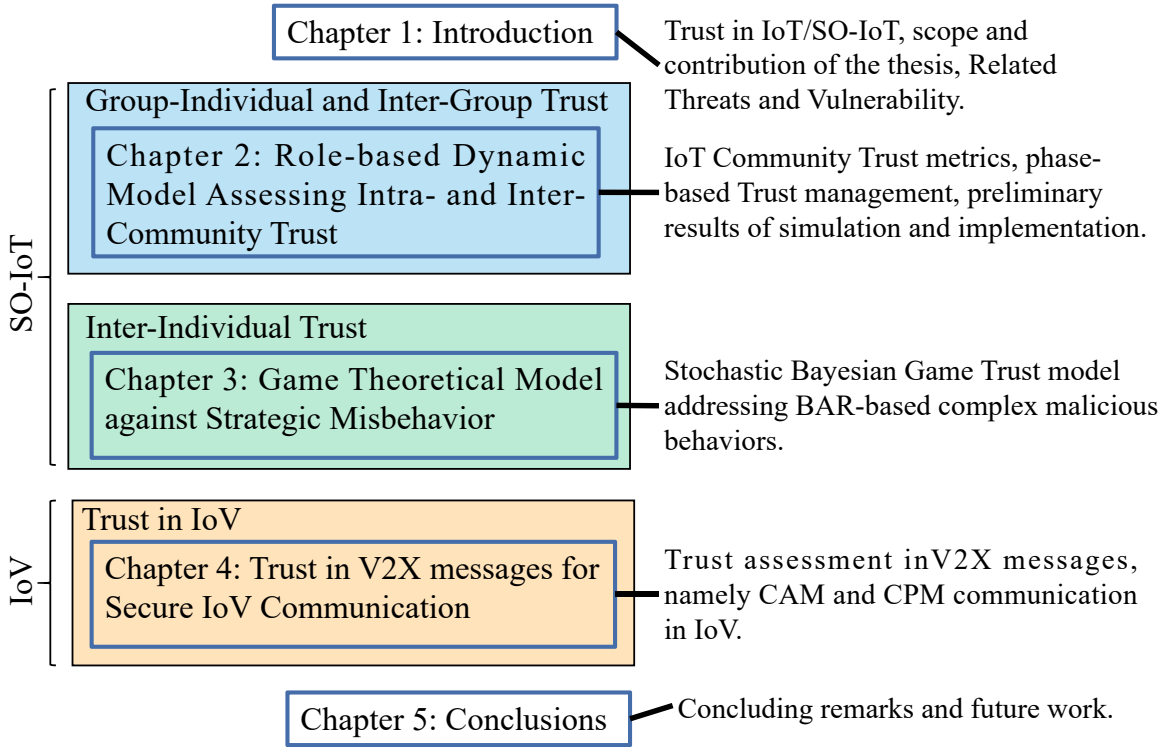


Figure 1.8: Organization of the thesis

1.5 Threats, Vulnerability

1.5.1 Review on Trust-Related Attacks (TRA)

Solutions involving intelligent IoT applications perform functions more efficiently and accurately, and intelligent technologies also benefit the constantly growing data generated from IoT network. SO-IoT not only improves existing IoT applications but also creates new smart services handling current or future challenges such as Communication-as-a-Service (CaaS) [27]. On the one hand, SO-IoT enables to build an intelligent multi-service environment, where smart devices within the network are encouraged to participate in service-related activities. On the other hand, the advancements combining IoT and intelligence also lead malicious nodes to become smarter in a way that execution strategies and offensive capabilities of malicious attacks can be enhanced by the use of intelligent technologies [28, 29]. More precisely, the intelligent attacker can act in an unnoticeable manner rather than roughly creating damages as false code injection or perturbing the communication as a black hole attack [30, 31].

The current TM models often consider a few types of attacks, but many other attacks are insufficiently addressed. Table 1.1 and Fig. 1.9 detail trust-related attacks in SO-IoT, which aim to mislead the trust evaluation by misbehaving on service or ratings.

Unfair Rating Attacks (URA) aim to disrupt systems by providing deceptive ratings, manifesting in three forms: **Ballot Stuffing Attack (BSA)**, where the assailant elevates sub-par service providers to enhance their reputation; **Bad Mouthing Attack (BMA)**, involving the tarnishing of the reputation of commendable service providers through negative ratings; and **Self-promoting Attack (SPA)**, where the attacker submits a false report to boost its own standing, seeking selection for service provision. These attacks contribute to disorder in the

Table 1.1: Categories of attacks on services

Src.	Attack		Target	Ref.
SR	URA	BSA	Rating	[32]
		BMA		[33]
		SPA	Service, Rating	
	NCA			
SP	IBA	CBA		[2]
		OOA		
		SBA		[34]

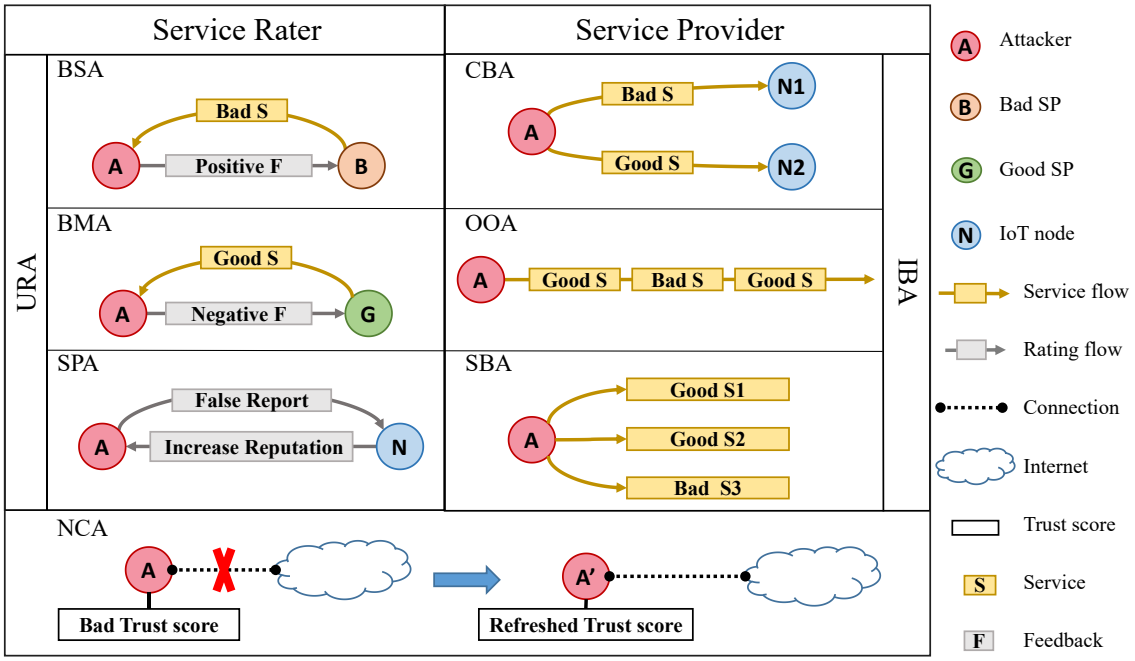


Figure 1.9: Categories of related attacks

rating ecosystem. Simultaneously, **Inconsistent Behavior Attacks (IBA)**, originating from malicious service providers, involve attackers altering their behavior to keep their trust score above a specified threshold. This category comprises three distinct types: **Conflicting Behavior Attack (CBA)**, where the attacker behaves differently with various nodes; **On-off Attack (OOA)**, characterized by alternating between good and bad behavior over time; and **Selective Behavior Attack (SBA)**, wherein the attacker alternates between competent and subpar performance across different services.

Newcomer Attack (NCA) aims to whitewash the attacker's trust score The attacker by re-entering the network with a new identity. The attack source can be both the service provider and the service rater when NCA occurs.

1.5.2 Strategic Malicious Intelligent Objects: BAR Behavioral Model

- **Altruistic**: performs actively and correctly by carrying out its dedicated task (terms equivalent to 'altruistic': 'honest', 'unselfish', and 'self-denying').

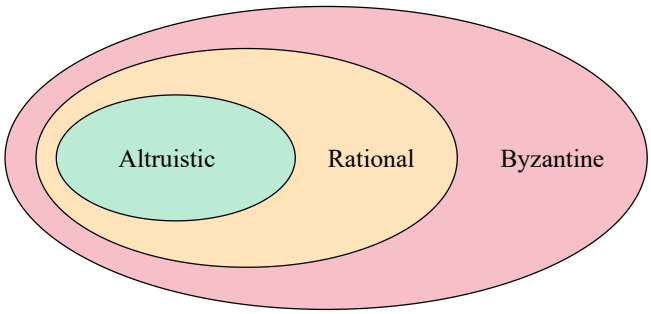


Figure 1.10: Euler diagram of behaviors in BAR-based threat model

- **Rational**: follows the specified crowdsourcing protocol in case the resource needed is sufficient. Otherwise, it may deviate from the suggested protocol (this type of node can also be referred to as 'greedy' or 'self-interested').
- **Byzantine**: performs intentionally disturbing and misleading the requestor, which causes the current Crowd-IoT system to be harmed (also called 'compromised', 'malicious', or 'adversary').

Fig. 1.10 gives the Euler diagrams of the possible behaviors according to the BAR-based threat model [35, 36]. The source of the concept of BAR-based threat model is Byzantine General Problem [37], where the Altruistic, Rational, and Byzantine behaviors are distinguished. It can be observed that the altruistic ones can only perform positive actions. Such a type of node is regarded as reliable and well-resourced. The rational can act not only altruistically but also selfishly. The reasons for rational behaviors are various, such as resource constraints that force the node to interact with others in a selective manner; or, the node, based on its own judgment of utility, desires to refuse cooperation rather than to engage in it. However, rational node cannot behave in a way that threatens the current IoT system, while the byzantine kind possesses the highest number of possible behaviors, which include all behaviors of the previous two types. To rephrase, a byzantine node can behave maliciously, altruistically, or rationally, depending on its purpose (i.e., being harmful, hiding its true motivation, and so on).

1.5.3 Systematic Limitations due to TM Architecture

Unfortunately, the current TM models present some considerable limitations within the SO-IoT context. The most important are listed in the following. First, as stated before, TM can support IoT applications, named trust-based IoT applications, such as intelligent clustering technique [38] and self-adaptive access control scheme [39]. However, these applications mainly aim to accomplish a single specific mission such that some trust parameters collected will not be used again after the trust updating. This leads to a TM that works only locally without offering a global opinion. Indeed, trust parameters collected in a single application or mission can be employed to help others. For instance, a node's software specification gathered in the access control can be adopted in the node selection when looking for a qualified service provider in terms of software. Hence, there is no reason that the TM is only implemented locally without considering the continuity and reuse of trust parameters.

Second, the interaction mechanisms between service-oriented activities and TM are not explored enough. As shown in Fig. 1.7, the service consumer becomes a service rater when sending

its feedback to TM after service provision, and then both the service rater and service provider will be evaluated by the TM. Next, TM can help service brokers by updating the results obtained.

The current TM models mainly utilize a predefined and static approach to assign nodes service provision missions, i.e., no consideration of differentiating service by type and nodes in terms of the role and capability. For instance, authors in [40] developed an AI-based TM model to accelerate decision-making for service provision but this model does not fit SOA-based IoT due to insufficient discussion for service registration and composition. As smart services in SO-IoT systems are composable and multidimensional, this limitation makes service-oriented activities, such as service composition and evaluation, unreliable. Furthermore, because nodes not only provide service but consume and rate services, the trust metrics in SO-IoT show an additional complexity compared with classic IoT. For example, an IoT service provider performing service type X unsatisfactorily may be excellent in performing another service type Y. Similarly, a node being outstanding in service provisioning may be very bad in service rating. Hence, TM models applied in general scenarios cannot conform to other scenarios of service-oriented environments like SO-IoT [41].

1.6 Conclusion

The existing TM models in the context of SO-IoT exhibit notable limitations. Firstly, while these models support trust-based IoT applications, they are often designed for specific missions, resulting in a lack of global applicability. Trust parameters collected during a mission are not systematically reused, hindering the creation of a comprehensive global opinion. For instance, data gathered for access control may not be utilized when selecting a service provider. This localized approach limits the continuity and adaptability of TM across various applications. Secondly, the interaction mechanisms between service-oriented activities and TM are insufficiently explored. The dynamic relationship between service consumers, raters, providers, and brokers is not thoroughly addressed, leading to a gap in leveraging TM for comprehensive evaluations and updates. Lastly, existing TM models employ predefined and static approaches for assigning nodes service provision missions, lacking differentiation based on service type, node role, and capability. This limitation compromises the reliability of service-oriented activities like composition and evaluation in SO-IoT systems. The multidimensional nature of smart services in SO-IoT adds complexity, as nodes both provide and consume services, introducing nuances not adequately addressed by conventional TM models. In conclusion, these limitations underscore the need for more adaptable, globally applicable, and dynamically interactive TM models tailored to the intricate service-oriented landscape of SO-IoT. In the following chapter, we will first study Group-Individual and Inter-Group trust assessment through a role-based dynamic model.

Group-Individual and Inter-Group Trust in SO-IoT

Chapter 2

A Role-based Dynamic Model Assessing Intra- and Inter-Community Trust

2.1 Introduction

As discussed before, to overcome scalability and resource allocation limitations, IoT systems are often formed by groups/communities with respect to their own interests or functionalities. Within this context, evaluating the trust of each individual within the group is needed to maintain the intra-group trust level, and equally importantly, the trust between groups should also be assessed to prevent negative effects caused by dysfunctional or malicious groups/communities. However, in the literature, the current studies mainly focus on handling Group-Individual Trust. Kowshalya *et al.* in [42] presented a service score mechanism to help with service evaluation between IoT devices, and the TM model in [34] proposed to measure the asymmetry between capabilities and service types before service provision is slightly beneficial for service discovery. However, their TM architectures, either fully centralized or fully distributed, remain challenging for IoT systems. Authors in [43] designed a clustering TM architecture by grouping IoT nodes into a community on the basis of interest and relationships between nodes, where a leader should be elected to manage the trustworthiness within the community. The memory storage issue is improved, and OOA is addressed in this work, but the lack of countermeasures against BMA and BSA makes the TM model vulnerable since it proposed a leader selection scheme. Alshehri *et al.* in [44] introduced an intelligent hierarchical approach to create a hybrid TM environment, where the architecture consists of the Master Node that manages cluster nodes and the Super Node that handles the allocation of the cluster for MN. This model proposed an algorithm to eliminate ratings outliers. However, the dishonest rater detection and isolation mechanism are missing, and the accuracy of trustworthiness evaluation will be reduced when honest ratings are not counted. Moreover, the attacks from the service provider side are not considered.

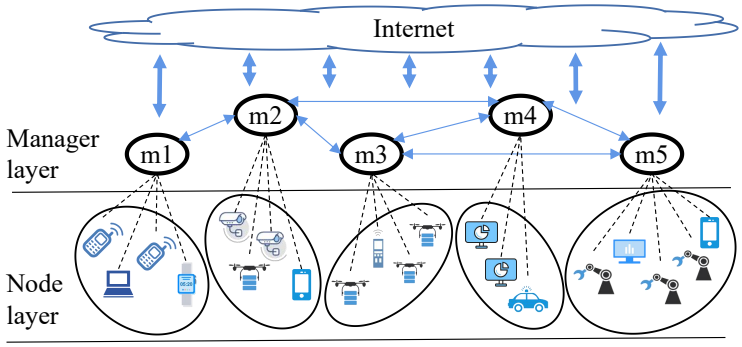


Figure 2.1: Illustration of community-based IoT systems composed of intra-community nodes and managers per each community

On the one hand, most of the aforementioned works addressed a few attacks on services such as OOA, BMA, and BSA. On the other hand, these models do not fit the SOA due to the lack of discussion of service activities in IoT. More importantly, some of them are either fully distributed or fully centralized, which puts their suitability in question, i.e., Inter-Group Trust is missing. To overcome these limitations, we design a hybrid architecture to support and improve the Group-Individual and Inter-Group Trust in IoT. Fig. 2.1 demonstrates the proposed architecture of the proposed trust framework with two levels: intra- and inter-community TM, representing Group-Individual and Inter-Group Trust, respectively. Smart devices are assembled at the intra-community level, where nodes in the same community can participate in cooperative missions to interact in a multi-service environment. Each trust manager is in charge of intra-community TM as a local responsible entity. Moreover, managers are networked so that the communication and

TM at the inter-community level are distributed. Notably, each community manager's access control (AC) policy is not identical since different communities may have diverse preferences in terms of service. Hence, newcomer nodes will be examined much more strictly in those communities with specified demands. In this chapter, Section 2.2 explains intra-community trust evaluation, and then 2.3 presents the inter-community trust evaluation. The simulation results and performance analysis are presented in Section 2.4. Section 2.5 details the implementation realized and the preliminary results obtained. Section 2.6 draws the conclusion and outlines our future work.

2.2 Intra-Community Trust (Group-Individual Trust)

2.2.1 Overview of Four Phases

From the perspective of SO-IoT, a node within the community can provide service and consume service by performing SP or SR. In this sense, the manager should assess intra-community nodes since their entry, monitor their behaviors as SP and SR, and output a global opinion to determine their trustworthiness. This trust evaluation process enables the manager to measure the intra-community trust, i.e., Group-Individual trust. Fig. 2.2 illustrates the overview of the intra-community trust scheme consisting of four phases: access control, service provider selection, service evaluation, and node classification. Initially, node identification allows nodes' attributes to be treated in the access control phase in order to decide if their entry into the current community can be authorized. Once service is requested in the community, the SP selection phase ranks available SPs. After the service is given, the feedback from service consumers will be collected to support the service evaluation phase, and thus, service consumers become so-called service raters (SR). Eventually, based on the trust evaluation results, nodes' trustworthiness as SP or SR will be classified to determine if they are malicious or trustful. The four-phase model will be detailed in subsections 2.2.2-2.2.5, and in particular, different AC cases will be discussed in subsection 2.2.6.

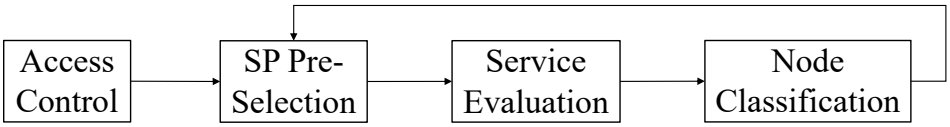


Figure 2.2: Four-phase intra-community trust assessment

2.2.2 Phase 1: Access Control

-Attribute-based Access Control

Since recording the node by its asserted attributes is more advantageous in terms of security [45], we employ the ABAC (Attribute-Based Access Control) for the AC phase, which is regarded as a logical methodology for AC in IoT [46, 47]. For defining permissions in AC phase, we are concerned with the three types of attributes: function (fct), social (soc), and context (ctx). Fix a node i until the end of the section. The set of attributes of node i is denoted as $A_i = \langle A_i^{\text{fct}} |\ A_i^{\text{soc}} |\ A_i^{\text{ctx}} \rangle$.

- **Function attribute** (A^{fct}) requires services that the comer node is able to perform, meaning that $A_i^{\text{fct}} = \langle \{s, s \in S_i\} \rangle$. By validating nodes' capabilities concerning IoT services, the

services S^{fct} that are needed in the current community will be categorized as S^{fct}, and thus $S_i^{\text{fct}} \subseteq S_i$.

- **Social attribute** (A^{soc}): Although social features are widely studied in the Social Internet of Things (SIoT), the IoT nodes also have such features since they interact with each other, share data, and collaborate for service provision [48]. In our model, we consider three main object relationships: parental (PR) and co-work (CWR). Nodes belonging to the same manufacturer have higher PR, as their characteristics in terms of software specification are somehow approximated. Moreover, we measure the similarity of functional services of nodes because the CWR value increases when nodes have more opportunities to cooperate in service.

- **Context attribute** (A^{ctx}) describes the relevant contextual information that can be used as security characteristics [49]. In the proposed model, the context attribute contains a hash value that enables verifying if the comer node is a newcomer.

-Default Score (DS) Initialization

- DS by using function attribute is denoted for node i by DS_i^{fct}, and defined by:

$$DS_i^{\text{fct}} = \begin{cases} 1, & \text{if } S_i^{\text{fct}} \neq \emptyset \\ \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

- DS by using social attribute DS_i^{soc} is defined by:

$$DS_i^{\text{soc}} = \mu^{PR} PR_i + \mu^{CWR} CWR_i, \tag{2.2}$$

with $\mu^{PR} + \mu^{CWR} = 1$,

$$PR_i = \frac{1}{|CN|} \sum_{k \in CN} v_{ik}^{PR}, \tag{2.3}$$

$$CWR_i = \frac{1}{|CN|} \sum_{k \in CN} \frac{|S_i^{fct} \cap S_k^{fct}|}{|S_i^{fct} \cup S_k^{fct}|}, \tag{2.4}$$

where for all k, v_{ik}^{PR} is an indicator describing if i and k belong to the same production batch.

- DS by using context attribute DS_i^{ctx} is defined by:

$$DS_i^{\text{ctx}} = \begin{cases} 0.5, & \text{if } i \text{ is newcomer} \\ \\ 1, & \text{otherwise} \end{cases} \tag{2.5}$$

With three sub-DS values calculated by (2.1-2.5), the DS of node i is defined as:

$$DS_i = (\omega^{\text{fct}} \cdot DS_i^{\text{fct}} + \omega^{\text{soc}} \cdot DS_i^{\text{soc}})^{1/DS_i^{\text{ctx}}}, \tag{2.6}$$

where $\omega^{\text{fct}} + \omega^{\text{soc}} = 1$. Newcomer node is permitted to enter if its DS>0.5. In IoT environments, nodes frequently participate in cooperative or collective services, which means they are SR and SP at the same time, e.g., CABS (Cooperative Awareness Basic Service) and CPS (Collective Perception Service) in IoV [3], where vehicles simultaneously perform and benefit from these services. From this perspective, rating services given by others and being assessed by others are equally important.

2.2.3 Phase 2: Service Provider Selection

-Service Composition

Upon receiving a service or a mission request, denoted as S_{req}, the SP selection phase will search for qualified SP to perform service while some missions require a workflow composed of numerous services [50], which often occurs in industrial context [51]. On the other hand, taking the two above-mentioned services in IoV as an example, since they are safety-related, and thus, such services are somehow always in need. In this case, the community manager will not conduct SP selection but eliminate disqualified SP to ensure trust in service. Fig.2.3 illustrates commonly-used service composition constructs, including sequence → (s1, then s2), loop * (s1 several times), flow ⊕ (s1 and s2), and switch ⊗ (s1 or s2).

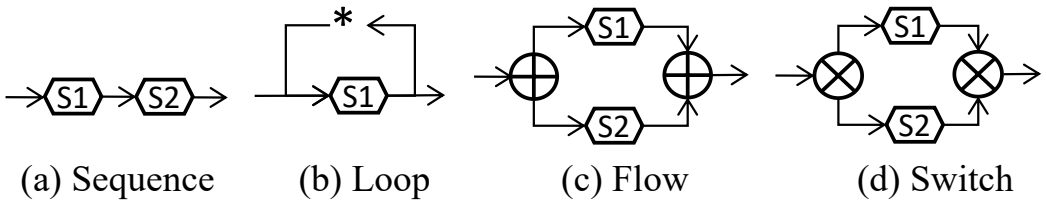


Figure 2.3: Constructs for workflow

-Service Grade (SG) and Service Score (SS) Calculation

Consider so-called Service Grade SG_i^n representing the quality of the node i's service of the type n, (2.7) gives the calculation of OSG in case of workflow, we fix r for loop * representing the times of repetitions.

$$OSG_i = \begin{cases} \prod_{n \in S_{\text{req}}} SG_i^n, & \text{for } \rightarrow \\ (SG_i^n)^r, & \text{for } * \\ 1 - \prod_{n \in S_{\text{req}}} (1 - SG_i^n), & \text{for } \oplus \\ max_{n \in S_{\text{req}}}(SG_i^n), & \text{for } \otimes \\ SG_i^n, & \text{if } n{=}S_{req} \text{ (single service)} \end{cases} \tag{2.7}$$

Next, the community manager will generate a ranking of selection scores (SS_i), calculated as follows:

$$SS_i = \begin{cases} QSP_i \cdot OSG_i, & \text{for a workflow} \\ \\ DS_i, & \text{if newcomer.} \end{cases} \tag{2.8}$$

The examination of the candidate SP in (2.8) concerns two sides: QSP gives the opinion from a more general view, e.g., the stability (refer to Section 2.2.4, where we explicate how QSP is evaluated); and SG/OSG measures node's competence with regard to services or workflow required. Any node with poor QSP or SG/OSG will obtain a low value for SS. Due to the fact that the community manager selects only the best-ranked candidates with significant SS, those of low rank barely have the opportunity to assist in service provision.

2.2.4 Phase 3: Trust Computation

As authors analyzed in [52], the trustworthiness measurement of interactions between IoT nodes remains challenging when the service badly performs or the service is poorly evaluated. Therefore, in the service evaluation phase, the trustworthiness of nodes will be measured by four metrics, namely TS, QSR, QSP, and SG. TS gives an overall opinion on the basis of QSP and QSR, and SG corresponds directly to the quality of each service type. The feedback originated from the service consumer (so-called SR in our model) j to rate the service quality of the SP i is in the range of $[0, 1[$ (0 means no service conducted from service provider) and denoted as f_{ji}. For this, we consider a rating process performed by node, e.g., the mission success rate or network performance in [53], and service evaluation in [3].

-Trust score (TS)

We set:

$$TS_i = \frac{|RS_i|}{|RS_i + PS_i|} \cdot QSR_i + \frac{|PS_i|}{|RS_i + PS_i|} \cdot QSP_i, \tag{2.9}$$

where RS_i and PS_i denote the services rated and provided by the node i, respectively, and QSR_i and QSP_i are given by (2.10) and (2.12). It can be seen that the global opinion concerning the trust of a node depends on the trust of this node's two roles: SR and SP. This means that the evaluation of a node depends on the behavior under both roles, e.g., a reputable SP may be dishonest when rating others' services; likewise, an excellent SR may be terrible at service provision. That is also the reason that we deploy the quantity parameter, i.e., node's workload respectively on SR and SP sides, to weight (2.9). In such a way, the quality and quantity impact of QSR and QSP can be carried out accurately to assess the trustworthiness of the node as both SR and SP.

-Quality of service rater (QSR)

We set:

$$QSR_i = \varphi \cdot CQSR_i + (1 - \varphi) \cdot LQSR_i \ , \tag{2.10}$$

where $\varphi \in [0.5, 1)$, $CQSR$ and $LQSR$ denote respectively the current and last values of QSR.

We put QSR=$CQSR$ for all newcomers, since they do not possess any rating records upon arrival. The $CQSR$ in (2.10) is computed as follows:

$$CQSR_i = 1 - \frac{1}{|R_{i-}|} \sum_{j \in R_{i-}} |f_{ij} - \bar{f}_j|^{1/l} \ , \tag{2.11}$$

where R_{i-} represents the set of nodes that are rated by i in the service evaluation phase, $\bar{f}_j$ is the average value of feedbacks evaluating node j, and l is a punishment degree, such that dishonesty is amplified by the exponent $1/l$. Notably, the $CQSR$ value will be assigned zero in case the service rating is missing.

Indeed, the calculation of QSR is based on the comparison between the opinions of the rater node and the average value of other raters, which enables us to distinguish the dishonest service raters by identifying the gap in this comparison. In IoT, the feedback f_{ij} from i to evaluate j can emerge by a predefined measurement scheme that is objective thus it will not have a large variance, as in SO-IoT, due to the user preference or environmental perturbation. Therefore, an unfair rating from a dishonest rater that either ruins a well-behaved node's reputation (e.g., BMA) or boosts a misbehaved node's reputation (e.g., BSA), can be detected.

-Quality of service provider (QSP)

Similar to (2.10), we consider both the last and the current measurement of the rater to determine its QSR:

$$QSP_i = \varepsilon \cdot CQSP_i + (1 - \varepsilon) \cdot LQSP_i \ , \tag{2.12}$$

where ε is set in $[0.5, 1)$ to weight the current value ($CQSP$) and the last value ($LQSP$). We put $LQSP$=DS for newcomers, which is reasonable since we set DS to QSP for newcomer nodes in the AC phase. The $CQSP$ in (2.12) is computed as follows:

$$CQSP_i = \frac{1}{|R_{-i}|} \sum_{j \in R_{-i}} \theta_{ji} \cdot \lambda_{ji} \cdot QSR_j \cdot f_{ji} \ , \tag{2.13}$$

where R_{-i} represents the set of nodes that rated services from i, θ and λ are stability parameters against OOA and CBA, respectively given by (2.14) and (2.15):

for all $j \in R_{-i}$,

$$\theta_{ji} = \mathrm{sinc}(1 - f_{ji}) \cdot \mathrm{sinc}(\Delta f_{ji})^{\Delta t_{ji}}, \tag{2.14}$$

$$\lambda_{ji} = 1 - |f_{ji} - \bar{f}_i|^{1/l}, \tag{2.15}$$

where Δt_{ji} and Δf_{ji} are time gap and difference of last feedback (lf_{ji}) and present feedback (cf_{ji}), i.e., $\Delta t_{ji} = t_{cf_{ji}} - t_{lf_{ji}}$ and $\Delta f_{ji} = |cf_{ji} - lf_{ji}|$ (or 0 for newcomers). The (normalized) sinc function is defined as:

$$\mathrm{sinc}(x) = \begin{cases} 1, & \text{for } x = 0 \\ \dfrac{\sin(\pi x)}{\pi x}, & \text{for } x \neq 0, \end{cases} \tag{2.16}$$

and is chosen because it is continuous at point 0, maps $[0, 1]$ onto $[0, 1]$, and has inflections that can be used to penalize the large Δf_{ji} and poor f_{ji}.

The unstable behaviors over time are penalized by use of θ_{ji}, since it is increasing in Δf_{ji}, with an exponent Δt that renders unacceptable any drastic changes in service quality. In (2.15), $\bar{f}_i$ is the average value of i's notes rated by other rater nodes and l is the punishment degree, as in (2.11). In other words, conflicting behavior will be captured due to the parameter λ, which compares the service quality of each individual to the average level. By the very definitions of the coefficients θ and λ, the unique possibility for the node to gain reputation is to keep steadily providing satisfying services.

-Service grade (SG)

Since malicious nodes may perform well and badly between service types in an alternative manner, a dedicated trust score to precise the SP's performance in terms of service type is necessary. To evaluate the service quality of type n, the service grade SG_i^n is computed as follows:

$$SG_i^n = \kappa \cdot CSG_i^n + (1 - \kappa) \cdot LSG_i^n, \tag{2.17}$$

where $\kappa \in [0.5, 1)$ weights the current value (CSG) and last value (LSG), and

$$CSG_i^n = \frac{1}{|R_{-i}^n|} \sum_{j \in R_{-i}^n} QSR_j \cdot f_{ji}^n, \tag{2.18}$$

where R_{-i}^n is the set of nodes that rated the service of type n provided by i and f_{ji}^n denotes the feedback from j for the service of type n provided by i. Notably, $SG^n = CSG^n$ for newcomer nodes. SG is used to observe specifically the service quality of each type that is marked as 'functional' since the AC phase, i.e., if any single type of S^{fct} gets a low value of SG, it will be regarded 'nonfunctional' service and cannot provide such service type anymore. Accordingly, this service type will be removed from S^{fct}.

As a result, the SG^n will decrease if a node persists in providing unsatisfying service on a particular type n. After that SG^n<0.5, the community manager must label this service type as malfunctioning and immediately remove it from S^{fct}. After the removal, in order to prohibit the node from being selected as SP for the service type n. Hence, the misbehavior aiming at service types from malicious SP, namely SBA, can only provide fewer and fewer service types due to the SG mechanism. Finally, two situations may occur: either it performs well for the other service types to stay in the current community, or it progressively loses its competitiveness in the SP selection and will eventually be eliminated from the community.

2.2.5 Phase 4: Node Classification

As stated in section 1.5.3, a node underperforming service provision may outperform service rating, and thus securing a service-based IoT requires observation for both SP and SR sides. More importantly, the attack on service is divided into two categories by attack source, namely SP and SR. The node classification should take into consideration a scheme to enable the identification of the source of the attack. For this, by classifying the values of TS, QSP, and QSR under good (>0.5) and bad (≤0.5), the node classification scheme illustrated in Fig. 2.4 enables the community manager to categorize nodes into 6 groups:

Figure 2.4: Node classification scheme

- Good Node (GN): will surely stay in the current community since its TS, QSP, and QSR are all at a good level.
- Weak Service Rater (WSR): will be banned from requesting services as the QSR is ineligible for rating of the service.
- Weak Service Provider (WSP): Different from the treatment of WSR, a WSP node is not deprived of anything. However, it has been categorized into WSP because of its low QSP, thus, it has little chance of being picked since its QSP induces incompetence.
- Bad Service Rater (BSR)/Unfair Rating Attacker (URA): It is difficult to determine precisely if this node is just incapable or malicious, but in any of the two cases, the community manager must eliminate the node in order to minimize the adverse effects of erroneous ratings from the community manager's view.
- Bad Service Provider (BSP)/Inconsistent Behavior Attacker (IBA): Analogously, SP belonging to this group may be simply unreliable in terms of service quality or maybe an attacker who misbehaves. In any of the two cases, the community manager must eliminate the node.
- Malicious Node (MN)/Mixed Type Attacker (MTA): It is the worst case among the node classification as all three metrics consisting TS, QSP, and QSR are bad. The community manager must remove a node belonging to this group immediately.

In fact, the reason why WSP and WSR nodes are not isolated from the network is because their TS values remain good, i.e., they still have some valuable aspects that can benefit the current community from a global perspective. Some works suggest using Machine Learning (ML) technologies to support the decision-making of trust management, e.g., [40] and [54]. On the one hand, ML brings the possibility of improving the classification scheme. On the other hand, the implementation of ML may cause higher costs, and more importantly, due to the dynamic nature of IoT systems, there is no guarantee of possessing enough data for training, and thus, the accuracy may be perturbed. For these reasons, as our rule-based classification scheme remains imperfect, the utilization of ML is not necessary.

2.2.6 Re-entry to the same community or move to a new community

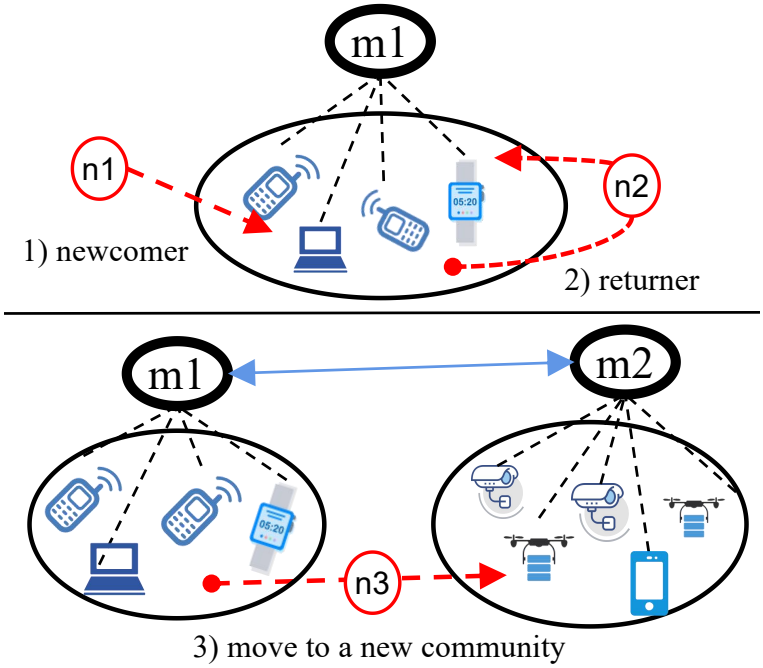


Figure 2.5: Three AC cases and related checking mechanisms

It is worth noting that, in IoT, there are other possibilities besides the newcomer case. For instance, a device whose power source is rechargeable will be disconnected for recharging and then reconnected to the network. In order to keep the reputation of such node consistent throughout the recharging operation and prevent an undesirable node from whitewashing its reputation (namely NCA), this type of node should be treated as a returner rather than as a newcomer. Fig.2.5 demonstrates three cases, namely newcomer, returner, and the node moving to the new community. Inspired by work in [1], the checking mechanism is as follows: the newcomer node n1's attributes will be collected and assessed to decide if its entry can be authorized. n2's trust is controlled by m1, m1 stores n2's information as $info_{n2}$, then generates a key converted by one-way hash function h_{n2}=Hash($info_{n2}$). When n2 returns, m1 verifies this key. The comer node that gives an incorrect key will be viewed as a malicious one and cannot enter the community, and this key will not be required for newcomer ones. Once n2's entry is allowed, m1 employs n2's previous trust scores for the following evaluation. n3 is going to enter the community controlled by m2 from the community controlled by m1. In this case, m2 will request m1 the $info_{n3}$ and verify the key sent by n3 for the context attribute. m2 will consider n3's previous QSR for the following assessment. The reason that the QSP and SG cannot be utilized again in the new community is that the service environment differs from the original community. The checking mechanisms are necessary in terms of security, this is because an NCA attacker attempts to re-enter the original community or move to a new community to obtain a refreshed trust score. On the other hand, such checking mechanisms enable the well-behaved nodes to maintain their reputation in case of returning to the original community or moving to a new community. Fig. 2.8 shows how the community manager employs differently in different AC cases.

2.3 Inter-Community Trust

2.3.1 Overview of Three Phases

After intra-community trust, we will take a closer look at inter-community trust where managers of each community are responsible for exchanging information and conducting trust assessments on the other communities. Unlike the intra-community TM, there is no service provision or rating at the inter-community level but the cooperation of service migration (node moving to a new community case). For example, in the production line given in [55], different industrial factories are dedicated for specific production missions to optimize the supply chain and achieve a common task in the production line. Therefore, the main objective of the inter-community TM is to identify 'unfriendly' communities that may endanger the current community security by sending malicious nodes. Similar to [56], the community manager is in charge of the inter-community trust, and we design a simple three-phase mechanism to monitor the inter-community trust. The interaction between the evaluated and evaluator communities is illustrated in Fig.2.6. The current community manager evaluates the other community in the initial evaluation phase when 'they do not know each other', and this case often occurs in IoV because of vehicles with high mobility. In the cooperation evaluation phase, the evaluator observes the cooperativeness of the other community by analyzing the behaviors of the nodes coming from the evaluated community. Finally, the community classification phase determines evaluated community categories.

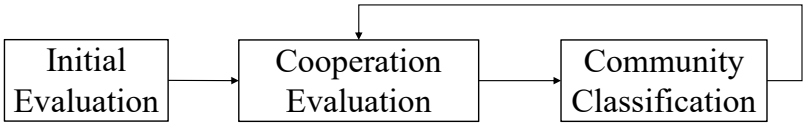


Figure 2.6: Three-phase inter-community trust assessment

2.3.2 Phase 1: Initial Evaluation

In this phase, the evaluator community will assess the closeness of the opponent community in terms of service. First, if two communities hold more common services, nodes moving from one to another will be more likely to be accepted. Second, the evaluator community cannot benefit from the opponent one if their service environments largely differ, especially for some safety-related services such as the CPS in IoV that we mentioned before. For example, while a vehicle performing CPS/CABS moves to the evaluator community, the safety-related information gathered by its CPS will be significantly useful in increasing road security. It should be noted that the evaluated community will assess the evaluator community at the same moment to initialize the inter-community trustworthiness as they were not aware of each other before. The calculation of the initial score IS is defined as:

$$IS_{mp} = \frac{|S_m \cap S_p|}{|S_m|}, \tag{2.19}$$

where S_m and S_p denote respectively the sets of all functional services needed in communities m and p. The IS examines the similarity of two communities in terms of community interest, and in such a way, they become complementary if this value is considerable. For instance in the industrial context, given p and m have high IS, if a sudden failure of an essential service type comes to the factory p, it can immediately ask the other factory m to help by sending nodes providing this service.

2.3.3 Phase 2: Cooperation Evaluation

To deal with the above-mentioned issue in the industrial context or due to their own need (e.g., disconnect to recharge the battery), nodes may malicious nodes may misbehave in the AC phase or service provision phase to mislead the new community. By implementing the intra-community trust assessment discussed before, the malicious nodes can be detected and removed, but the conclusion remains at the intra-community level, i.e., no evidence to confirm the role of the source community, the node-sender community, especially if it has been compromised. For this reason, the cooperation evaluation should take into consideration the observation of nodes coming from other communities to determine their nature in terms of security.

As stated already, the nodes that move to a new community may behave badly in the AC phase or service provision, therefore, we measure the number of good nodes out of all those that moved to the new community to compute the cooperativeness value (CO):

$$CO_{mp} = \frac{|GN_{pm}|}{|MN_{pm}|}, \tag{2.20}$$

where GN_{pm} represents the good nodes (evaluated by the current community m according to the node classification scheme given in Fig. 2.4) from p to m, and MN_{pm} are all the nodes that moved from p to the new community m.

The cooperation score (CS) of p evaluated by m can be computed in an iterative way as follows:

$$CS_{mp} = \begin{cases} IS_{mp}, & \text{before any interactions} \\ \\ \eta^{CO} \cdot CO_{mp} + \eta^{IS} \cdot LCS_{mp}, & \text{otherwise,} \end{cases} \tag{2.21}$$

where LCS_{mp} represents the last CS_{mp} value, $\eta^{IS} + \eta^{CO} = 1$. $LCS_{mp} = IS_{mp}$ for the first evaluation. IS_{mp} given in the initial evaluation phase is somehow regarded as a threshold since two close communities should cooperate more, but a gap may emerge between the threshold and the reality that a number of malicious nodes come from a community with great IS. Thus, we should also consider the current cooperativeness of the evaluated community, i.e., the value of CO.

2.3.4 Phase 3: Community Classification

With the CS value, Fig 2.7 classifies the evaluated community into two groups: convenient community (C-com) and distant community (D-com).

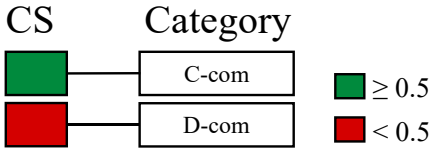


Figure 2.7: Community classification scheme

Consequently, the evaluator community will reduce the communication frequency with the evaluated communities whose CS value is low since their interactions are considered valueless. On the other hand, the evaluated communities with great CS value have higher priority for

the evaluator community when looking for support since the evaluated one is more profitable than others. It should also be noted that the community classification is meaningless when two communities interact insufficiently, e.g., an evaluated community with a very low IS score cannot give any remark on its cooperativeness.

2.4 Simulation Results

As we focus on intra- and inter-community trust evaluation, it is assumed that the network and communication between nodes are reliable. By using the MATLAB platform, we constructed an analytical framework to simulate trust evaluation within a single community and between different communities to verify the effectiveness of intra- and inter-community TM of the proposed framework. Parameter settings are given in the next subsection, the simulated scenarios of intra- and inter-community will be detailed in subsections 2.4.3 and 2.4.3, respectively. In intra-community trust, we will analyze trust values over four phases, particularly, we will evaluate trust values in the presence of the mentioned trust attacks to validate the countermeasures. In inter-community trust, we will focus on changes in trust values of different community types over three phases.

2.4.1 Parameter Settings

Table 2.1: Simulation parameters values

Parameter	Value		Parameter	Value
ω^{fct}, ω^{soc}	0.5		η^{CO}	0.2
μ^{PR}, μ^{CWR}			η^{IS}	0.8
φ, ε, κ			l	2

As illustrated in Table 2.1, we consider DS^{fct} and DS^{soc} identically critical for the AC phase, and as we have that $\omega^{\text{fct}} + \omega^{\text{soc}} = 1$, consequently we set ω^{fct} and ω^{soc} 0.5. Likewise for μ^{PR} and μ^{CWR}. For η^{CO}, we set 0.8 since the real-time evaluation is considered significant to demonstrate the cooperativeness between communities. Respecting the constraints of $\eta^{IS}+\eta^{CO} = 1$, we assign η^{CO} 0.2. ε, φ, and κ are given 0.5 for the reason that the last and the current evaluations are equally important. Finally, the punishment degree l is set to 2. Other simulation configurations concerning intra- and inter-community TM are given in Sections 2.4.2 and 2.4.3, respectively.

2.4.2 Intra-Community Trust

-Configuration of Intra-Community Trust

Our approach concerning intra-community consists of three parts: 1) The first part concerns the access control phase, explicating the DS calculation and different AC cases; 2) The Second part gives a concrete example of SS computation and conducts a comparison showing the proposed SP selection scheme is advantageous regarding the service composition; 3) The final part validates the resilience of the proposed intra-community TM model against various attacks in service, namely OOA, CBA, SBA, BMA, and BSA. Table 2.2 illustrates the simulation configuration of the intra-community TM, and the service types are numbered to simplify the representations. In

the current community, there are 16 nodes belonging to 4 types (4 per each type), and these nodes are considered trusted by giving a random value in the range of [0.9 0.95] for their trust values. For the service composition, we randomly select 1 out of 5 possibilities (single, $\rightarrow$, *, $\oplus$, and $\otimes$), where $\rightarrow$, $\oplus$, and $\otimes$ contain 2 service types each (excluding $s4$ since understaffed). In addition, r=2 for * case. Finally, only 1/2 of candidate SP can participate in the final service provision ((candidate SP+1)/2 if odd), and service consumers are all nodes that are not candidates to involve more SR.

Table 2.2: Configuration of intra-community TM simulation

Conf.	Description
Manager	Single one
Service types	s1∼s4
Node type	see in Table 2.3
Population of nodes	(t1∼4)×4=16; (NC)×2; (Re)×2

NC = Newcomer, Re = Returner

-Trust Initialization

Table 2.3: Nodes setting concerning AC phase

Type	S^{fct}	PB		Type	S^{fct}	PB
-	1	a		NC1	1,2,3	a
	1,2	b		NC2	1,4	e
	1,2,3	c		Re1	1,2,3	a
	1,3	d		Re2	1,4	e

PB = Production Batch

As defined in Table 2.3, NC1 and Re1 possess the same functional services and PB, likewise for NC2 and Re2. Table 2.4 illustrates the DS calculation of comer nodes, namely NC1, NC2, Re1, and Re2. Note that the nodes in the current community remain unchanged for each DS calculation. DS^{fct} will be 0 for nodes that cannot provide S^{fct}, meaning their entries are basically impossible since the threshold of the AC phase is set to 0.5. For this reason, such 'incapable' type is not considered in the AC phase evaluation. We can observe in the table that NC1 and Re1 receive significant DS^{soc} as nodes of the 'a' type PB already exist in the community, and thus their social relationships are considered stronger than two other types of comer nodes. We can also notice that returner nodes Re1 and Re2 are assigned higher DS^{ctx}, which reflects their DS values higher than newcomer nodes. On the other hand, the NC2 node is refused to enter the community due to its poor DS^{soc} and DS^{ctx}. Based on the above review of DS calculation, the DS calculation is strict: to be allowed to enter the community, the newcomer nodes have to be capable of providing S^{fct} and holding considerable social relationships in terms of PB. The returner node, such as Re2, its entry is weakly accepted even though its DS^{fct} and DS^{ctx} values are both great.

Table 2.4: DS values for newcomer and returner nodes

DSs	NC1	NC2	Re1	Re2
DS^{fct}	1	1	1	1
DS^{soc}	0.4579	0.1771	0.4579	0.1771
DS^{ctx}	0.5	0.5	1	1
DS	0.5313	0.3463	0.7289	0.5885
AC Decision	Y	N	Y	Y

-Re-entry to the same community or move to a new community

As shown in Fig. 2.8, we consider analyzing three scenarios of AC phase: no checking mechanism in the AC phase, re-entry, and moving to a new community. We chose the NC1 node for (a) and the Re1 node for (b) and (c).

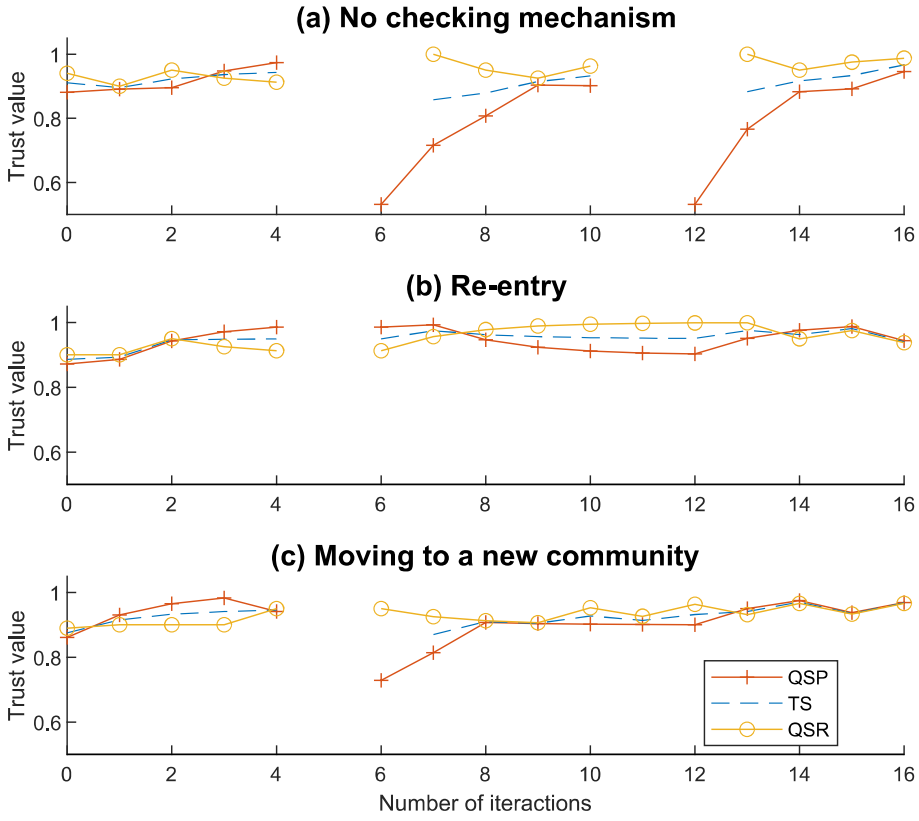


Figure 2.8: Changes in trust values in three scenarios in the AC phase

In Fig. 2.8(a), without the checking mechanism in the AC phase, the comer node suffers from gaining reputation after its re-entry even though its trust values remain great before it quits. Differently, Fig. 2.8(b) gives an example that the node benefits from all previous trust values since it is treated as a returner rather than a newcomer. Lastly, in moving to a new community case, the comer node can only continue using its QSR value but not QSP value.

-Service Provider Ranking by Service Score

We demonstrate the effectiveness of the SP selection phase through two parts, the first one details how SP selection works in this subsection, and the second part explains the importance of this phase in intra-community TM in the next subsection (Section 2.4.2).

Fig. 2.9 gives an example of the SP selection process, where we are looking for 5 SP out of 8 candidates to conduct a workflow → composed by s1 and s2.

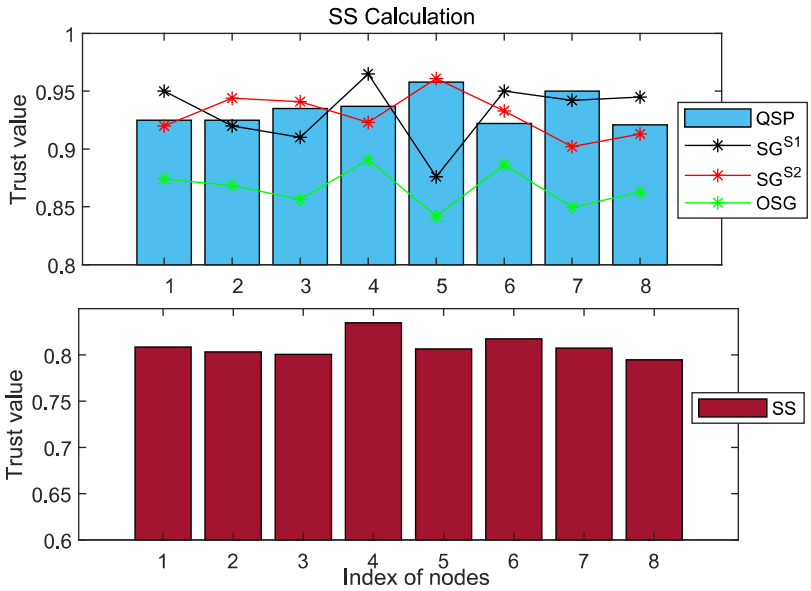


Figure 2.9: SS calculation based on candidate nodes' OSG and QSP

As we can see, a node being outstanding at one service type may not be equally great at others, such as node 4's s1 service. Furthermore, the ranking of SS also relies on the QSP of each node, e.g., node 7 with relatively poor OSG gets fourth place in the SS ranking due to its outstanding QSP. The performance analysis is discussed in the next part.

-Service Provider Selection Process: Importance of Counting Service Composition

We consider 3 scenarios: ranking the candidate SP by SS, only QSP, and without the selection scheme (i.e., randomly select). To realize the comparison between the above scenarios, we select 10 nodes that act in a WSP manner such that their service provision would be rated 0.25. In addition, we employ OSG to compare three scenarios to illustrate the real quality level in terms of service composition.

As shown in Fig.2.10, ranking by SS scenario's curve remains stable and outperforms two others. The scenario without the selection scheme is unsteady, and its SS values are evidently bad. Ranking candidate SP only by QSP case is more stable than the random one, but it is still occasionally exceeded by the random one, i.e., it does not extract the best SP. Moreover, it also has a decreasing trend since QSP is positively correlated with negative feedback. Therefore, ranking by SS combining OSG and QSP is optimal in SP selection, as it enables the selection of the best SP among candidates and prevents SBA by measuring the SG values.

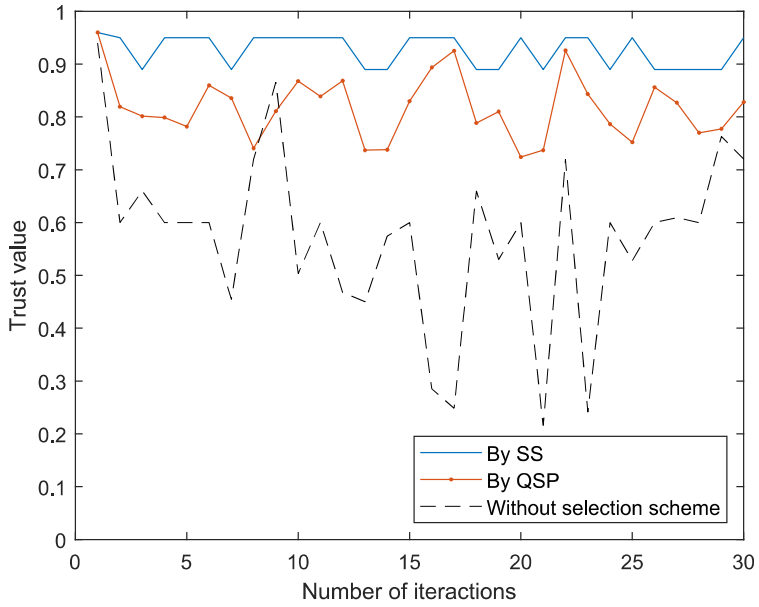


Figure 2.10: Comparison of trust values between three scenarios of the SP selection

-Resilience

This section aims to demonstrate the resilience of the proposed intra-community TM model against attacks on services. First, we focus on the performance evaluation against CBA, SBA, and SPA attacks. Next, we observe changes in related trust values under OOA, BMA, and BSA attacks with a comparative analysis of two other TM models.

CBA:
In the simulation performed to observe CBA, we consider the attacker misbehaving with 30% SR nodes during its service provision. Table 2.5 illustrates average feedback values from the attacked nodes and other nodes to evaluate the attacker.

Table 2.5: Feedback values for the attacker after the attack launched

Description	Value
Avg feedback from the attacked nodes	0.452
Avg feedback from others	0.9442

In Fig. 2.11, QSP of 'with λ' case decreases faster than the case 'without λ' since λ enables the reduction of the QSP of nodes that behave differently with different nodes. The punishment degree of the 'without λ' case is insufficient to segregate the attacker from general nodes, even though the simulation lasts long enough, i.e., it is too difficult to detect a CBA attacker without λ.

SBA:
In this scenario, we deploy one node performing three services as the SBA attacker. We consider all three service types $s1 \sim s3$ targets of SBA, i.e., in each service provision phase, the attacker picks one service type to misbehave (rated 0.45), and it does well for other types (rated 0.95).

Fig. 2.12 illustrates the changes in SG values of three service types. As it can be seen, the attacker switches between services to alternatively behave well and badly, e.g., it recovers SG^{s1}

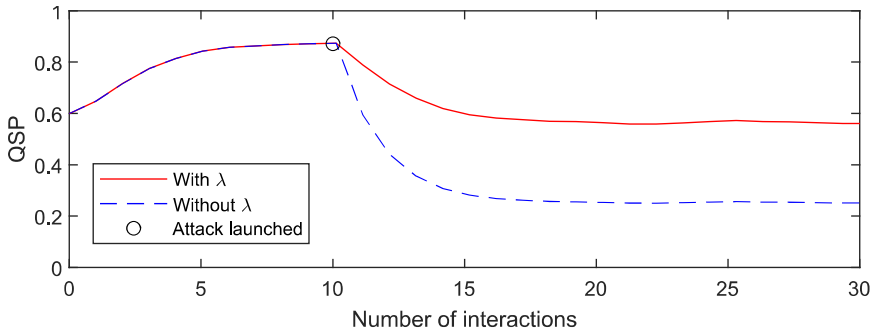


Figure 2.11: Changes in QSP values with λ and without λ in presence of CBA attack.

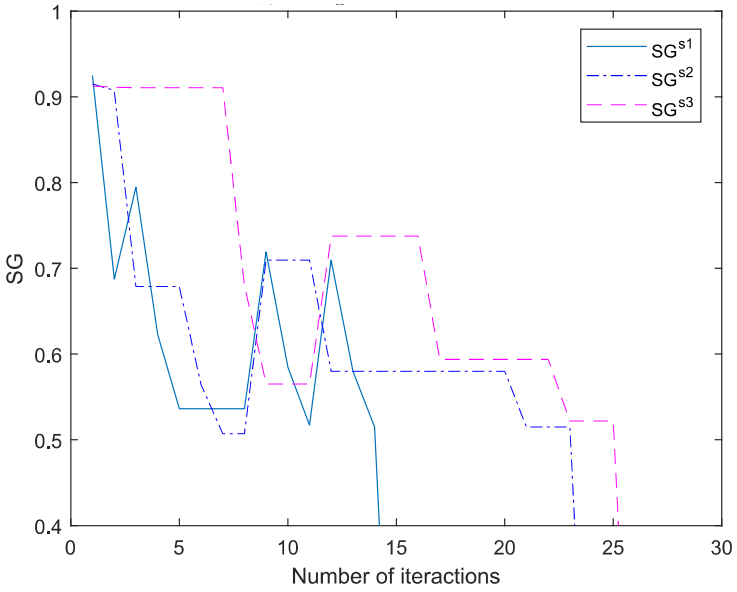


Figure 2.12: Changes in SG values regarding service types in the presence of SBA attack.

and SG^{s2} when misbehaving in $s3$. Eventually, SG values all drop under 0.5. As defined in Section 2.2.4, the service types whose SG goes under 0.5 will be regarded as nonfunctional, and the SP cannot provide such service anymore. Furthermore, we measure the selection score SS by looking at the SG and QSP, the attacker's SG values are poor because of conducting bad services, and this also decreases its QSP value. Therefore, it has less chance of being selected as the SP.

SPA:
SPA mainly consists of two kinds [57]: The first one indicates that an end-user possessing multiple nodes in the network can promote these nodes by self-assigning good feedback. To have greater competitiveness in SP selection, the node may promote its importance by boosting several trust values in the second one. The first one often occurs in SIoT since users can easily hold multiple endpoints, but it is constrained in the proposed intra-community TM model due to the manager as a centralized entity to conduct the local TM. Moreover, the SP is disallowed to rate the service provided by itself in our model. To prevent the second one, it is necessary to exclude the metrics that are not relevant to service type and provider. For example, in our model, the way that the node can improve its importance in the SP selection is to be rated positively to increase its SG and QSP. To do so, it has to conduct outstanding services, and thus earning the reputation without giving satisfying services is impossible.

OOA:
To demonstrate the behavior of the OOA attacker, we set a malicious node that provides services continuously by switching between good (0.95) and bad (0.45) over time. Fig. 2.13 shows that the OOA attacker behaves intelligently to keep its QSP above a certain threshold, e.g., 0.5. With the help of θ, the manager can detect earlier the OOA attacker by measuring the stability of behavior in terms of time and punishing the services without good feedback. Furthermore, it takes a longer time for the attacker to recover its reputation. As discussed in Section 2.2.4, nodes can only gain a reputation through providing good services in a continuous way.

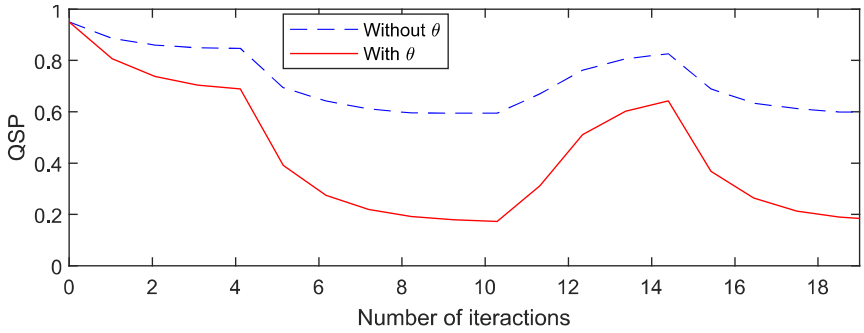


Figure 2.13: Changes in QSP values with θ and without θ in the presence of OOA attack.

BMA/BSA:

These attacks lead a good SP to be snubbed and a bad SP to be promoted. To handle them, comparing individual feedback with average level can determine the honesty of service raters. We set a compromised node that acts as SR that dishonestly rates the 30% of SP (rate 0.45/0.95

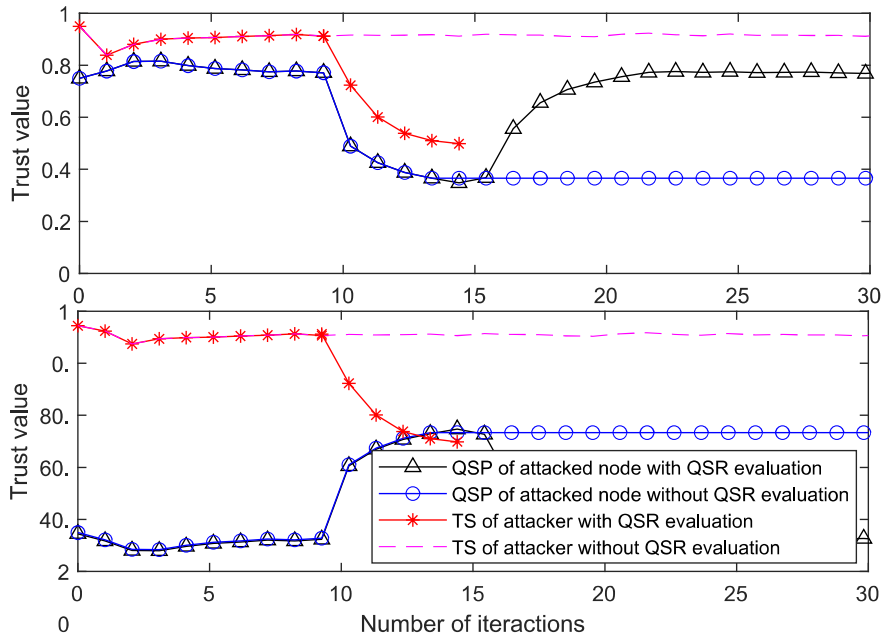


Figure 2.14: Changes in trust values of both attacked and attacker nodes with QSR evaluation and without this evaluation in the presence of BMA and BSA attacks.

for good/bad services). As shown in the upper part of Fig. 2.14, the attacked node's QSP recovers its trustworthiness since the attacker node has been detected and isolated because of $TS < 0.5$. Analogously, badly-performing nodes' QSP drops after the isolation of the attacker node, in the lower part of Fig. 2.14, where we forced the attacker to stay in the community as BSA attacker to visualize the changes in its trust values. Indeed, BMA and BSA act in opposite ways about each other, but they both aim at disrupting the rating mechanism in a way that the good SP does not get positive feedback and the malfunctioning/malicious ones become reputable.

Comparative Analysis

In this part, we compare the proposed model with models proposed in [43] and [44] (thereafter referred as "CoI" and "CITM") to prove the robustness and ability of intra-community TM under OOA, BMA, and BSA. We chose these two models since they are recent TM models addressing the aforementioned three attacks, their proposed TM models are partly suitable for community-driven IoT, where the community is controlled by a community manager. Unfortunately, they did not discuss inter-community trust evaluation, and thus the comparative analysis work involves only the inter-community part. The CoI and CITM models both require adaptations to be simulated with a suitable context, we retain the same number of nodes in the community (called 'cluster' in CITM) and we consider the same scenarios of OOA and BMA/BSA attacks. Firstly, we focus on a comparative performance analysis of the proposed intra-community TM against CoI and CITM models under OOA and BMA attacks, where the attack scenario remains unchanged as above. Next, we focus on an F-score analysis by varying the percentage of malicious nodes (pm) to demonstrate the global performance evaluation, namely recall and precision. To better evaluate the behavior of the attacked node, we only illustrate the changes in the trust value of the attacked SP.

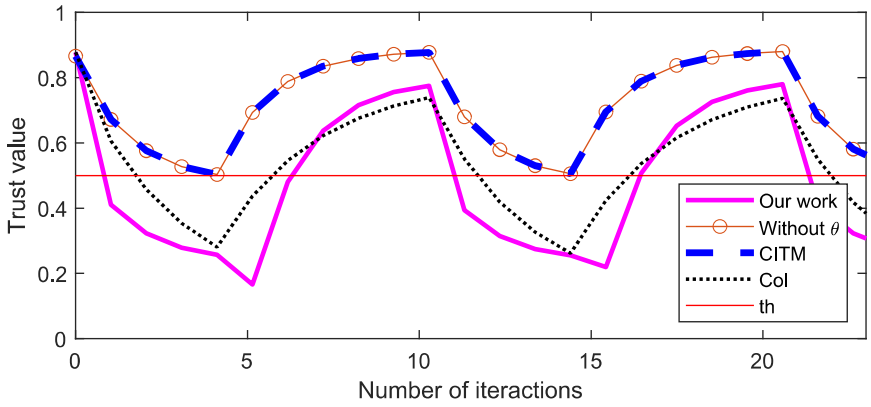


Figure 2.15: Changes in trust values of the attacker node in the presence of OOA attack.

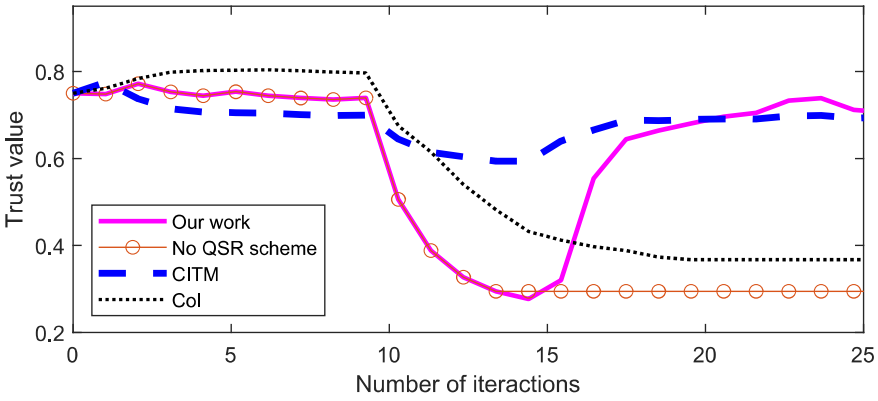


Figure 2.16: Changes in trust values of the attacked node in the presence of BMA attack.

The precision and the recall are defined as follows [58]:

$$\text{Precision} = \frac{tp}{tp + fp}, \quad \text{Recall} = \frac{tp}{tp + fn}, \tag{2.22}$$

where tp refers to attackers accurately detected, fp means normal nodes identified as attackers, and fn counts attackers not detected. As we focus on detecting the attacker node, precision and recall can be considered the accuracy and the sensitivity, respectively.

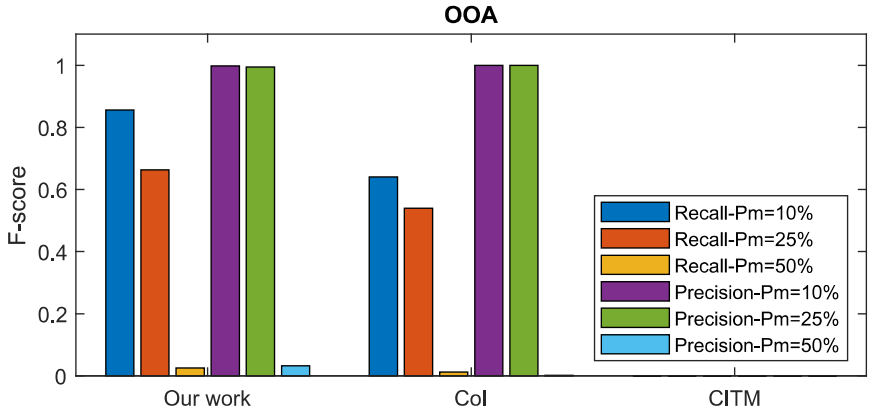


Figure 2.17: Performance of F-scores in the presence of OOA attack.

Fig. 2.15 illustrates the changes in trust values of SP under the OOA attack. We can observe that our model is more reactive than both CoI and CITM models. We also notice that the attacker's trust values never reach the threshold in the CITM model, as well as in the case without θ evaluation in our model. Latter two values represent that unstable service provision damages the current community but the attacker remains undetectable due to the lack of a scheme that accurately punishes nodes switching between good and bad services. In our proposed model and CoI model, the OOA attacker is detected as its trust value is less than the threshold, and we also notice that our model finds the attacker earlier than the CoI model and the misbehaved node can recover its trustworthiness more quickly when it performs satisfactory service.

Fig. 2.16 demonstrates the changes in trust values of SP, namely the attacked node, under the BMA attack. It is obvious that our model recovers the attacked node's trustworthiness, but the trust values continue to decrease and never re-increase in the CoI model and in our work without the QSR scheme, where the convergence of our work is also faster than the CoI model. On the one hand, we notice that the performance of the CITM model has been much less influenced than both our work and CoI models as its algorithm isolates outlier values. On the other hand, this model lacks the BMA/BSA attacker detection mechanism and its algorithm will eliminate honest and fair ratings when the malicious population increases.

As shown in Fig.2.17, with a population of malicious nodes 10% and 25%, our proposed model outperforms the CoI model in the recall factor. This is due to the amplification of changes in QSP values in our model to penalize inconsistent SP. Moreover, both precision and recall factors are given zero in the CITM model, i.e., there are no OOA attackers detected, and this is justified by the lack of mechanism against OOA in the CITM model. We can also notice that precision and recall factors are close to zero when the population of malicious nodes exceeds 50%.

Similarly, Fig. 2.18 visualizes the performance of F-scores under the BMA attack by varying the population of malicious nodes. We can observe that recall and precision factors appear only

in our model because both CoI and CITM lack the attacker detection mechanism. While the population of malicious nodes increases, it is more difficult to detect the BMA attacker accurately in our model. Since the PM reaches 50%, the precision factor remains a low value because half of the nodes within the community are malicious. In our model, the trust computation process is based on a weighted majority voting approach such that the accuracy of attack detection can be mostly guaranteed when the percentage of malicious nodes does not exceed half. However, the inconsistency of trust computation would be created because of malicious nodes' unfairness, with every dishonest node reporting their fake feedback to ruin the system's fairness, and finally, the community trust manager cannot distinguish dishonest or honest ratings. This also explains that F-scores in both Figs. 2.17 and 2.18 decrease significantly when the population of malicious nodes reaches 50%.

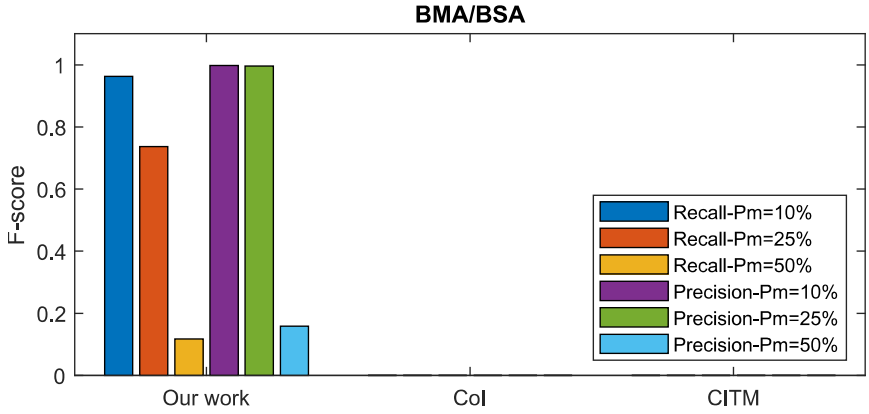


Figure 2.18: Performance of F-scores in the presence of BMA attack.

From the above observation of simulation results and analysis of the proposed countermeasures, we can summarize that the defense techniques against the attacks on services can be categorized into two groups: preventing the source of the attack and punishing the misbehavior. The former type aims to make attacks avoidable, while the latter can only react after the attack has occurred. In the proposed TM model, NCA and SPA are addressed with predefined strict policies as described in Sections 2.2.6 and 2.4.2. The countermeasures of other attacks, namely OOA, CBA, SBA, BMA, and BSA, are exclusively working after the service ratings are launched since defense strategies compare the individual opinion with others' or detect the gaps in terms of the time or service types. To summarise, the first type of attacks can be bounded by a systematic barrier, such as setting up a centralized TM to prohibit multiple identities, disallowing SP to rate the services by itself, and enforcing dynamic and strict AC policies for newcomers. The second type of attack is more like facing a disciplinary mechanism in which the attacker will be penalized once the misbehavior is detected.

2.4.3 Inter-Community Trust

This section moves to the evaluation of the inter-community TM. We set 3 communities simulation configurations, p1 is the evaluator community and two others are the evaluated ones. Nodes in p1 perform s1∼4 as S^{fct}, but these services are somehow in case 'understaffed', meaning more nodes performing these services are in need. To validate inter-community TM, we consider that nodes in p2 and p3 are able to provide s2∼5 and s3∼7, respectively, which also means p2 holds higher closeness to p1 in terms of service. Besides, they both send nodes performing S^{fct} to

relieve p1's 'understaffed' issue. Moreover, nodes from p2 are forced to misbehave in p1.

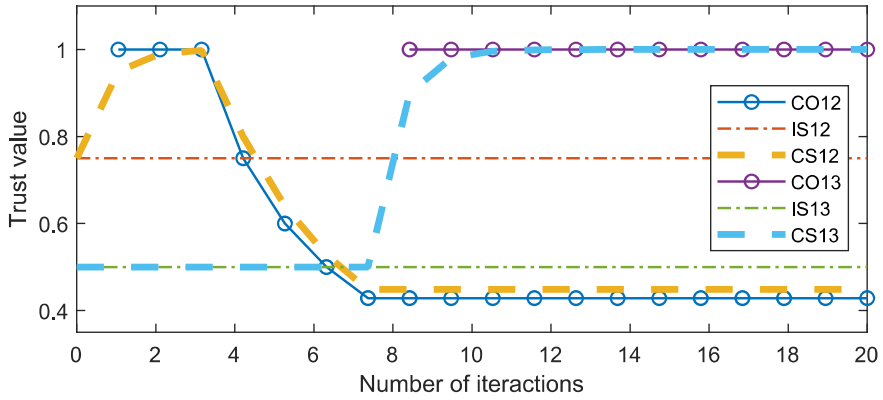


Figure 2.19: Changes in IS, CO, and CS values of p1 evaluating p2 and p3

Since we set p2 to perform in an uncooperative manner with p1, as can be seen in Fig. 2.19, the CO_{12} and CS_{12} values shortly go down, and notably after that, CO_{13} and CS_{13} values increase. p2 sends nodes providing functional services to help p1 address p1's constrained status, as illustrated at the beginning, and then the CS_{12} value quickly decreases due to the nodes from p2 misbehaving in p1, and thus p1 switches the source of nodes to p3. Indeed, initially, p1 does not count p3 as helpful due to the poor IS_{13} value. When p1 detects that p2 is uncooperative, i.e., $CS_{12}<0.5$ as classified into the D-com category in Fig. 2.7, p1 then turns to p3 looking for help. On the other hand, we can notice that while p3 is viewed as less close in terms of service, the increase in CO_{13} and CS_{13} explains the fact that p3 is a trustworthy community to p1 as nodes from p3 perform positively in p1, and thus, p3 matches the C-com category discussed in Section 2.3.4.

2.5 Preliminary Results on Implementation with A Robotic Multi-Agent System

After we simulated the proposed model and analyzed the results obtained, we would like to go further by implementing this model with real-world IoT devices to verify the feasibility and conduct performance evaluations based on the preliminary results of such implementation.

2.5.1 Scenario

As illustrated in the left part of Fig. 2.20 (a), a three-robot scenario is considered for implementation, where robots accomplish a common mission using their cameras to monitor a human cooperatively. The image transmission frequency is fixed at every 500ms. As the preliminary implementation of the designed trust model, a HOG (Histograms of Oriented Gradient) [59] human recognition algorithm from OpenCV is adopted by each Raspberry Pi card to return the probability describing the existence of the target human, i.e., each SR (Raspberry Pi card) evaluates three SPs (cameras). In such a manner, a 3-SR and 3-SP case is built, and the above-mentioned probability will be taken into trust computation as SRs' feedback. Besides, we set λ=0.5 since the historical record and current behavior are considered equally important.

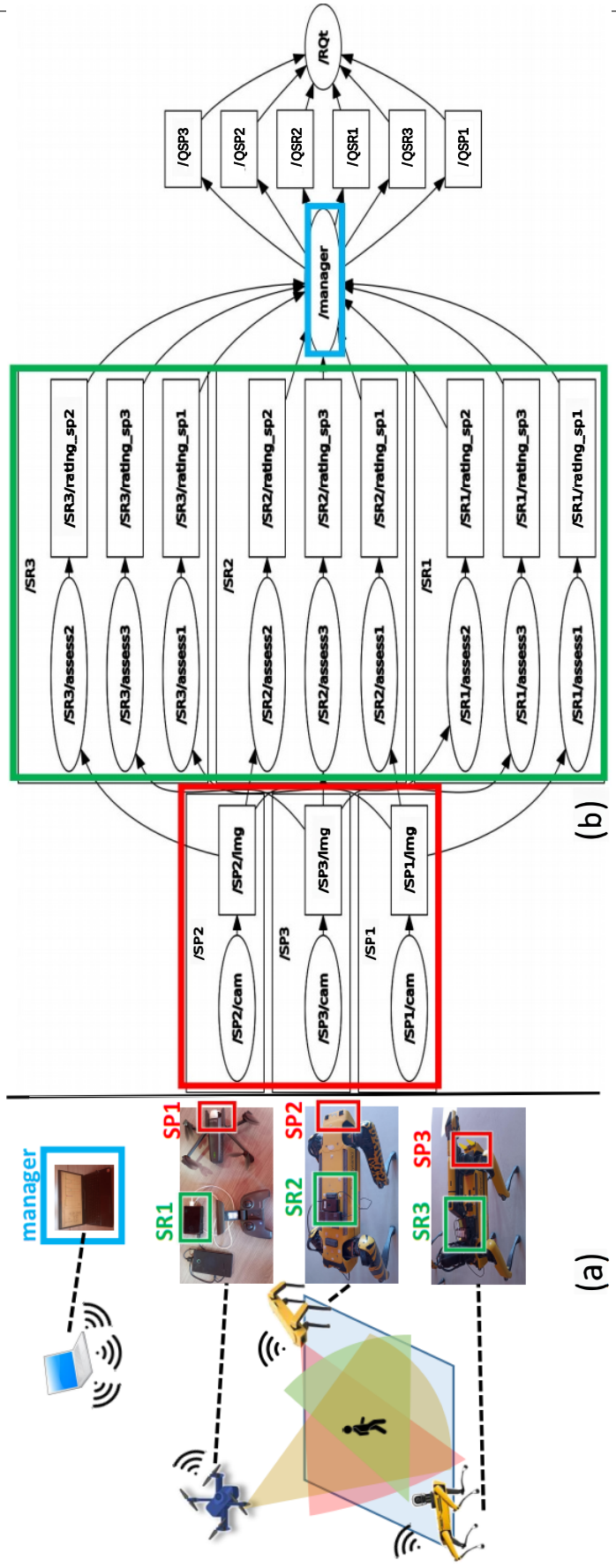


Figure 2.20: Real MAS implementation by using ROS 2, where SR, SP, and trust manager are highlighted by corresponding colors: (a) Considered scenario and implemented hardware; (b) Software-level architecture generated by RQt

Table 2.6: Implemented Hardware

Appearance	Qty	Role in TM
	1	Trust Manager
	2	Service Provider
	1	
	3	Service Rater
	1	Communication

2.5.2 Implemented Hardware

Implemented hardware, as shown in Table 2.6 (a), consists of an aerial drone (Anafi, ™Parrot [60]) and two quadruped ground robots (Spot, ™Boston Dynamics, [61]). Each ground robot is controlled by a Raspberry Pi card, and for the drone, a remote control is connected to one other Raspberry Pi card (All utilized Raspberry pi card are of model 3-B+ [62]). A laptop (Dell 7550) is utilized as the trust manager. All Raspberry Pi cards and the manager run Ubuntu 22.04 and ROS 2 Humble [63], and a 5GHz Wifi access point (model RT AX92U [64]) is set to enable all above-mentioned hardware's communication.

2.5.3 Implemented Software

Compared with other Robotics Software Frameworks (RSF), ROS 2, an open-source software platform for robotics based on DDS (Data Distribution Service) [65], is best suited to multi-agent robotic systems and for data exchange [66]. For this reason, we considered ROS 2 for the implementation. The software architecture is depicted in Fig. 2.20 (b) by a ***Node Graph*** [67], which is composed of nodes, topics, and namespaces. The namespaces correspond to the involved 3 SRs and 3 SPs. Each node is an executed process: the first node `cam` retrieves images from robots' cameras. Each SR contains 3 `assess` nodes that return the feedback assessing 3 SPs. Before calculating QSR and QSP values, the manager will realize an approximate synchronization of nine ratings (feedback) produced by the nine `access` nodes. After that, the node `manager` computes the trust of role-based agents by employing the Trust computation approach in subsection 2.2.4. Finally, 3 QSR and 3 QSP values will be output by the `manager`. While one robot and the equipped Raspberry Pi card can be regarded as an individual SR+SP agent to conduct ToA calculation, we only evaluate SR and SP roles separately in our preliminary implementation.

The Robot Operating System (ROS) 2 is an open-source middleware framework designed to build and manage robotic systems. It represents a significant evolution from its predecessor, ROS, offering enhanced features and capabilities to meet the demands of modern robotics applications. Developed by the Open Robotics organization, ROS 2 addresses the limitations of the original

ROS, making it more scalable, versatile, and suitable for a broader range of robotic platforms.

One notable improvement in ROS 2 is its support for real-time and resource-constrained systems, expanding its applicability to a diverse set of robots, from small embedded devices to large-scale robotic systems. The middleware provides a communication infrastructure, tools, and libraries for developers to build complex robotic applications more efficiently. ROS 2 also introduces a more flexible and modular architecture, allowing for better integration with different software components.

Moreover, ROS 2 fosters interoperability and collaboration within the robotics community by adhering to open standards. Its development is guided by the ROS 2 Technical Steering Committee, ensuring a robust and community-driven approach.

With features like improved security, real-time capabilities, and a broader set of supported programming languages, ROS 2 serves as a foundational framework for the development of cutting-edge robotic applications across various industries, including manufacturing, healthcare, and autonomous systems. Its versatility and commitment to openness contribute to advancing the field of robotics and accelerating innovation in the development of intelligent and capable robotic systems.

2.5.4 Preliminary Results

Via RQt, agents' trust values are illustrated in Fig. 2.21 and Fig. 2.22, where **On-Off Attack** (OOA) and **Bad Mouthing Attack** (BMA) are launched, respectively. We can notice that in both figures SRs and SPs are working properly at the beginning, where QSP and QSR values are close to 1.

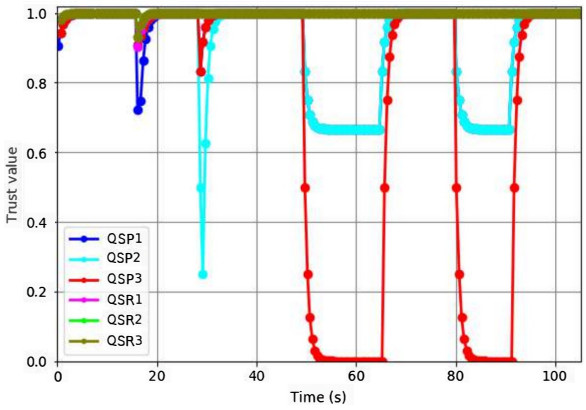


Figure 2.21: Changes in QSR and QSP in the presence of OOA, launched by SP3 at the 50s. Before that, QSP and QSR values converge to 1 but remain unstable due to environmental perturbation (e.g., shooting angles and lighting).

We imposed the target human moves quickly to create environmental perturbation, once at 18 and once at 30 seconds, which explains several changes in trust values before the 40s in Fig. 2.21. Then, between 50 and 90 seconds, the camera SP3 of the quadruped robot was dedicated to performing OOA, where it switches between good and bad over time. It can be seen the red curve representing the OOA attackers' $QSP3$ decreases to 0 while $QSP1$ and $QSP2$ are also slightly lowered. As the gap between the attacker and the well-behaved ones is sufficiently large, the OOA attacker can be identified.

One other type of attack is tested and visualized in Fig. 2.22, where SR3 is fixed as the BMA attacker between 20 and 45 seconds to rate 0.5 for all received services, no matter how

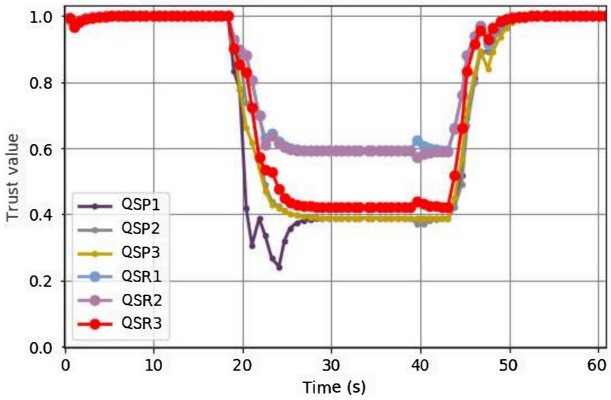


Figure 2.22: Changes in QSR and QSP in the presence of BMA, launched by SR3 at the 18s. While the BMA attacker can be identified, the SPs' QSP values are largely influenced in a negative manner.

the SPs really perform. The attacker aims to ruin the reputation of good SPs by rating them negatively. The red curve representing the BMA attacker's $QSR3$ decreases to 0.4, where $QSR1$ and $QSR2$ remain at 0.6. On the one hand, this figure clearly shows that the BMA attacker SR3 can be distinguished from normal SRs. On the other hand, all well-behaved SPs are influenced harmfully in a way that their QSP values drop to a low level below 0.5. This is because the 1/3 malicious rater case reaches the limit of the Byzantine problem, in a larger-scale MAS with more SRs and SPs, such negative effects caused by dishonest SR will be significantly reduced.

2.6 Conclusive remarks

This chapter presents a role-based attack-resilient dynamic TM model containing both intra- and inter-community trust evaluation, which is suitable for assessing Group-Individual and Inter-Group Trust. First, the intra-community TM enables the IoT nodes within the same community to be monitored dynamically based on their service roles, namely SP and SR. Second, the inter-community TM examines the trust between different communities in terms of cooperativeness. The proposed model has been simulated under various attacks on service. The numerical results show the effectiveness in evaluating both intra- and inter-community trustworthiness. Moreover, the preliminary results of implementation demonstrate the feasibility of the proposed model and also partly validate the proposed model in practice. This chapter focuses on Intra- and Inter-Group trust, and in the following chapter, we will study inter-individual trust evaluation, where we propose a game theoretical model to address nodes' misbehavior.

Inter-Individual Trust in SO-IoT

Chapter 3

A Game Theoretical Model against Strategic Misbehavior in Inter-Individual Trust

3.1 Introduction

To model the interactions between SO-IoT nodes in the context of Inter-Individual Trust, we consider the service process scenario as described in [68], where the service process consists of four main steps: 1) The evaluated node (service provider/worker) launches the task proposal once the communication is established with the evaluator node (service requestor); 2) The evaluated node will be recruited, and then assigned the task; 3) When the task is completed, the

evaluator node sends the incentive; 4) Once the evaluated node is informed of the reception of the incentive, the service-related data will be released. Given this, we design a SBG game to model the interactions between the evaluator node and evaluated nodes appropriately, where evaluated nodes perform actions independently, i.e., there is no inter-affection between them. With the purpose of modeling Inter-Individual Trust in SO-IoT, Game Theory has been taken into consideration. Studies adopting the prisoner's dilemma (PD) game to analyze malicious behaviors in SO-IoT are proposed in [69]. A recent work considering an iterated version of previous prisoner's dilemma (IPD) games to ensure the cooperativeness between SO-IoT nodes was introduced in [70]. However, these two works [69, 70] are both based on a symmetric payoff matrix treating the requestor and the worker in a homogeneous manner. In [71], authors designed an incentive model using the repeated game for SO-IoT, but the defense scheme addressing insider attackers is insufficiently discussed. To treat that the opponent player behaves in a complex manner, the work in [72] allows the players to learn the optimal action in a particular state by maximizing the expected payoff, and the approach proposed in [73] enables players to learn the action frequencies of others conditioned on the modeling player's own action, which is called conditional joint action.

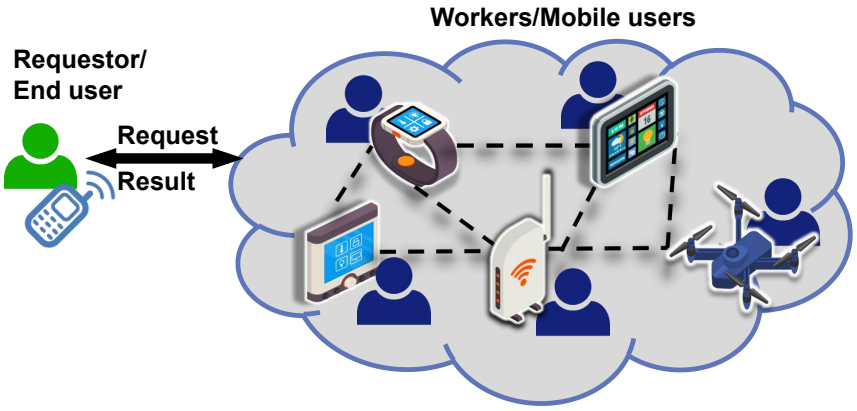


Figure 3.1: Architecture considered evaluating Inter-Individual Trust in SO-IoT

From the above review, there are still several limitations unsolved. First, the majority of existing game theoretical trust management solutions focus on a simple set of actions (e.g., cooperate/defect) such that the actions of the requestor and workers are homogeneous, which does not match the SO-IoT reality. Second, the complex strategic behavioral model of malicious attackers is not taken into consideration, which means that the attacker remains undetectable when switching its actions to mislead the evaluation system. Third, the distinction between self-interested behavior and misbehavior is missing, where the former comes from non-malicious nodes and causes less damage. Lastly, the cooperativeness between SO-IoT nodes, i.e., the requestor and workers, is insufficiently discussed. More precisely, how their cooperation can be encouraged with the aid of the game model. In this context, we propose a game theoretical model using SBG to overcome the above-mentioned limitations. We focus on evaluating the interactions between the evaluator node and evaluated nodes in distributed SO-IoT. Fig. 3.1 illustrates the task evaluator node and evaluated nodes in distributed SO-IoT.

3.2 Stochastic Bayesian Game (SBG)

Uncertainties of mixed behaviors may arise when the evaluator node evaluates the interactions with the evaluated nodes due to the complex attack strategies of malicious evaluated nodes. We formulate the problem as an SBG (Stochastic Bayesian Game), as introduced in [74], where opponent players' behaviors can be modeled through various **behavior types**. The methodology can be adapted to our work: a behavior type refers to one of three categories defined in Fig 1.10, and the **type distribution** can be used to calculate each type's occurrence frequencies. By definition, a general SBG consists of:

- A state space S, including an initial state s^0 and terminal state $\overline{s}$;
- A set of players N of cardinality n, and for each player $i \in N$,
  - An action set A_i for player i's interaction. Throughout, we set $A = A_1 \times ... \times A_n$;
  - A behavior-type space Θ_i modeling player i's type. Throughout, we set $\Theta = \Theta_1 \times ... \times \Theta_n$;
  - A payoff function $u_i : S \times A \times \Theta_i \to \mathbb{R}$, where $u_i(s, a, \theta_i)$ defines the gain/loss of player i whenever the system is in state s, the players have executed the joint action a, and player i has behavior type θ_i.
  - A strategy function π_i: $\mathbb{H} \times A_i \times \Theta_i \to [0, 1]$, where $\mathbb{H}$ denotes the set of all histories $(H^t : t \geq 0)$ of the form $H^t = \langle s^0, a^0, s^1, a^1, ..., s^t, a^t \rangle$, where $s^0, ..., s^t \in S$ and $a^0, ..., a^t \in A$, for $t \geq 0$.
- A state transition function T: $S \times A \times S \to [0, 1]$;
- A type distribution $\triangle$: $\Theta^+ \to [0, 1]$, where Θ^+ is a finite subset of Θ.

For any $i \in N$, the behavior type θ_i of i is sampled from Θ_i before each round of the game. On the basis of the history H^t up to time t, player i selects an action depending on its strategy $\pi_i(H^t, a, \theta_i)$ until the state $\overline{s}$ is reached.

3.3 Game Formulation

3.3.1 Players and action sets

In the proposed game theoretical model, the game is played by the evaluator node r and one typical evaluated node w, i.e., we set $N = \{r, w\}$. Their set of actions is given in Table 3.1: A_r=$\{$**S**, **T**, **D**$\}$, A_w=$\{$**S**, **C**, **I**, **M**$\}$. The action **S** (Standby) is identical for the evaluator node and the evaluated node as they both perform waiting as being standby for the new IoT service. In task completion, the evaluated node performs either **C** (Cooperate) or **M** (Misbehave), otherwise, it performs **I** (Interruption) in case it does not contribute to the task. The difference between selfish and malicious behaviors through actions **I** and **M** should be noted as the evaluated node does not produce any false information in the service-related data by performing the former action, whereas the latter does, which also leads to more negative consequences caused by the latter action. After receiving the service-related data from the evaluated node, the evaluator node will perform either **T** (Trust) or **D** (Distrust) depending on its own strategy, which will be discussed in Section 3.6.

Table 3.1: Set of actions of the evaluator node and the evaluated node

Player	Action	Description
Requestor	S (Standby)	Wait to begin a new crowdsourcing service process.
	T (Trust)	Trust the data crowdsourced by the worker and release the incentives.
	D (Distrust)	Distrust the data crowdsourced by the worker and lower the incentives.
Worker	S (Standby)	Wait to begin a new crowdsourcing service process.
	C (Cooperate)	Task is assigned and complete it with efforts.
	I (Idle)	Task is assigned but not engage in the task.
	M (Misbehave)	Task is assigned but perform misbehavior for crowdsourcing service.

3.3.2 Game States

Employing the set of actions illustrated in Table 3.1 to fit the IoT service process described before, we consider seven-game states in SBG game, which are given in Table 3.2 with a description per each.

Table 3.2: Game states

State	Description
PE (Process End)	The crowdsourcing service process ends or the communication between the requestor and the worker fails.
S (Standby)	Both requestor and work stay at Standby waiting to begin the new crowdsourcing service process.
TC (Trust, Cooperate)	The requestor acts Trust, and the worker acts Cooperate.
DC (Distrust, Cooperate)	The requestor acts Distrust, and the worker acts Cooperate.
TM (Trust, Misbehave)	The requestor acts Trust, and the worker acts Misbehave.
DM (Distrust, Misbehave)	The requestor acts Distrust, and the worker acts Misbehave.
I (Interruption)	The worker is assigned the task but does not engage in the task, and thus the crowdsourcing service process is interrupted.

In some cases, and as in the first experimentation conducted in [75], the state of the system at time t can be represented as the joint action at the previous time, i.e., we set $s^t = a^{t-1}$. This is precisely the case whenever the joint action (and thus, the state of the system at the next time slot), is **I**, **TC**, **DC**, **TM** and **DM**. This also means that game states are not homogeneous, and we name such particular states 'action states'. For example, if the evaluator node and the evaluated node perform the joint action (Distrust, Cooperate) contemporarily, then we consider

that the state of the system at the next time slot is precisely **DC**. Furthermore, the utilization of action states simplifies the presentation and storage space of H, and the players' payoff can also be more efficiently matched through Table 3.3.

In **PE** state, both the evaluator node and evaluated node cannot communicate with each other as the service process ends or their communication fails in this state. If a new service process is launched or the communication recovers, the game turns to a Standby state, where both the evaluator node and evaluated node perform action Standby for IoT service. After the task proposal is released, the evaluated node will be recruited and assigned the task accordingly. Unlike other action states, the state **I** will be reached if the evaluated node has made no contribution and must return to **PE** as the service process will be viewed as ended. Or, one of **TC**, **DC**, **TM**, and **DM** states will be reached as the result of the current service process. We determine a goal number of interactions as total game rounds to avoid the game being played infinitely. This also means that it is not necessary to return to **S** state from action states for every service process, which can be observed through transitions between action states **TC**, **DC**, **TM**, and **DM**. Based on the above description, Fig. 3.2 presents the diagram of possible transitions between states, i.e., state space S={**PE**, **S**, **TC**, **DC**, **TM**, **DM**, **I**}, the initial state s^0 and terminal state $\overline{s}$ are both the state **PE**.

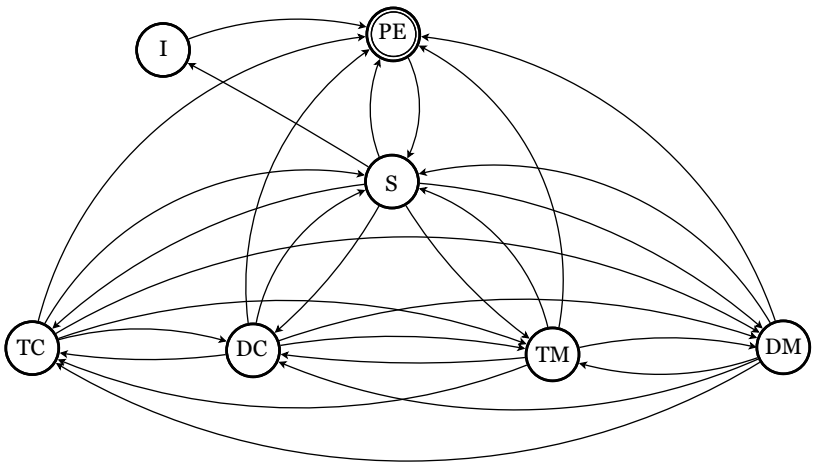


Figure 3.2: Diagram of possible transitions between game states of the proposed model

For any state s and action a, we set the transition $T(s, a, \mathbf{PE})) \equiv P_E$, where the value P_E represents the probability that communication between players fails. On the other hand, from state **PE**, the system must move to the state **S** in which both players perform Standby, and thus $T(\mathbf{PE}, a, \mathbf{S}) = 1$ for any joint action a, in order to start a new service process. We denote by $P_I := T(\mathbf{S}, \cdot, \mathbf{I})$, the transition to the state in which the evaluated node performs action **I**. As for other transitions depending on the strategies for both evaluator node and evaluated node will be explained in the following subsections.

3.4 Payoffs

Based on the game states defined in the previous section, the payoffs of the evaluator node and the evaluated node (u_r, u_w) are given in Table 3.3:

Following the service process defined in Section 3.1, there are some constraints in payoffs:

Table 3.3: Payoff matrix of the evaluator node and the evaluated node

r \ w	S	C	I	M
S	-	-	$-Cr_S$,0	-
T	-	Gr_{TC} - Cr_T , Gw_{TC} - Cw_C	-	$-Cr_T$ - Lr_{TM} , Gw_{TM} - Cw_M
D	-	- Cr_D , Gw_{DC} - Cw_C	-	$-Cr_D$, Gw_{DM} - Cw_M

Gr=Gain of the requestor; Lr=Loss of the requestor; Cr=Cost of the requestor;
Gw=Gain of the worker; Cw=Cost of the worker.

- As the gain of the evaluated node represents the incentives offered by the evaluator node, we impose that $Cr_T = Gw_{TC}$, and likewise, that for $Gw_{DC} = Gw_{DM} = Cr_D$.
- We impose that $Gr_{TC} - Cr_T > Cr_T$, this is because the overall payoff of the evaluator node, after a normal service, should be greater than its cost of performing action **T**. Otherwise, it would become discouraged to request due to a non-reasonable obtained payoff. Similarly, as the malicious evaluated node aims to cause damage such that it gains a higher overall payoff than the cost of misbehaving, we set $Gw_{TM} - Cw_M > Cw_M$.
- $Cr_T > Cr_D > Cr_S$, as performing **T** signifies greater incentives are required than the action **D**, and no incentives are offered when the evaluated node performs action **I**.
- Attacking behaviors cost more than cooperating for the malicious evaluated node, as it has to create false information based on original service-related data. Thus, we impose that $Cw_M > Cw_C$.
- By convention, the loss and the cost of the evaluator node should be equal to the gain of the malicious evaluated node. Thus $Lr_{TM}+Cr_T = Gw_{TM}$.
- As the malicious evaluated node should obtain a higher overall payoff when its misbehavior successfully misleads the evaluator node, we impose that $Gw_{TM} - Cw_M > Gw_{TC} - Cw_C$.
- We set $Gw_{DM} - Cw_M < 0$, otherwise the attacker would receive a positive payoff while the evaluator node performs distrust, an absurdity.

3.5 Strategies for the evaluated node

In studying security by applying Game Theory, it is essential to define the threat model, specifically our assumptions about the behavioral model of malicious attackers. As stated in Section 3.1, one of the limitations of existing IoT security solutions is the lack of effective distinction between selfish and malicious behaviors. For example, a SO-IoT evaluated node performing inactive or selfish cannot be certainly determined as malicious type, it may perform action **I** with the purpose of maximizing its benefit by reducing energy consumption. Given this, the Byzantine Altruistic Rational (BAR) model [35, 36] can be employed for the threat model, where the evaluated node is classified into three categories. Given this, as the type space Θ_w of the evaluated node is unknown, we assume instead that the evaluator node hypothesizes a user-defined type space $\Theta_w^* = \{\theta_w^A, \theta_w^R, \theta_w^B\}$, where the types are as follows:

- **Altruistic** (θ_w^A)
- **Rational** (θ_w^R)
- **Byzantine** (θ_w^B)

Table 3.4: Evaluated node types with the definition of strategies

Type	Beh.	Definition
θ_w^A	AC	$\pi_w(H^{\mathrm{t}}, \mathrm{C}, \theta_w^A) = 1$
θ_w^R	RC	$\pi_w(H^{\mathrm{t}}, \mathrm{I}, \theta_w^R) = \mathrm{P_I}, \quad \pi_w(H^{\mathrm{t}}, \mathrm{C}, \theta_w^R) = 1 - \mathrm{P_I}$
θ_w^B	RS	$\pi_w(H^{\mathrm{t}}, \mathrm{C}, \theta_w^B) = 1/3, \pi_w(H^{\mathrm{t}}, \mathrm{I}, \theta_w^B) = 1/3, \pi_w(H^{\mathrm{t}}, \mathrm{M}, \theta_w^B) = 1/3$
	AM	$\pi_w(H^{\mathrm{t}}, \mathrm{M}, \theta_w^A) = 1$

Beh.=Behavior; AC=Always Cooperate; RC=Rational Cooperate;
RS= Random Shift; AM=Always Misbehave;

Fig. 1.10 gives the Euler diagrams of the possible behaviors according to the BAR-based threat model. It can be observed that the altruistic ones can only perform positive actions, such type of evaluated node is regarded as reliable and well-resourced. The rational can act not only altruistically but also selfishly. The reasons for rational behaviors are various, such as resource constraints that force the evaluated node to interact with others in a selective manner; or, the evaluated node, based on its own judgment of utility, desires to refuse cooperation rather than to engage in it. However, rational evaluated node cannot behave in a way that threatens the current IoT system, while the byzantine kind possesses the highest number of possible behaviors, which include all behaviors of the previous two types. To rephrase, a byzantine evaluated node can behave maliciously, altruistically, or rationally, depending on its purpose (i.e., being harmful, hiding its true motivation, and so on).

Table 3.4 defines the behaviors per type of BAR-based threat model: AC evaluated node cooperates in any case. RC evaluated node will perform **I**, if it is not willing to contribute to the task, i.e., with the probability P_I, otherwise it cooperates. In our work, we consider two malicious strategic misbehavior, namely RS and AM. RS evaluated node randomly shifts its behaviors in every game round (i.e., the probability of performing action **C**, **I**, or **M** is identical), and the AM evaluated node misbehaves for all time.

3.6 Strategies for the evaluator node

To track the mixed behavioral evaluated node and output the action decision-making of the evaluator node, we adopt the algorithm *Harsanyi-Bellman ad-hoc coordination* (HBA) as the strategies of the evaluator node [74]. HBA utilizes the concept of Bayesian Nash equilibrium in a planning procedure to find optimal actions in the sense of Bellman optimal control [74]. Here, we still fix r to represent the evaluator node, and thus the HBA strategy is defined as $a_r^t \in arg\,max_{a_r \in A_r} \mathbb{E}_{s^t}^{a_r}(H^t)$ for all t, where for any state $s \in S$, any action $a_r \in A_r$, and any history $\hat{H}$,

$$\mathbb{E}_s^{a_r}(\hat{H}) = \sum_{\theta_w^* \in \Theta_w^*} \Pr(\theta_w^* | \hat{H}) \sum_{a_w \in A_w} Q_s^{(a_r, a_w)}(\hat{H}) \pi_w(\hat{H}, a_w, \theta_w^*) \tag{3.1}$$

is the long-term payoff of the evaluator node taking the action a_r in the state s after history $\hat{H}$, and for all $a \in A$,

$$Q_s^a(\hat{H}) = \sum_{s' \in S} T(s, a, s') \left[u_r(s, a) + \gamma \max_{a_r \in A_r} \mathbb{E}_{s'}^{a_r}(\langle \hat{H}, s', a \rangle) \right] \tag{3.2}$$

determines the long-term payoff for the evaluator node r when joint action a is executed in state s after history $\hat{H}$, $\gamma \in [0, 1]$ is the discount factor whose value will be detailed in Section 3.7, and $\langle \hat{H}, s', a \rangle$ in (3.2) denotes the concatenation of history $\hat{H}$ with a future state and joint action (s', a).

We consider that the behavior of the evaluator node r is completely specified by HBA, that is, in our model r has a single fixed type: $\Theta_r^+ = \{\theta_r^{HBA}\}$, where θ_r^{HBA} is outputted by (3.1) and (3.2). For any history $\hat{H}$ and any $\theta_w^* \in \Theta_w^*$, the definition of the posterior belief probability (PBP) $\Pr_w(\theta_w^*|\hat{H})$ in (3.1) follows from Bayes formula:

$$\Pr_w(\theta_w^*|\hat{H}) = \frac{L(\hat{H}|\theta_w^*) P_w(\theta_w^*)}{\sum_{\hat{\theta_w^*} \in \Theta_w^*} L(\hat{H}|\hat{\theta_w^*}) P_w(\theta_w^*)}, \tag{3.3}$$

where $P_w(\theta_w^*)$ is the prior belief (probability) that the evaluated node w is of type θ_w^* before any action is observed, and $L(\hat{H}|\theta_w^*)$ is the likelihood of history $\hat{H}$ under the assumption that the evaluated node w is of type θ_w^*. To specify the likelihood L in (3.3), we consider the sum posterior given in [74], which allows HBA to learn mixed type distribution by recognizing changing types. Thus, $\hat{\theta_w^*}$ in (3.3) refers to all possible hypothesized types of the evaluated node w in $\hat{H}$.

3.7 Simulation Configuration

3.7.1 Parameter Settings

Table 3.5: Simulation parameter values

Parameter	Value		Parameter	Value
P_E	0.1		$\mathrm{Lr_{TM}}$	0.55
P_I	0.2		$\mathrm{Gw_{TC}}$	0.45
γ	0.9		$\mathrm{Gw_{TM}}$	1
$\mathrm{Gr_{TC}}$	1		$\mathrm{Gw_{DC}}$	0.1
$\mathrm{Cr_S}$	0.05		$\mathrm{Gw_{DM}}$	0.1
$\mathrm{Cr_T}$	0.45		$\mathrm{Cw_C}$	0.2
$\mathrm{Cr_D}$	0.1		$\mathrm{Cw_M}$	0.45

As illustrated in Table 3.5, we set P_E 0.1. For the same IoT protocol, its failure rate calculated in [76] is 12%, and its communication stability given in [77] is 0.92, we take the average of these values to represent the probability that the communication fails, i.e., P_E=0.1 (obviously [0.12+(1-0.92)]/2=0.1). In our proposed model, we consider the rational evaluated node may perform action **I** in case of resource-constrained, e.g., in trouble of insufficient battery. In the

above table, P_I is fixed at 0.2 as we employ the value of a parameter in work [78] describing the capability of solving battery issues, which is set to 0.8. Thus we consider P_I=1-0.8=0.2. The discount factor γ is fixed at 0.9 as in [74]. Besides, the initial prior considered is uniform prior for the empty history, where three types have identical prior values. For the favorable history, the initial prior is calculated by (3.3), as the posterior of the previous time slot. We fix the maximal gain to 1 for both the evaluator node and the evaluated node. By respecting the constraints of payoffs mentioned in Section 3.4, the rest of the parameters are accordingly assigned as given in Table 3.5: $Gr_{TC}(1)$-$Cr_T(0.45)$>$Cr_T(0.45)$; $Gw_{TM}(1)$-$Cw_M(0.45)$>$Cw_M(0.45)$; $Cr_T(0.45)$>$Cr_D(0.1)$>$Cr_S(0.05)$; Gw_{DC}=Gw_{DM}=Cr_D=0.2; $Lr_{TM}(0.55)$+$Cr_T(0.45)$=$Gw_{TM}(1)$; Gw_{TC}=Cr_T=0.45; $Gw_{TM}(1)$-$Cw_M(0.45)$>$Gw_{TC}(0.45)$-$Cw_C(0.2)$; $Cw_M(0.45)$>$Cw_C(0.2)$; $Gw_{DM}(0.2)-Cw_M(0.45)$<0. Fixing the target number of interactions between the evaluator node and the evaluated node, we run 50 game rounds for simulation. To validate the effectiveness of our proposed model, we will evaluate the changes in PBP given by (3.3), the occurrence rate of game states obtained per scenario, and the average payoff of each scenario.

r \ w	S	C	I	M
S	-	-	-0.05 ,0	-
T	-	0.45 , 0.25	-	-1 , 0.55
D	-	- 0.1 , -0.1	-	-0.1 , -0.35

Table 3.6: Payoff matrix with parameter values

3.7.2 Scenarios Considered

In the simulation, the type-based behaviors of the evaluated node defined in Table 3.4 are all taken into consideration, namely AC, RC, RS, and AM. We designed two kinds of history, as illustrated in Table 3.7.

Table 3.7: Scenario description

π_r	controlled by HBA				
π_w	AC	RC	RS		AM
H	Ø	Ø	Ø	F	Ø

Ø= Empty history; F=Favorable history.

- An empty history where the evaluator node and the evaluated node have yet to interact.
- A favorable history in which only **TC** is reached among action states, and this signifies a possible situation where the attacker hides its true behavioral type by performing only cooperate in the past and it starts misbehaving at a moment given, this also corresponds to the intelligent attack types of the insider attacker analyzed in [2].

The favorable history and RS misbehavior will be utilized for the comparative analysis since they somehow represent a more complex context in simulation.

3.8 Performance Evaluation under Different Scenarios

3.8.1 AC scenario:

As we can see in Fig. 3.3, since the action **C** can also be performed by a Rational evaluated node, this PBP value of the Rational type increases a little at the beginning. However, this value goes down rapidly due to no action **I** performed at all by the evaluated node, and finally converges to 0. On the other hand, the PBP value of the Altruistic type converges to 1, which corresponds to the occurrence rate diagram in the same figure, showing that the evaluator node only performed the action **T** based on HBA to optimize the evaluator node's long-term payoff.

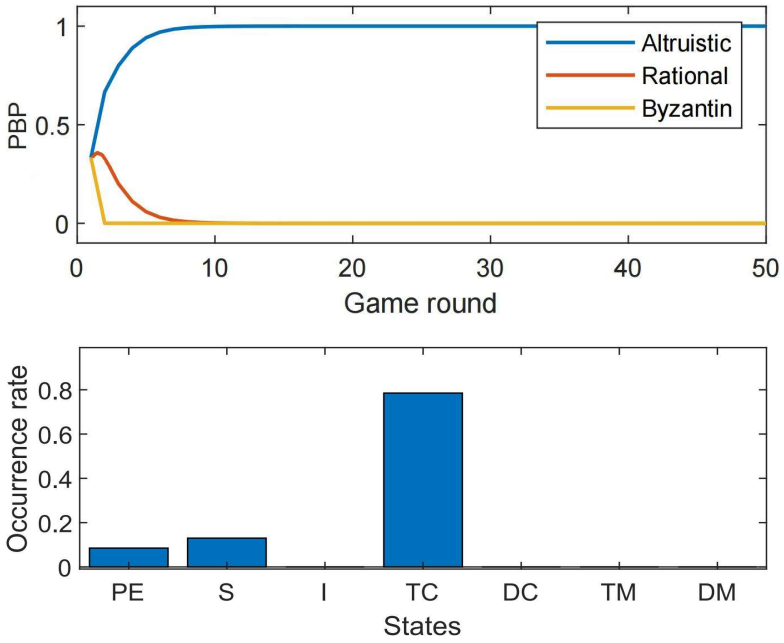


Figure 3.3: Changes in PBP and the occurrence rate of game states in AC evaluated node scenario

3.8.2 RC scenario:

Differently, the changes in PBP values in Fig. 3.4 show that the Rational evaluated node performed action **I** at the very early game rounds, and thus the PBP value of the Altruistic type decreases. After that, the evaluated node cooperated with the evaluator node so that we can observe a rise in the PBP value of the Altruistic type around the tenth game round. With more and more action **I** being performed by the evaluated node, the PBP of the Rational type increases steadily while the evaluated node performed cooperate in some cases. Since **I** state must return to **PE** state by definition, it can be noticed that the occurrences of **PE** and **S** state are relatively higher than in other scenarios. Furthermore, it can be noticed that the occurrence of **I** state is much lower than that of **TC** state, which matches the value of P_I=0.2.

3.8.3 RS scenario:

As the most complex malicious behavior defined in Table 3.4, the RS evaluated node will randomly shift its actions between **I**, **C**, and **M**, it can be observed in Fig. 3.5 that the occurrences

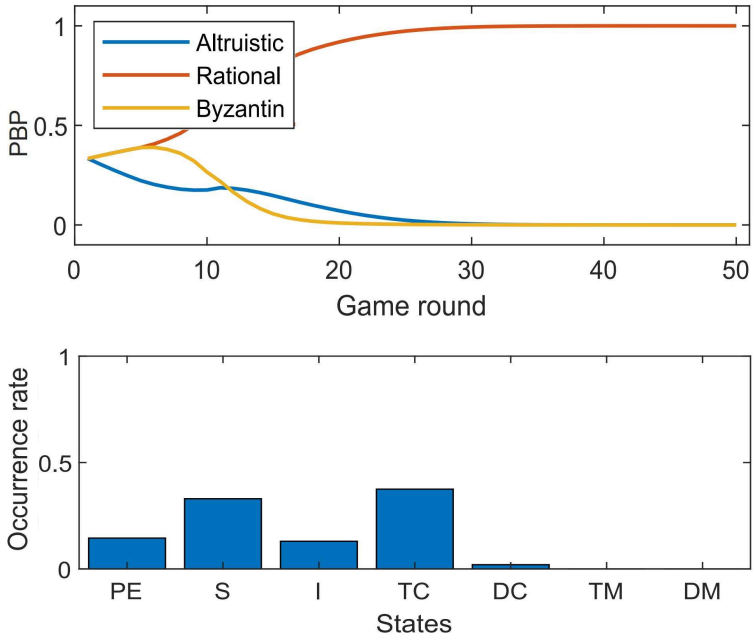


Figure 3.4: Changes in PBP and the occurrence rate of game states in RC evaluated node scenario

of state **I**, **DC**, and **DM** are close. The changes in PBP value demonstrate that the evaluated node performed action **I** at the beginning. And then, the value of the Rational type goes down immediately due to the misbehavior of the evaluated node, we can also observe that the evaluated node repeated action **I**, which leads to the PBP value of the Rational type remaining unchanged. On the other hand, the evaluator node performed very rarely action **T**, this is because performing **D** maximizes the long-term payoff. In other words, to cope with a RS malicious evaluated node, performing action **D** is optimal based on the calculation of HBA. Finally, with more and more action **M** being performed, the PBP of the Byzantine type converges to 1.

3.8.4 AM scenario:

The AM evaluated node will misbehave immediately from the beginning, and the evaluator node will perform action **T** at the first game round since the HBA maximizes its long-term payoff for the first interaction with the evaluated node, which outputs that it will perform action **T**. As the malicious evaluated node continuously misbehaves during the game, the PBP of the Byzantine type in Fig 3.6 increases till it converges to 1, and PBP values of Altruistic and Rational types are overlapping and both decrease to 0. Except one **TM** state is reached as the evaluator node performed action **T**, only **DM** is reached among all action states as the type of the evaluated node is reasoned as Byzantine, the evaluator node will keep distrusting the evaluated node. In our simulation, we run 50 game rounds even though the type of malicious evaluated node has been identified. Indeed, such malicious evaluated node will be removed from the group of evaluated nodes once it is remarked as the attacker.

3.8.5 Average Payoffs

Fig. 3.7 illustrates the average payoff obtained by the evaluator node and the evaluated node after running 50 game rounds. As we can see, AC and RC scenarios are win-win cases for the

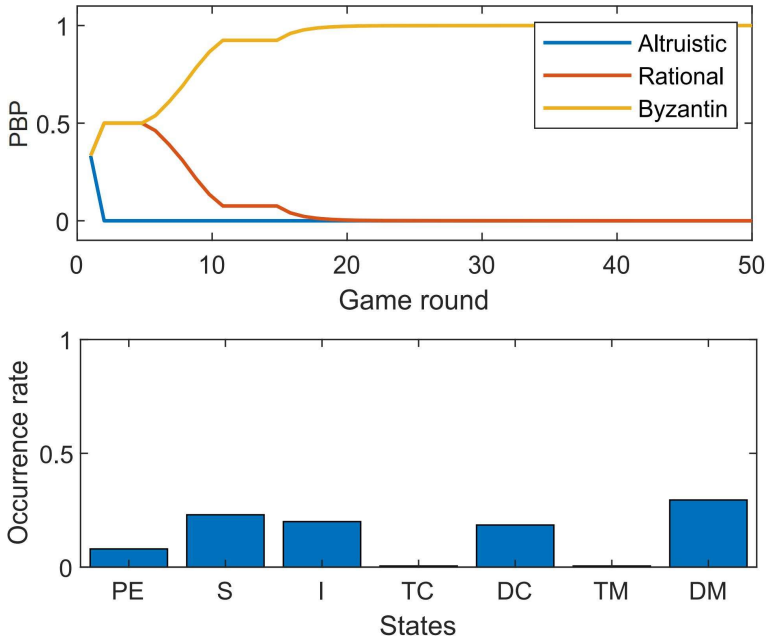


Figure 3.5: Changes in PBP and the occurrence rate of game states in RS evaluated node scenario

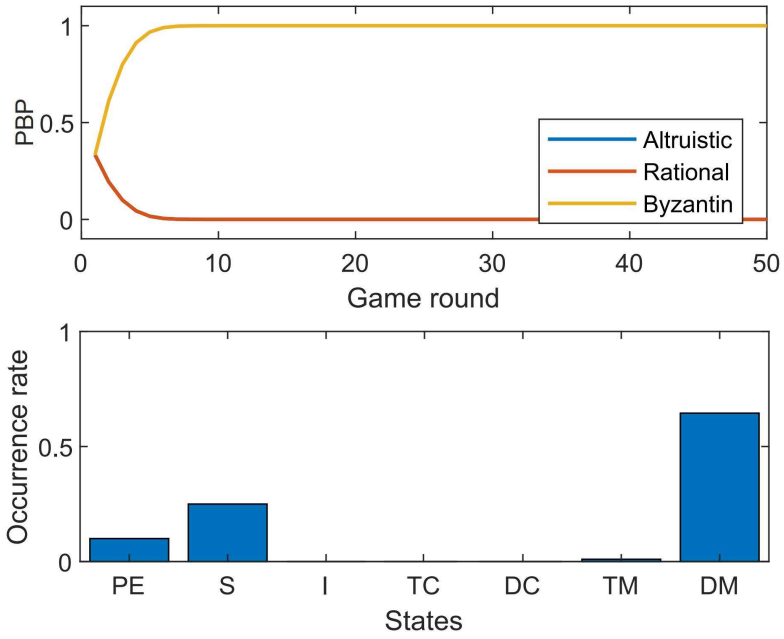


Figure 3.6: Changes in PBP and the occurrence rate of game states in AM evaluated node scenario

evaluator node and the evaluated node. Although the RC evaluated node performed more **I** action leading to a cost for the evaluator node, Fig. 3.7 indicates that their interactions still output positive payoffs, i.e., the selfish evaluated node is not considered as malicious. On the other hand, in RS and AM scenarios, the evaluator node and the evaluated node both receive negative payoffs, but it is obvious that the evaluated node loses much more, which means the evaluator node is able to minimize its loss when playing with a malicious evaluated node. From

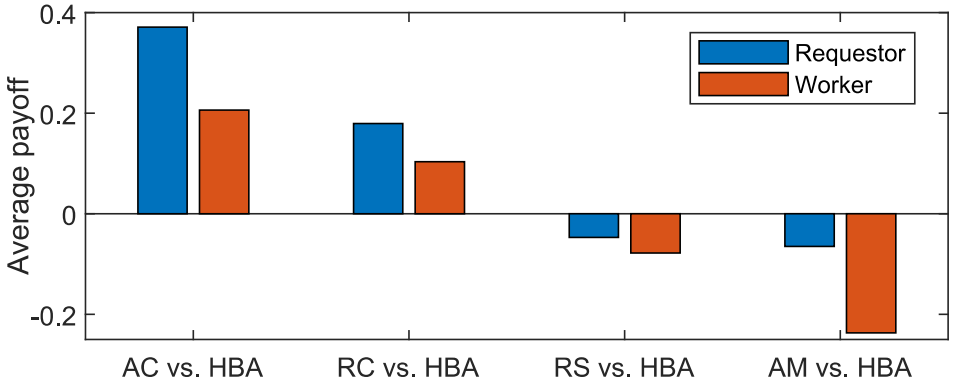


Figure 3.7: Average payoff of the evaluator node and the evaluated node in different scenarios

the above performance evaluation based on changes in PBP values, the occurrence rate of game states, and average payoff in different scenarios, we notice that the proposed model encourages cooperation between the evaluator node and the evaluated node, reduces the loss and the cost of the evaluator node if facing malicious evaluated node, and penalizes the malicious evaluated node. Moreover, the changes in PBP values become stabilized within 50 game rounds, which also results in the true types of the evaluated node being identified accurately.

3.9 Comparative Analysis with Other Approaches

In this subsection, we compare our work with approaches presented in [72] and [73] (thereafter referred as "QL" and "CJAL") to demonstrate the ability of the proposed model in the presence of RS evaluated node with a favorable past history. We chose these two approaches since comparing our work with other game theoretical trust models remains demanding due to the variety of game formulations and payoff matrix, and these two approaches are reputable learning schemes that enable modeling the behavior of opponent players to optimize player's payoff. We involve RS scenario and a favorable past history for having a complex initialization of the game. QL and CJAL approaches both require adaptations to be simulated with a SO-IoT context, we thus retain the same parameter settings given in Table 3.5.

As shown in Fig. 3.8, in the presence of RS malicious behavior, the proposed model outperforms the other two approaches in the average payoff obtained. By reviewing the three approaches' occurrence rates of game states, we can notice that the evaluator node of QL approach performs more action **T** as the **TC** and **TM** states are reached more. On the other hand, the action **D** is much less performed than CJAL and ours, particularly when the malicious evaluated node performs action **M**. This also explains that the evaluator node of QL receives the worst average payoff among all three approaches, and its evaluated node obtains a very small negative payoff meaning that nearly no loss is caused. As for CJAL, as the evaluator node of CJAL approach performs less **T** and more **D** actions, the misbehaving of RS evaluated node does not damage the evaluator node as much as of QL, and thus the evaluator node's loss is reduced. On the other hand, the success rate of RS evaluated node's misbehaving becomes smaller, and consequently, the malicious evaluated node will receive a lower negative payoff than the QL evaluated node, which means the evaluated node's misbehavior is punished more by applying CJAL. It can be seen in this figure that the evaluator node's average payoff is higher than the malicious evaluated node in our work, which QL and CJAL both cannot achieve. This is because our work

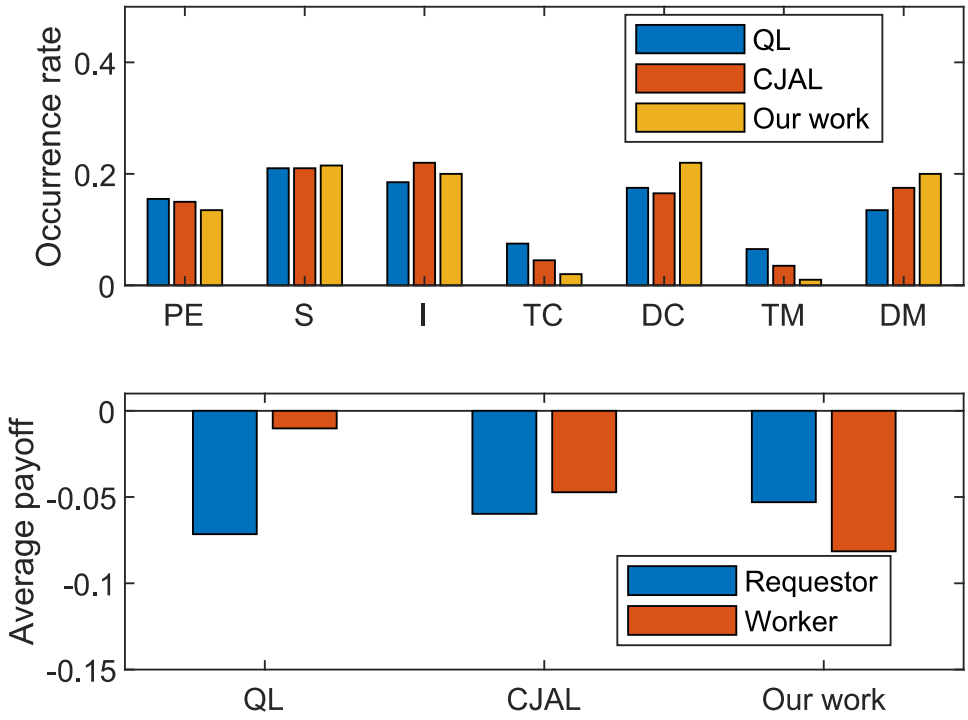


Figure 3.8: Comparison between different approaches based on occurrence rate and average payoff in RS scenario with Favorable history

reaches more **TC** and **DM** states such that the evaluator node gains more when the evaluated node performs cooperate and loses less when the evaluated node misbehaves, which also means that, compared with QL and CJAL, the evaluator node's optimizes its strategies more efficiently when facing RS malicious evaluated node. Moreover, as a malicious attacker with favorable history corresponds to the newcomer attack behavior discussed in [2], where the attacker benefits from refreshing its historical record, our work also shows resilience against this attack type.

3.10 Conclusive remarks

We propose a Stochastic Bayesian Game (SBG) to address the Byzantine Altruistic Rational (BAR) based misbehavior in Inter-Individual Trust evaluation, where service workers' behavioral types can be deduced reasonably, and the requestor can perform optimal actions accordingly by taking the long-term gain into consideration. To validate and evaluate the performance of the proposed model, we simulate various scenarios and conduct a comparison with other approaches. The numerical results show the effectiveness and feasibility of our proposed solution. We studied Trust in SO-IoT in the first two chapters, and in the following chapter, we will take a look at a concrete environment, i.e., IoV, to evaluate trust in inter-vehicle communications.

Trust of V2X messages in IoV

Chapter 4

Trust of V2X messages for Secure IoV Communication

4.1 Introduction

In this chapter, we evaluate Trust in IoV by examining V2X (Vehicle-to-Everything) messages in inter-vehicle communication. A number of V2X messages are standardized by the European Telecommunication Standardization Institute (ETSI) [79], such as CAM (Cooperative Awareness Message) and CPM (Collective Perception Message). Since road safety and traffic efficiency are on the basis of the assumption that correct and accurate V2V messages are shared, ensuring the trustworthiness of these V2X messages becomes an essential task in IoV (Internet of Vehicles) security [80, 3]. However, containing safety-related information makes V2X messages susceptible to malicious insider attacks from compromised vehicles after the PKI (Public Key Infrastructure) authentication step [81], such as Ghost Vehicles (GV) [82], passively or actively reaching a 'ghost' state in terms of communication, position, etc. By integrating CPS (Collective Perception Service) in the Veins simulator, our work aims to propose a trust assessment model in IoV against several types of CAM- and CPM-based GV to increase security. The simulation results provide a

preliminary analysis of the feasibility of the proposed model and show the effectiveness in terms of assessing V2X messages' trustworthiness.

4.2 Proposed Model addressing Trust in V2X messages

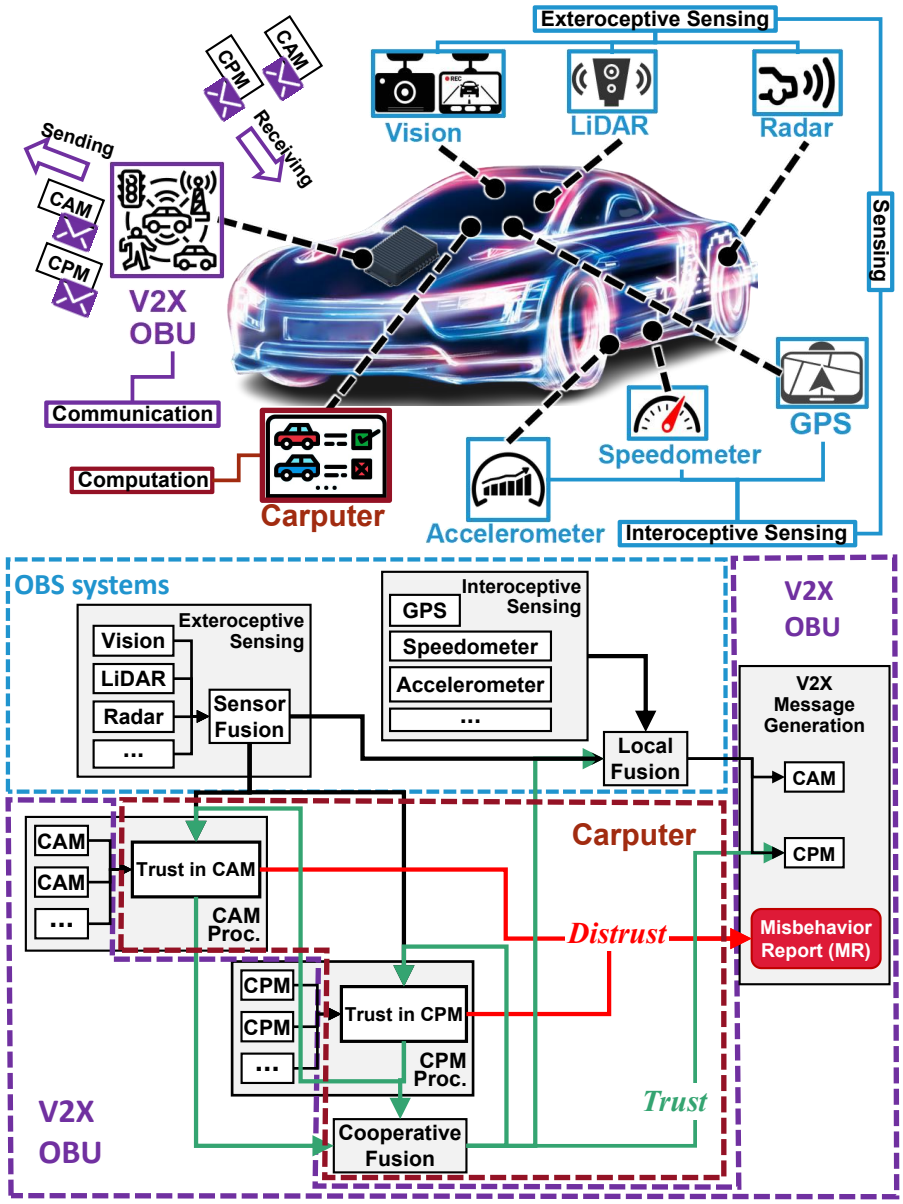


Figure 4.1: The functional flows showing how the trust model interacts with OBS and V2X OBU.

In IoV, sensing, communication, and computation capacities for vehicles are required, we colored these three in blue, purple, and brown in Fig. 4.1, respectively. The figure's upper part displays a vehicle in cooperative IoV with equipment, and the lower part illustrates functional flows within the vehicle and how the proposed trust model interacts with V2X OBU (On-Board Unit) and OBS (On-Board Sensor) [83, 84]. In IoV, **V2X OBU** supports the communication

between IoV entities, including both receiving and transmitting V2X messages: Vehicles or other entities periodically broadcast CAM to share their states and be aware of others through processing received CAM; Unlike CAM's 'I am here' manner, CPM is 'I see someone here' message to complement CAM; **OBS** in IoV consists of exteroceptive and interoceptive sides, where the former senses the surroundings and the latter monitors the vehicle's dynamics. Lastly, the **Carputer** refers to computing hardware in the vehicle, where the trust in CAM and in CPM will be investigated. We designed an extended cooperative scheme for CAM and CPM messages: the vehicle's sensing data will be counted to evaluate all incoming messages; Trustful CPM will be utilized to assess other incoming CPM. Once misbehavior of either CAM or CPM is detected, MR will be generated and sent to Misbehavior Authority (MA) as defined in [85], and thus fraudulent V2X messages will be rejected and marked.

4.3 Trust in CAM

Trust in CAM can be affected by numerous QoS (Quality of Service) factors: communication success rate, freshness of the message, etc. Since CAM is a multi-casting one-hop and one-way message standard, CAM-based communication is without request, reply, or forwarding operations [86]. It also means that transmission failure cannot be detected. As defined in the CAM standard, each vehicle can only passively receive CAM messages from others in a single hop. Moreover, the CAM message may be generated in an unstable manner due to the high-dynamic nature of IoV and the complex road traffic situation. Based on the above discussion, as shown in Fig. 4.2, we consider assessing the freshness of the message and the level of acquaintance to measure the trust in CAM.

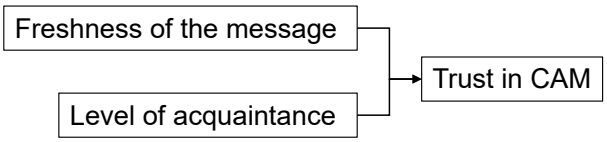


Figure 4.2: Composition of Trust in CAM

4.3.1 Freshness of the message p_1:

With the purpose of avoiding using outdated information, from the CAM receiver's point of view, the more recent the CAM is, the more the message can be trusted. In this sense, the exponential time decay model can be employed to weigh the CAM information regarding the message's timestamp. The weight for n^{th} CAM $w[n]$ from CAM sender i is:

$$w^i[n] = \rho^{t-t_n^i} \tag{4.1}$$

where $\rho \in]0,1[$ refers to the decay factor, which reflects the importance of the history, i.e., $\rho = 0.5$ indicates that the trust in the CAM drops by half every second, t is the current time and t_n^i is the timestamp of n^{th} CAM from the vehicle i. The computation of p_1 of a CAM sender i is defined as:

$$p_1^i = (1-\rho) \sum_{t_n^i \leq t} w^i[n] \tag{4.2}$$

where $w^i[n]$ is given in (4.1). As t goes to infinity, the sum in (4.2) converges to $\sum_{n=0}^{\infty} \rho^{nT^i} = \frac{1}{1-\rho^{T^i}}$, where T^i is the CAM transmission period of sender i. Since $\frac{1}{1-\rho^{T^i}} \leq \frac{1}{1-\rho}$. By giving the weight $(1-\rho)$, the value of p_1^i is always in the range of [0 1].

4.3.2 Level of acquaintance p_2:

Malicious attackers may try to refresh their trust in IoV by re-communicating by fabricating a new identity, which is one of the intelligent trust-related attacks named NCA [2]. To deal with this, newcomers should not be trusted as much as known ones, meaning the known vehicle's trust can be gained more easily than newcomers. Given this, the number of communications is utilized for differing 'known' and 'less-known' vehicles, and p_2 is defined as:

$$p_2^i = \rho^{\lambda \cdot (\frac{t}{T^i})^{-1}}, \quad \lambda \in R_+, \tag{4.3}$$

where t and T^i are the same as before, and λ is a scale factor, e.g., under the parameter setting ρ=0.5, λ=5, the 5^th (n=5) CAM outputs p_2=0.5, meaning that the level of acquaintance is average.

4.3.3 Total trust in CAM counting p_1 and p_2:

To take both p_1 and p_2 into computation, we consider that they are equally important for the trust in CAM:

$$T_c^i = (p_1^i * p_2^i)^{\frac{1}{2}} \tag{4.4}$$

To summarize, $p1$ value calculates the freshness of the message, and $p2$ value determines the level of acquaintance. In such a manner, the OOA attacker misbehaves within a fixed period by pausing sending CAM, or the NCA attacker re-communicates by faking its identity will be punished.

4.4 Implemented CPM structure

Before we explain the trust in CPM, we will present the integration of CPS into the Veins simulator, as CPS is incompletely supported in Veins. PO can be broadcast by vehicles via CPS,

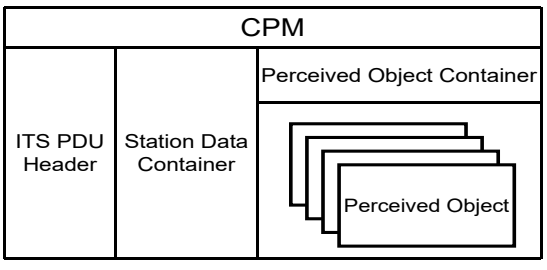


Figure 4.3: Structure of implemented CPM in Veins

which enhances local perception, and road safety can be thus improved [87]. In our work, CPM was taken into consideration for IoV communication. To achieve this, we first integrated CPM into Veins in the form of a message in OMNeT++. Previous V2X studies on standards of ETSI

are based on CAM and the corresponding C language standard library. We refactored CPM in C++ on the basis of the Veins-Inet subproject use case, bypassing encapsulation to enable more dynamic calling and debugging, as well as a more consistent message structure defined by the OMNeT++ framework. CPM will be sent in segments to simulate vehicles' sending capabilities and increase data processing flexibility.

As shown in Fig. 4.3, the implemented CPM structure is composed of: (1) an ITS PDU Header including the information of the protocol version, the message type, etc.; (2) The Station Data Container provides information containing the station type and the position of the CPM generator; And (3) a Perceived Object Container, which will be added in case that any road object has been perceived.

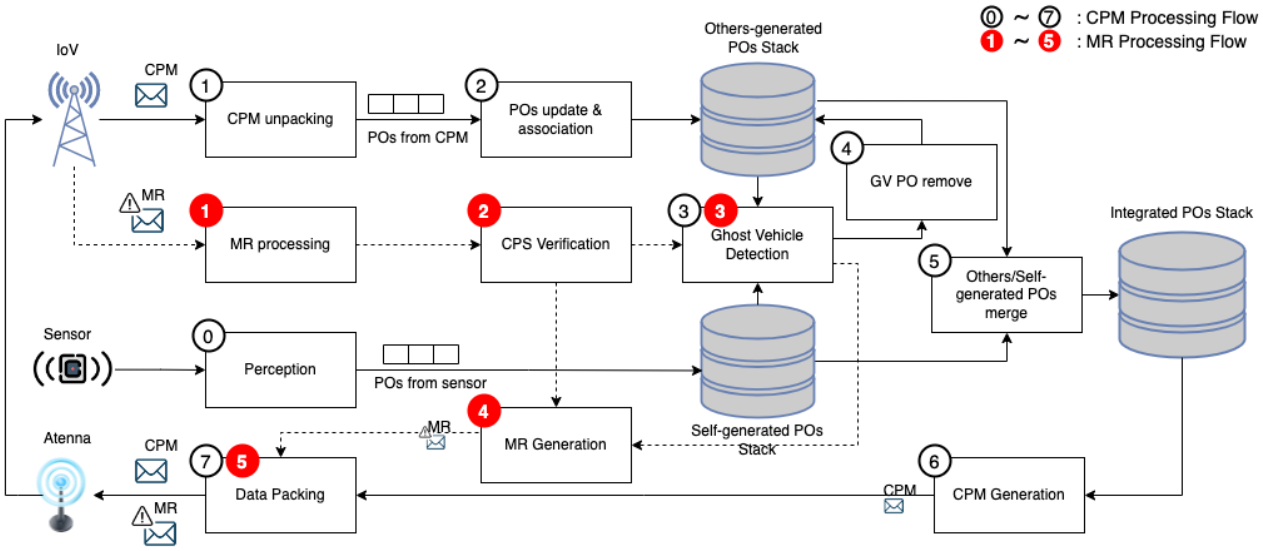


Figure 4.4: Pipeline of CPS application integrated in Veins Simulator

4.5 Trust in CPM

Figure 4.4 shows the pipeline of CPS application integrated into Veins simulator that is categorized into two principle processing flows, namely CPM and MR, numbered by two-color labels, respectively.

For any vehicle, sensors' data is regarded as the most credible source because of the first-hand information. We approximate the sensors' detection range as a circular area with a pre-configured radius to simulate the vehicle's perception capability.

As can be observed in Fig. 4.4, the source of POs can be either self-perceived or from incoming CPM, the latter is called others-generated in our work. After incoming CPM from other vehicles is unpacked (①), POs should be updated and associated by the receiver vehicle (②). For example, the receiver vehicle may receive a CPM in which one of the POs is itself, and thus there is no reason that this vehicle adds itself to the outgoing CPM. Both self-perceived and others-generated POs must be verified by GV detection process (③ & ❸). Similarly, the GV detection can be realized by either the incoming MR (❶ & ❷) or the receiver vehicle itself. When an MR informing an identified GV is broadcast, the receiver vehicle can directly forward this MR (❹) and remove the GV in POs (④). Or, if the vehicle detects the GV through its own perception capability, it will report this GV (❹). After that, the remaining self-perceived and

others-generated POs will be merged into the integrated POs Stack (⑤) and then be utilized to generate outgoing CPM (⑥). Finally, the outgoing CPM or MR will sent via the vehicle's antenna (⑦ & 5).

As in ③ & 3, the GV detection is mandatory for self-perceived and other-generated POs, this is also the reason that we separated them to represent different PO sources in Fig. 4.4.

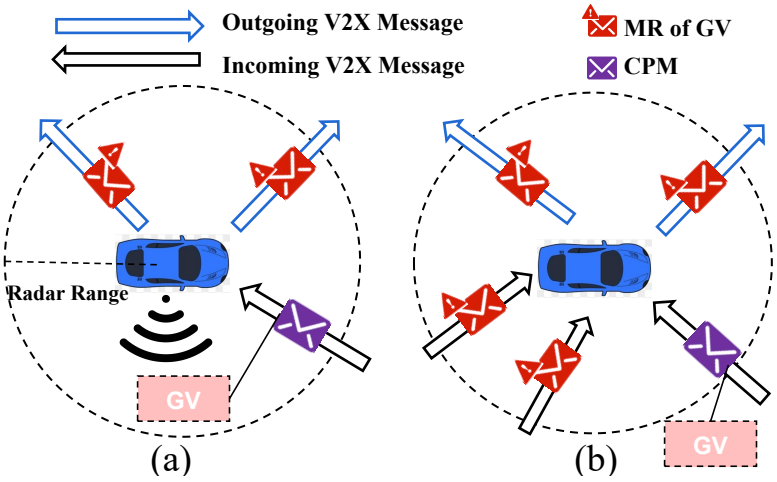


Figure 4.5: Two GV detection cases: in (a) or out (b) of the evaluator vehicle's perception range

Upon receiving an incoming CPM with a new PO, if this PO is in the self-perception range and can be detected, and the associated PO is searched in the other-generated POs stack, it will be regarded as a normal PO. In case the PO cannot be detected by the CPM receiver vehicle in its perception range, it will remove this PO as in ④ and include this PO as a GV in outgoing MR as demonstrated in Fig. 4.5(a). Or, when the PO is out of the CPM receiver vehicle's perception range, as shown in Fig. 4.5(b), the PO will be considered GV if the vehicle receives two or more MR indicating this PO is GV as discussed in [88]. In other words, in this case, the GV detection can only work with the aid of incoming MR from other vehicles.

4.6 Attack Model

CAM-based trust-related Attack Model: Two CAM-based attack types are considered in our work.

- OOA: The attacker vehicle switches its behavior between good and bad over time to mislead the trust evaluation. In our work, we consider the OOA attacker vehicle will misbehave by intentionally doubling its original communication frequency of CAM.
- NCA: The attacker vehicle fabricates a new identity to convey CAM with the purpose of refreshing its trust.

In the simulation, v2 is the CAM receiver, and thus, the trust evaluator and v0 will misbehave by launching the above attacks.

CPM-based GV Attack Model: We still fix v0 as the attacker broadcasting fake CPM of GV, and two other nodes are victims. The GV attack can be regarded as a specific form of Sybil attack, where fake-identity vehicles are created. CPM-based GV differs from CAM-based GV

in a way that the attacker generates CPM containing other GV (i.e., not the attacker itself via CAM). It should be noted that CPM-based GV has no physical counterpart. As illustrated in Table 4.1, we involve four different types of GV in our simulation, which have been studied as CAM-based GV in [89].

Table 4.1: GV Attack Parameters

GV Type	Parameters/Description
Constant	$x = 461.937, y = 414.526$
Constant offset	$\Delta x = -100, \Delta y = -50$
Random	Uniformly random in playground
Random offset	d uniformly random from [0,150] θ uniformly random from [0,2π] $\Delta x = d*\cos\theta, \Delta y = d*\sin\theta$

- **Constant**: The GV's position is fixed on the map.
- **Constant Offset**: The GV will appear at a Constant Offset from the attacker, like a follower.
- **Random**: The GV's position will be randomly generated on the map.
- **Random Offset**: The GV will randomly appear at any location within the range of the attacker's perception range.

4.7 Simulation Results

The simulation setup and implemented scenario will be presented first, and then the attack model will be detailed. After that, we analyze the performance of the proposed trust framework under CAM- and CPM-based malicious attacks. Lastly, we discuss the detection accuracy of CPM-based GV.

4.7.1 Simulation environment and traffic scenario considered

-Veins Simulator and our development

Veins is an open-source framework that is used for simulating communications and the interactions between vehicles in IoV [90]. It is based on two well-established simulators: OMNeT++ [91], an event-based data communication simulator, and SUMO, a road traffic simulator. Veins extends these two simulators mentioned above to provide a comprehensive simulation environment for both vehicular mobility and wireless communication. As CAM communication is already supported in Veins, we integrate CPS into Veins to enable CPM communication as described before. The simulator is extended from the example of veins_inet subproject. We summarize the simulation parameters in Table 4.2. For CAM-based GV attacks, we focus on analyzing the changes in trust values of OOA attacker, NCA attacker, and the normal vehicle to validate the

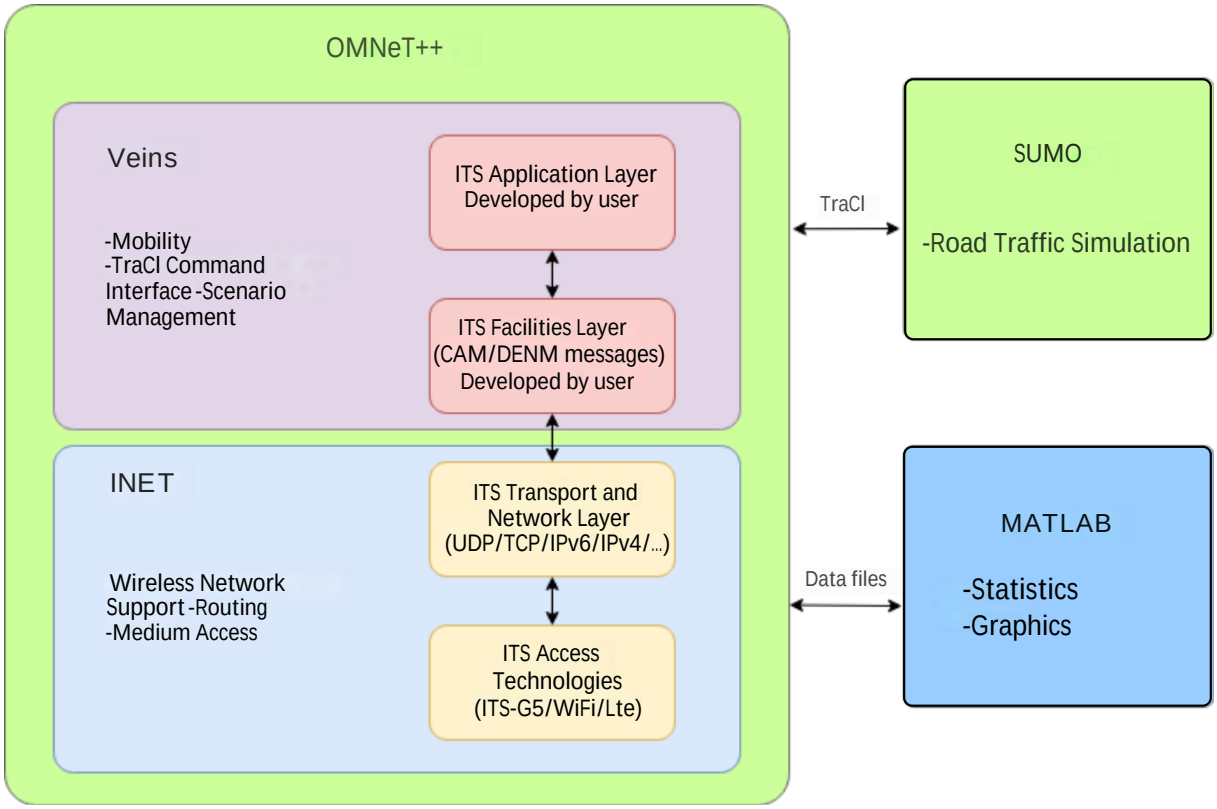


Figure 4.6: Developed architecture based on Veins simulator

proposed model's effectiveness. For CPM-based GV attacks, CPM-based GV is considered identified once the evaluator vehicle generates the MR, either by (a) or (b) in Fig 4.5. In addition. We also conduct a comparative analysis to assess the detection rate per CPM-based GV attack type.

-Traffic Scenario

The scenario considered is based on two main assumptions: Each vehicle can track the PO in the received CPM, and the MR can not be faked. Fig. 4.7 shows the scene around the largest intersection called Vélodrome in the center of the city Vandoeuvre-les-Nancy in France, as the urban traffic environment considered. A three-vehicle scenario is adopted in the simulation, where vehicle 2 (v2) is overtaking vehicle 0 (v0), and vehicle 1 (v1) is at a short distance in front of them. More precisely, v2 is an Emergency Vehicle with a higher speed, and other vehicles are of type City Car with a relatively lower speed. Circles in Fig. 4.7 (2D Visualizer) represent vehicles' perception ranges fixed at 150m.

4.7.2 Performance analysis of trust in CAM

Trust under OOA attack:

We can observe that the cooperative known vehicle reaches a much higher trust level than the OOA attacker one. The misbehaving of the uncooperative vehicle, i.e., OOA attacker, is reflected in a lower trust level as it intentionally doubles the CAM transmission frequency.

Table 4.2: Simulation parameter values

Parameter	Value	
Mobility	SUMO Vandoeuvre-lès-Nancy	
Update Interval	0.1s	
Radio Type	Ieee80211DimensionalRadio	
Radio Band	5.9GHz	
Radio Bandwidth	10MHz	
Transmit Power	80mW	
Vehicle Type	CityCar	EmergencyVehicle
Vehicle Speed	10 km/h	55 km/h
Perception Range	150 m	
CAM Broadcast Frequency CPM Broadcast Frequency	1Hz	
ρ λ	0.5	

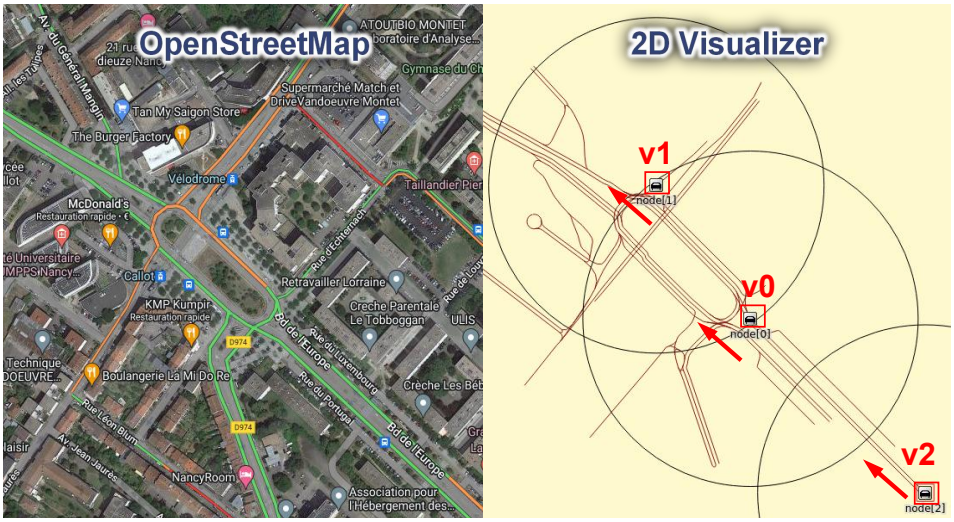


Figure 4.7: Considered Traffic Scenario

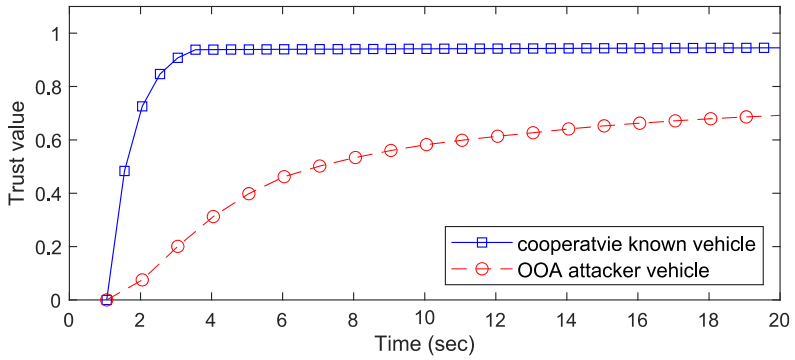


Figure 4.8: Changes in vehicles' trust values in the presence of OOA

Trust under NCA attack:

Differently, NCA attacker is considered at a relatively high trust level in the end, while its trust values increase more slowly than the known vehicle. This is because the newcomer vehicle lacks acquaintance of CAM messages, and thus, CAM from it will be considered less trustful.

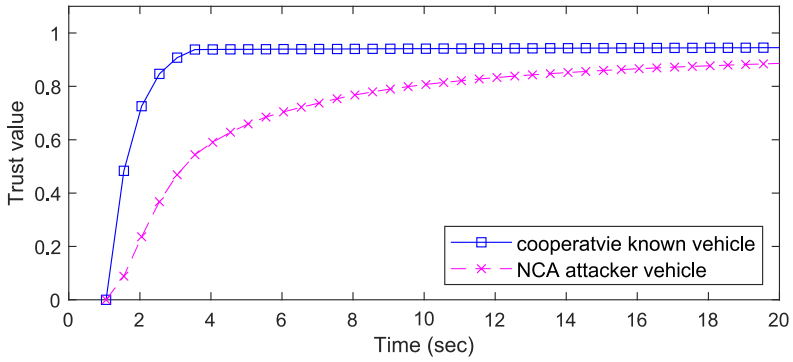


Figure 4.9: Changes in vehicles' trust values in the presence of NCA

Discussion:

As discussed in Section 4.3, we evaluate the freshness of the message and the level of acquaintance to measure the trust of the received CAM and the sender vehicle. The only optimal way to gain trust is to cooperate in transmitting CAM and remain known in IoV without faking the identity. For the MR generation, two thresholds are needed: 1) the number of received CAM messages and 2) the lowest acceptable trust value. In other words, the MR will be generated if the evaluator vehicle has received sufficient CAM messages and the trust value remains still less than the threshold. We note that the threshold can be dynamic depending on real-time traffic conditions instead of a predefined value [92].

4.7.3 CPM transmission and the evaluation of trust in CPM

For all simulation demonstrations, including CPM transmission, Constant, Constant Offset, Random, Random Offset GV, and related MR generation, please refer to our recorded videos[1].

CPM transmission:

As we can observe in Fig. 4.10, node0 is sending CPM, and node1 and node2 are receiving CPM. As node1 is perceived by node0, it has been included in node0's self-generated stack.

Constant GV detection:

Fig. 4.11 shows the GV is generated in node0's CPM with a pre-configured and fixed position (constant GV). Node2 will generate MR since node1111 (GV) is in node2's perception range but is not detected by node2.

[1] https://www.youtube.com/playlist?list=PLzIU1iYy4sJjPSz7HjvMLYme7z4D1E4KW

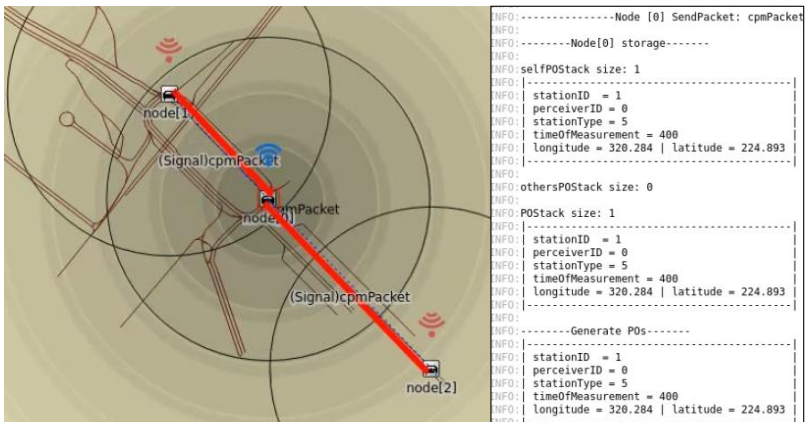


Figure 4.10: CPM Transmission Visualization

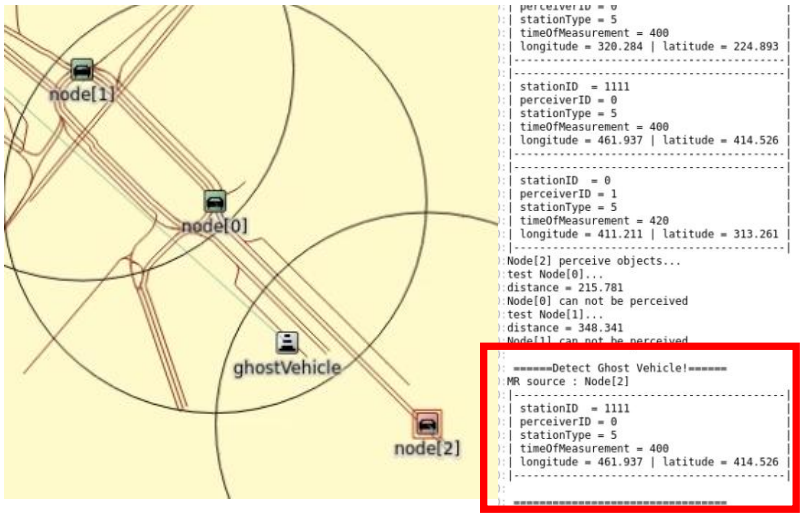


Figure 4.11: Constant GV and MR Generation

Constant Offset GV detection:

As depicted in Fig. 4.12, Constant Offset GV generated by node0 remains undetected for node1 even in node1's perception range, and thus node1 reports node1111 as GV in MR. In fact, the Constant Offset vehicle would move with the attacker node0, the capture of Veins simulator cannot provide such dynamics. For a comprehensive simulation visualization of Constant Offset GV and its MR generation, please refer to the video link given at the bottom.

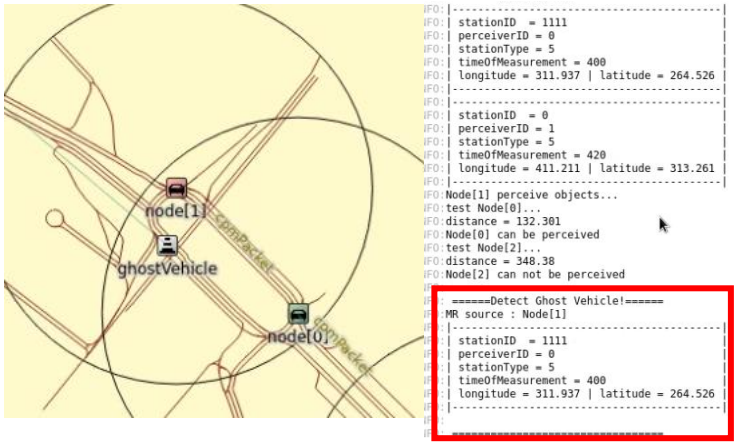


Figure 4.12: Constant Offset GV and MR Generation

Random GV detection:

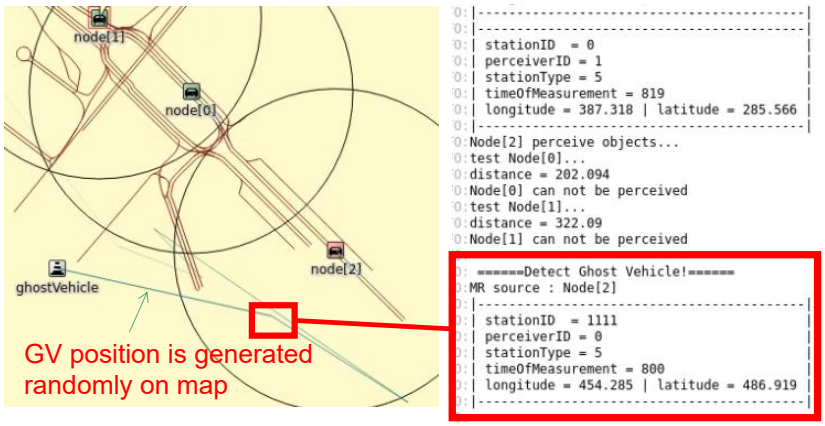


Figure 4.13: Random GV and MR Generation

For random GV, its position will be generated randomly on the map through node0's outgoing CPM. As can be observed in Fig. 4.13, the GV's position already changes several times. The MR generation of node2 in the figure occurred when the GV was in node2's perception range (the small red rectangle in the figure). On the other hand, none of the vehicles can ensure the GV detection when GV is out of all vehicles' perception ranges, which is exactly the case in the left part of Fig. 4.13.

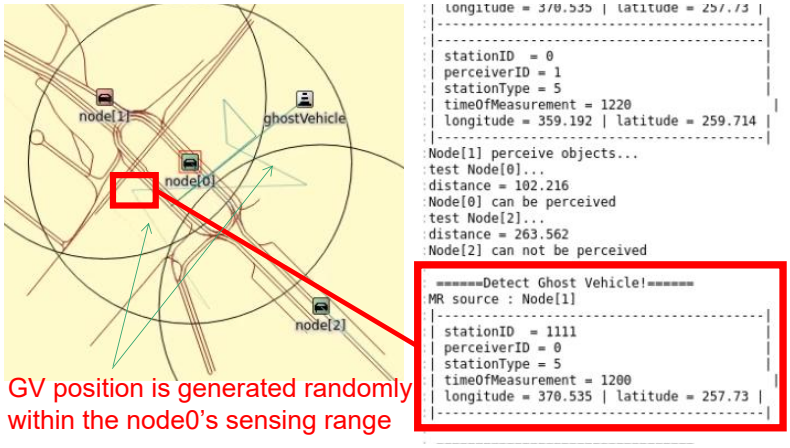


Figure 4.14: Random Offset GV and MR Generation

Random Offset GV detection:

Similar to random GV, Random Offset GV's position changes randomly but always in the attacker's perception range, i.e., node0's range. In some cases, the GV is too far away from the attacker and it becomes evident that the PO's information is faked in CPM as the CPM generator (attacker) cannot detect this PO out of its perception range. Fig. 4.14 demonstrates that GV's position changes randomly within node0's perception range. When it was in node1's perception range (the small red rectangle in the figure), one MR was generated by node1 to broadcast the identified GV in its received CPM.

Detection rate of CPM-based GV

Trust in CPM differs from Trust in CAM, since the latter is on the basis of a probabilistic value in the range of [0 1] to describe the trustworthiness of the CAM source, and the former is a policy-driven trust scheme, i.e., a binary question. For this reason, the detection accuracy of CPM-based GV should be discussed regarding four GV types.

We ran 10 times 30-second simulations to test the detection rate, in which the attacker vehicle sent 1 CPM of GV per second, and thus 30 CPM of GV in total. The results are illustrated in Table 4.3. It should also be noted that the GV may appear at a position where none of v1 and v2 can detect it, especially the Random GV one. Given this, while both v1 and v2 can generate MR if the GV is detected, the number of MR close to 30 is more or less satisfactory. This table shows that Constant, Constant Offset, and Random Offset detection rates are somehow acceptable, except for the Random GV, which remains at very low detection accuracy.

More than this, as can be seen in Fig. 4.15, it can be noticed that there are no error bars for the former two types of CPM-based, namely Constant and Constant Offset. On the other hand, the gaps in MR generation numbers of each simulation remain considerably different for the latter two types, namely Random and Random Offset CPM-based GV types. This is because simply the 'Random' GV's position changes over time, and the probability that they stay out of detector vehicles' perception range becomes larger.

Furthermore, this figure shows that the 'Offset' GV, either Constant Offset or Random Offset, remains more detectable than their original GV versions (Constant and Random). As the 'Offset' GV moves in a manner that follows one of the evaluator vehicles, it will be more likely to be in the detection range. The detection accuracy results are obtained by simulation of only two

Table 4.3: Number of MR generation under four CPM-based GV attacks

NO.	Constant	Constant Offset	Random	Random Offset
1	33	39	10	28
2	33	39	13	25
3	33	39	12	22
4	33	39	12	25
5	33	39	7	26
6	33	39	12	30
7	33	39	10	27
8	33	39	18	31
9	33	39	16	25
10	33	39	13	26

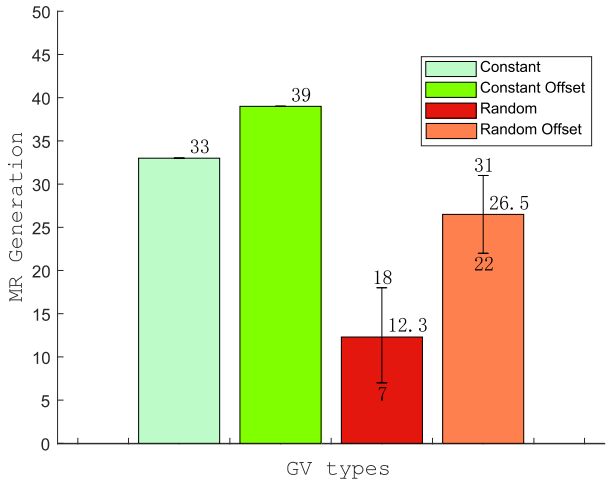


Figure 4.15: Comparison of detection rate of four CPM-based GV types

detector vehicles (honest CPS vehicles), and in this sense, we believe that as the number of CPS detectors increases, the detection success rate will grow significantly.

4.8 Conclusive remarks

Since CPS is rarely taken into consideration in existing works for IoV communication, we integrated CPS in Veins simulator to enable CPM communications between vehicles. Furthermore, we proposed a trust framework addressing two CAM-based trust-related attacks, namely OOA and NCA, and four CPM-based GV attacks, namely Constant, Constant Offset, Random, and Random Offset. A three-vehicle scenario simulation has been conducted under the above attacks to validate our proposed trust model. The simulation results show the effectiveness of the proposed model considering two CAM- and four CPM-based misbehavior kinds.

Since real-world traffic remains complex, the simulation scenarios and attack models will be explored and discussed in our future work through a larger-scale simulation involving more IoV entities.

Conclusion

Chapter 5

Conclusion and Perspectives

5.1 Conclusive remarks for the current research

-Group-Individual and Inter-Group Trust in SO-IoT

The intra-community TM is established on four phases: access control, selection, service evaluation, and node classification. In each phase, we designed a number of trust values to improve the service-oriented activities and address the security issues, i.e., the countermeasures against attacks on services, namely NCA, OOA, CBA, SBA, BMA, BSA, and SPA. Since there is no service provision or rating at the inter-community level, we studied the service migration case (node moving to a new community) to estimate the trustworthiness between communities, and we also investigated trust between communities from initial evaluation to community classification. Through intensive simulations, we have verified that the proposed model is adequate and accurate for dealing with trust issues in community-driven IoT at both intra- and inter-community levels. On the other hand, due to the nature of our model, the trustworthiness within the community is monitored in a locally centralized manner. This means that the manager is regarded as fully trusted, and as a consequence, the proposed trust framework is still locally facing the challenges of a single failure point issue. Moreover, addressing attacks on communication/message and attacks on service are not conflicting. Since we focus on the resilience against the latter attack type in this work, we assumed that the former attack type such as DoS is addressed by other security-related communication schemes, and it will be interesting to analyze the impact of attacks on communication for our future work. For putting forward our proposed model into a real IoT-based environment, we implement our model functions within real-world devices by using ROS 2, including SR and SP, and community managers. The first results of implementation show the feasibility of our proposed framework in a real IoT system and the resilience against two attacks, namely OOA and BMA. We can also observe that the negative perturbation from the malicious SR is much more harmful than the malicious SP side.

-Inter-Individual Trust in SO-IoT

To overcome several limitations of the current works and address misbehavior in SO-IoT in terms of Inter-Individual Trust, we presented SBG where the service requestor and the worker are regarded as heterogeneous with an asymmetric payoff matrix, and the selfish action and the malicious action can be distinguished. More importantly, the complex behavioral schemes of the worker, i.e., strategies, are considered in the SBG by applying the BAR threat model. We also involved these strategies in the simulation to assess the performance of the proposed model. From the observation in Section 3.10 through the simulation results of the performance evaluation of the proposed model under various scenarios and the comparative analysis with the other two approaches from the literature, we can notice that workers' behavioral types can be deduced accurately, and the requestor can perform optimal action by maximizing its long-term payoff. Therefore, the Altruistic and Rational worker performing cooperate can receive a positive payoff and the Byzantine worker's action can be tracked and punished. Through a comparison with two other approaches based on the most complex worker's strategy and history setting (RS worker and favorable history), our proposed model using HBA to specify the requestor's strategies outperforms other approaches in the average payoff obtained, and the numerical results also show the resilience of our proposed model when dealing with the malicious worker whose history has been refreshed. Furthermore, as the worker also has the purpose of gaining a higher payoff, the optimal strategy when facing a requestor controlled by HBA is to perform cooperate as much as possible, even though the successful attack allows the malicious worker to obtain the maximal gain in our designed payoff matrix. This also signifies that the cooperativeness between the requestor and the worker is encouraged in our proposed model.

-Trust in IoV

As CPS is rarely considered in existing works and there was no implementation of CPM in the popular Veins simulator, in this work, we integrated CPS in Veins, enabling inter-vehicle CPM communications. Furthermore, we proposed a trust framework addressing two CAM-based GV attacks, namely OOA and NCA, and four CPM-based GV attacks, namely Constant, Constant Offset, Random, and Random Offset. A three-vehicle scenario simulation has been conducted to provide a preliminary analysis of the feasibility of the proposed model and show the effectiveness in terms of assessing V2X messages' trustworthiness.

5.2 Outlooks for future research

5.2.1 Extensions and improvements following the thesis

-Role-based Trust evaluation in IoT communities

As presented in Chapter 2, the main computation of role-based trust evaluation is calculating trust values in (2.9), (2.10), (2.12), and (2.17). For a single service evaluation, the computation complexity of calculating trust values is $\mathcal{O}(g(N))$, where N is the number of ratings, showing the proposed computation scheme remains efficient as g is a linear function with respect to N. Based on this, future works can be extended by designing a more effective trust computation scheme. Furthermore, assuming that one rating can be scheduled to transmit within t, the data collection complexity for a single service evaluation is $\Theta(tN)$. It should be noted that the evaluation frequency cannot exceed $\frac{1}{t}$. Taking our implementation as an example, the image transmission is fixed at every 500ms, meaning the manager evaluates nodes' trust with the same frequency.

Once t is greater than 500ms, a delay between the evaluation and decision-making appears, and consequently, the evaluation scheme becomes no longer real-time. With this lesson learned from our implementation experience, finding the algorithm comforting the evaluation frequency and data collection will improve the current work.

-Self-organizing trust evaluation

In contrast to the trust model described in Chapter 2, which enables managers to evaluate trust within and between communities, the Inter-Individual trust model discussed in Chapter 3 employs a decentralized approach, i.e., a self-organizing trust evaluation. This method operates under the premise that every individual possesses the ability to carry out the required verification and calculation. Real-world IoT systems frequently encounter limitations in device capabilities and resources. Consequently, it is necessary to conduct a complexity and cost analysis while implementing a self-organizing trust evaluation. Specifically, when a device lacks the necessary processing capacity, it is important to identify an alternate solution that enables the device to engage in service activities while maintaining trust management at a minimal expense.

-Environmental effects on trust evaluation of V2X messages

Due to the wireless nature of IoV communication, it is subject to interference from various environmental effects such as weather conditions, communication perturbation from other IoV entities' signals, and physical obstructions like buildings and infrastructure that hinder signal propagation. In Chapter 4, such environmental effects are not considered, while the probability of occurring errors in IoV communication is expected to be low in ideal circumstances, it is not entirely nonexistent. Therefore, studying the influences brought by traffic environmental effects will be considered in our future work.

-Larger-scale and more complex scenario for simulation and implementation

For **Group-Individual and Inter-Group Trust in SO-IoT**, we are interested in testing the proposed trust model more comprehensively by extending the size of the system with more IoT nodes to investigate the scalability of the proposed model and also apply it to a more mobile scenario. Despite the implementation, we only involve 3 SP and 3 SR by using ROS 2, and the current work can be extended by involving more real-world devices. For **Inter-Individual Trust in SO-IoT**, as this work is only validated through simulation results, one direction to improve the current research will be conducting the implementation with IoT devices in order to test the proposed model. For **Trust in IoV**, since the simulation by using the Veins simulator is based on a three-vehicle scenario, with the proposed trust model integrating the CPS component in hand, our future work will be simulating larger-scale IoV scenarios involving more entities. Besides, implementing real OBU and RSU cards such as neoGLS (`https://www.neogls.com/`) will significantly support validating the effectiveness and feasibility of the proposed trust model.

5.2.2 Open topic in SO-IoT/IoT trust management research

-Coexistence of different Trust architectures

As we stated in Section 1.2, one of the necessary components of the TM process is trust architecture. Most existing studies consider a single type of trust architecture, i.e., fully distributed or centralized. Some work developed hybrid ones by combining them. Due to the scalability nature

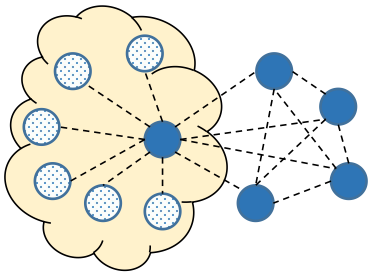


Figure 5.1: Centralized + Distributed Trust architecture

of IoT systems, one issue that remains to be addressed is the coexistence of different trust architectures. For example, as illustrated in Fig. 5.1, a number of IoT nodes are employed within the community for a dedicated mission and supervised by a master node, in case of need, the master node can interact with other nodes out of the community. The master node monitors the trust within the community in a centralized manner and assesses other nodes out of the community by itself. In such a scenario, how the centralized and distributed Trust architectures can co-exist and coordinate with each other becomes a problem. The coexistence of different trust architectures poses interoperability challenges, incompatibilities in trust formats and computation exacerbate interoperability woes, limiting effective collaboration. Additionally, varying trust inference levels further complicate co-existence, raising concerns about the reliability of each trust architecture, Overcoming these issues necessitates standardized frameworks and interoperability trust computation to facilitate harmonious coexistence among diverse trust architectures. However, the standardization raises another challenge.

-The way to the standardization of Trust in IoT

One of the difficulties of standardization is the definition of trust in IoT security since the concept of trust is derived from sociological research. For instance, when trust management demands qualifying and quantifying trust, problems such as whether trust values should be computed continuously or discretely arise. Another obstacle comes from the nature of IoT, e.g., the heterogeneity of IoT nodes (different functionalities) and networks (e.g., diverse protocols), leading us to adopt different computational approaches and architectural designs in different application contexts. More than this, We notice that existing works of trust in IoT systems considered model validation through simulation experiments, either based on a simulator of their own design or using a selected dataset. Some even adopt so-called confidential datasets, making reproducing and studying their work impossible. Bridging the gap between theoretical models and real implementation requires iterative refinement and adaptation to overcome unforeseen issues, meaning that we somehow have to necessitate compromises and adjustments to theoretical models, this is also what we learned from our implementation experience with ROS 2. Unfortunately, to the best of our knowledge, existing models have no experience in practicing using real-world IoT devices. All the abovementioned constitute impediments to standardizing trust management in IoT.

-Mix-type attacks and related countermeasures

On the one hand, in our work, we design related countermeasures against trust-related attacks in Chapters 2 and 4. In Chapter 3, strategic malicious behaviors are also addressed by the HBA algorithm. On the other hand, while the numerical results demonstrate that every single attack

can be detected, a very interesting question may arise about whether our trust management systems remain effective when attacks are mixed. Of course, the first time, we have to study the possibility of whether these attacks can be mixed or not and then study if it is still resilient when facing mix-type attacks. In the future, it will be very interesting to investigate the role of these attacks in the IoT in depth. Also, the role of phishing attacks [95] can be investigated deeply.

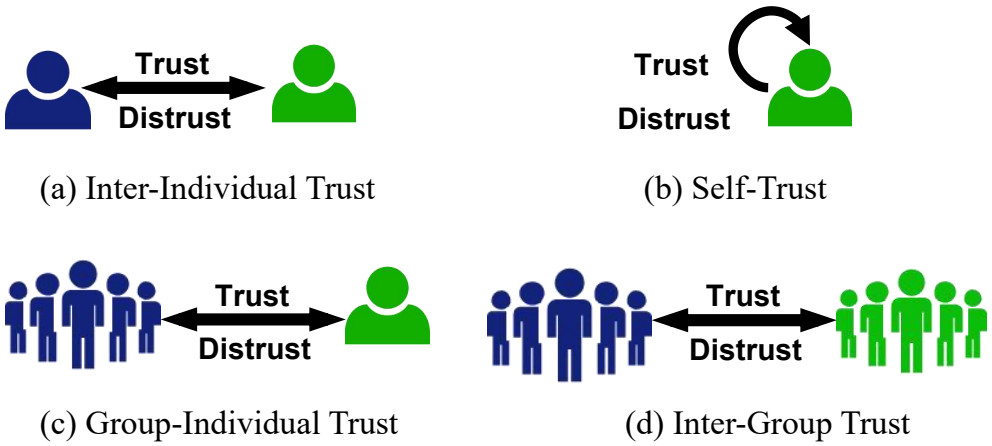
-ML- and blockchain-based solutions

Some trust-related works consider ML-based solutions [93]. On the one hand, ML techniques are extensively utilized in IoT systems because of their robust predictive analytics capabilities. On the other hand, ML requires a certain amount of data for effective training, which can be costly and time-consuming to acquire, and it is susceptible to biased data, leading to inaccurate outcomes especially since IoT users often hold their own preferences and even biases. For these reasons, ML is often used in dedicated application scenarios. Another emerging paradigm often involved in solving trust issues in IoT is Blockchain. Blockchain has several advantages as it is a distributed, unchangeable, and shareable database ledger that stores the registry of transactions and assets across a peer-to-peer (P2P) network. Nevertheless, blockchain-based methods remain still vulnerable [94]. For example, the private keys can be used to compromise the blockchain accounts. In this regard, it is crucial to have further research on developing efficient strategies to ensure the privacy of transactions.

Résumé étendu en Français

-Introduction générale du contexte

En sciences sociales, la confiance est généralement définie comme l'état psychologique de la personne qui fait confiance, qui s'attend à des résultats positifs dans l'action du fiduciaire, bien que ce dernier ne soit pas sous le contrôle de la personne qui fait confiance. En outre, il convient de noter que plusieurs synonymes associés à la confiance sont souvent employés comme alternatives : confiance, croyance, crédibilité, fiabilité, etc. Comprendre la confiance dans les sciences sociales permet de mieux comprendre la dynamique des relations et des structures entre les personnes qui accordent leur confiance et celles qui la reçoivent, ainsi que les mécanismes qui facilitent leur coopération et leur collaboration. Les chercheurs de toutes les disciplines, y compris l'informatique, étudient la confiance afin d'utiliser de manière exhaustive les utilités de la confiance qui sous-tendent les systèmes sécurisés composés d'agents de confiance et d'agents fiduciaires fiables.



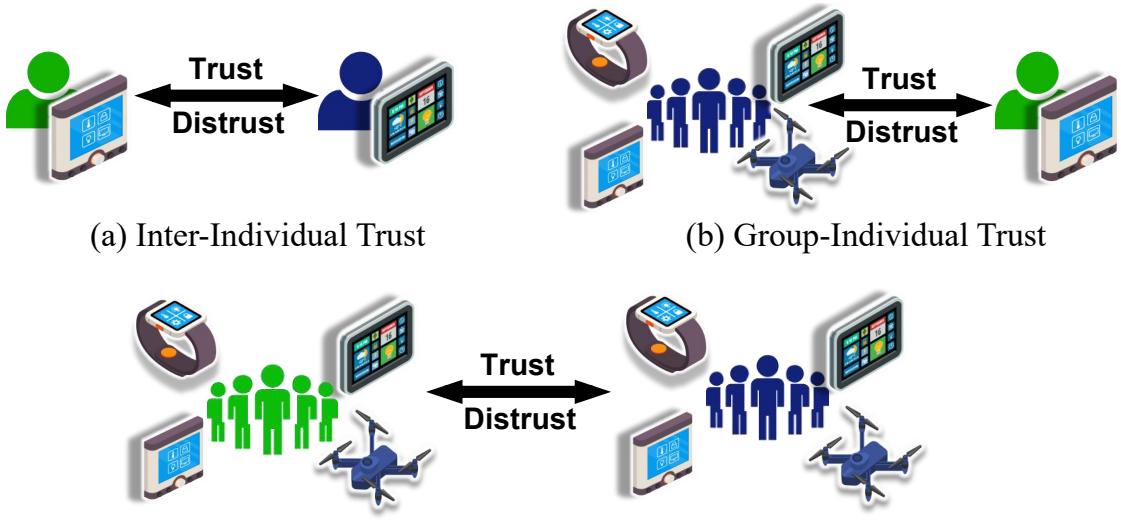
(a) Inter-Individual Trust (b) Self-Trust

(c) Group-Individual Trust (d) Inter-Group Trust

Figure 1 : Confiance dans les sciences sociales en fonction des "corrélateurs d'action".

La figure 1 montre que la confiance dans les sciences sociales est un concept à multiples facettes englobant diverses dimensions qui définissent les relations interpersonnelles et collectives. La confiance interindividuelle fait référence à la confiance que les individus s'accordent les uns aux autres et constitue le fondement des relations personnelles. La confiance en soi implique qu'un individu se fie à ses propres jugements et capacités. La confiance intergroupe étudie la dynamique de la confiance entre différents groupes sociaux, favorisant la coopération et la collaboration. La confiance entre groupes et individus explore la relation réciproque entre un individu et une entité sociale plus large, telle qu'une communauté ou une organisation. Chaque type de confiance joue un rôle essentiel dans l'élaboration des interactions sociales, la coopération et la création de liens

sociaux. La compréhension et l'analyse de ces dimensions de la confiance contribuent à une meilleure compréhension du comportement humain, des structures sociétales et des mécanismes qui sous-tendent les communautés fonctionnelles. La confiance, sous ses diverses formes, sert de pivot à la cohésion et à la fonctionnalité des systèmes sociaux, ce qui en fait un thème important dans le domaine des sciences sociales [14].



(a) Inter-Individual Trust (b) Group-Individual Trust

(c) Inter-Group Trust

Figure 2 : Confiance dans la sécurité de l'IdO

Comme le montre la figure 2, la confiance dans la sécurité de l'IdO implique diverses dimensions cruciales pour garantir la fiabilité et l'intégrité des dispositifs et systèmes interconnectés [18]. La confiance interindividuelle concerne la confiance établie entre les entités individuelles au sein du réseau IdO, soulignant la nécessité d'une communication sûre et digne de confiance. La confiance intergroupe étend ce concept aux relations entre différentes entités ou groupes d'appareils, favorisant la collaboration et la fiabilité entre divers composants. La confiance groupe-individu explore la dynamique entre un appareil ou une entité unique et l'écosystème IdO dans son ensemble, en soulignant la nécessité pour les composants individuels de fonctionner en toute sécurité au sein du réseau collectif. Ces types de confiance sont essentiels pour relever les défis en matière de sécurité, car la confiance est primordiale pour une communication efficace, le partage des données et la prise de décision en collaboration dans l'environnement IdO. Une compréhension et une mise en œuvre complètes de ces dimensions de la confiance sont essentielles pour renforcer la posture de sécurité des systèmes IdO, en veillant à ce que les dispositifs puissent interagir et échanger des informations d'une manière fiable et sécurisée. La confiance dans la sécurité de l'IdO, sous ses diverses formes, joue un rôle essentiel dans l'établissement des fondements de systèmes interconnectés résilients et sûrs. Le concept de gestion de la confiance (TM : Trust Management) en informatique a été introduit pour la première fois en 1996 [19], où les auteurs ont précisé que *"Trust management provides a unified approach for specifying and interpreting security policies, credentials, and relationships."* La gestion de la confiance est apparue comme l'un des moyens les plus prometteurs de garantir la continuité des performances d'un système sur lequel on peut compter. L'importance de la TM a été évaluée en relation avec l'IdO (Internet des objets) selon diverses dimensions, notamment la gestion de services fiables, le contrôle d'accès et l'optimisation de la stratégie. La TM contribue à réduire le degré d'incertitude et de risque

concernant divers services de l'IdO [20, 21, 22].

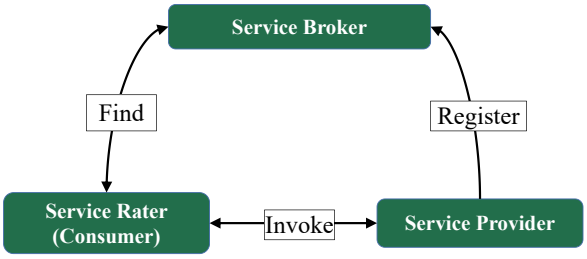


Figure 3 : SOA

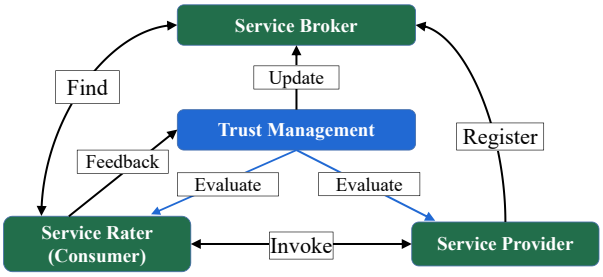


Figure 4 : Gestion de Confiance basée sur SOA

Dans l'IdO orienté services (SO-IoT : Service-Oriented IoT) [23, 24], les nœuds IdO intelligents peuvent participer de manière collaborative à des services IdO complexes sur la base de l'architecture illustrée à la figure 3. Comme on peut le voir, cette architecture se compose de trois éléments fondamentaux : le courtier de services, le consommateur de services et le fournisseur de services (SP : Service Provider). Le fournisseur de services publie ses services dans le référentiel du courtier de services, puis le consommateur de services découvre et enfin invoque les services. En résumé, les trois activités cruciales de l'architecture SOA sont l'enregistrement, la composition et la fourniture de services [25]. En intégrant la TM dans la SOA (Service-Oriented Architecture)[26], la figure 4 montre un modèle de TM basé sur la SOA pour l'IdO : Le consommateur de services devient un évaluateur de services (SR : Service Rate) lorsqu'il envoie son retour d'information à l'entité de TM après la fourniture du service, puis l'évaluateur de services et le fournisseur de services seront tous deux évalués par la TM. Ensuite, la TM peut aider les courtiers en services à prendre des décisions sur la base des résultats de la classification des nœuds et des valeurs de confiance, par exemple en vérifiant les types de services disponibles et en supprimant les nœuds qui fonctionnent mal ou les attaquants malveillants. Enfin, les informations relatives aux services seront mises à jour par le courtier en services. En effet, les mesures sont complexes pour l'évaluation des services, par exemple, un fournisseur de services qui fournit un service X médiocre peut être excellent pour le service Y. Par conséquent, la sécurisation d'un IdO basé sur les services exige que la TM interagisse avec les activités de service susmentionnées pour attribuer aux nœuds et aux services des valeurs de confiance exactes.

Le système SO-IoT fournit un modèle d'interaction qui met en œuvre des services pour améliorer les applications IdO actuelles. Afin d'étudier les différentes confiances en fonctions de "corrélateurs d'action", plus précisement, les confiances d'Inter-Individual, Group-Individual, et Inter-Group, cette thèse tente de répondre à la question de savoir comment évaluer et mieux évaluer la confiance dans les systèmes SO-IoT, et elle se compose trois parties principales, comme le montre la figure 1 : i) la confiance entre groupes et la confiance groupeindividu dans le SO-IoT; ii) la confiance interindividuelle dans le SO-IoT; et iii) la confiance dans l'IoV en évaluant messages V2X.

-Contributions

Confiance groupe-individu et entre les groupes dans SO-IoT

Comme nous l'avons discuté précédemment, les systèmes SO-IOT sont souvent constitués de groupes ou de communautés ayant leurs propres intérêts ou fonctionnalités. Dans ce contexte,

l'évaluation de la confiance de chaque individu au sein du groupe est nécessaire pour maintenir le niveau de confiance intra-groupe (GroupeIndividual Trust) et, ce qui est tout aussi important, la confiance entre les groupes (Inter-Group Trust) devrait également être évaluée pour prévenir les effets négatifs causés par des groupes/communautés dysfonctionnels ou malveillants. Toutefois, la plupart des recherches existantes n'ont porté que sur quelques attaques de services telles que OOA, BMA et BSA. En outre, ces modèles ne s'adaptent pas à la SOA en raison de l'absence de discussion sur les activités de service dans SO-IoT, en particulier, les rôles des nœuds ne sont pas pris en considération, c'est-à-dire le fournisseur de services et l'évaluateur de services. Plus important encore, certains d'entre eux sont soit entièrement distribués, soit entièrement centralisés, ce qui remet en question leur pertinence, c'est-à-dire que la confiance entre les groupes est absente. Pour surmonter ces limitations, nous avons conçu une architecture hybride pour soutenir et améliorer la confiance groupe-individu et intergroupe dans SO-IoT.

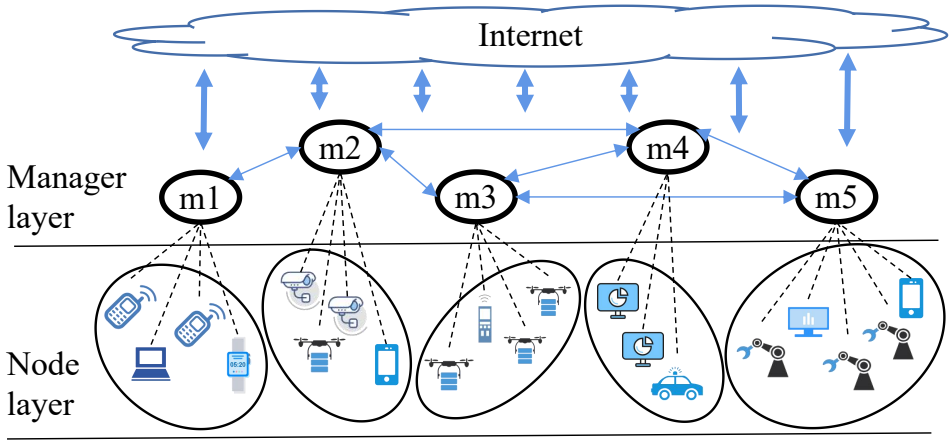


Figure 5 : Illustration de systèmes IdO/SO-IoT communautaires composés de nœuds intracommunautaires et de managers pour chaque communauté

La figure 5 illustre l'architecture du cadre de confiance proposé à deux niveaux : niveaux intra-communautaire et inter-communautaire, représentant respectivement la confiance groupe-individu et la confiance intergroupe. Les dispositifs intelligents sont assemblés au niveau intra-communautaire, où les nœuds d'une même communauté peuvent participer à des missions de coopération pour interagir dans un environnement multiservice. Chaque manager de confiance est chargé de la TM intracommunautaire en tant qu'entité responsable locale. En outre, les managers sont mis en réseau de sortes que la communication et la TM au niveau intercommunautaire sont distribuées. En particulier, la politique de contrôle d'accès (AC) de chaque manager de communauté n'est pas identique, car les différentes communautés peuvent avoir des préférences diverses en termes de service. Par conséquent, les nœuds nouveaux arrivants seront examinés de manière beaucoup plus stricte dans les communautés ayant des exigences spécifiques. Dans cette partie, premièrement, la TM intracommunautaire permet aux nœuds SO-IoT d'une même communauté d'être surveillés de manière dynamique en fonction de leurs rôles de service, à savoir SP et SR. Deuxièmement, la TM intercommunautaire examine la confiance entre différentes communautés en termes de coopération. Le modèle proposé a été simulé dans le cadre de diverses attaques de service. Les résultats numériques montrent l'efficacité de l'évaluation de la fiabilité intra- et inter-communautaire. En outre, les résultats préliminaires de la mise en œuvre démontrent la faisabilité du modèle proposé et le valident en partie dans la pratique. Cette partie se concentre sur la confiance intra- et intergroupe, et dans la partie suivante, nous avons étudié

l'évaluation de la confiance interindividuelle, où nous proposons un modèle théorique de jeu pour traiter le mauvais comportement des nœuds.

Confiance interindividuelle dans SO-IoT

Pour modéliser les interactions entre les nœuds dans les systèmes SO-IoT dans le contexte de la confiance interindividuelle (Inter-Individual Trust), nous considérons le scénario du processus de service décrit dans [68], où le processus de service se compose de quatre étapes principales : i) le nœud (qui est évalué, également appelé fournisseur de services ou travailleur) lance la proposition de tâche une fois la communication établie avec le nœud évaluateur (demandeur de service) ; ii) le nœud travailleur sera recruté, puis se verra attribuer la tâche ; iii) une fois la tâche achevée, le nœud évaluateur envoie l'incitation ; 4) Une fois que le nœud travailleur est informé de la réception de l'incitation, les données liées au service seront libérées. Dans ce contexte, nous concevons un jeu SBG pour modéliser les interactions entre le nœud évaluateur et les nœuds travailleurs de manière appropriée, où les nœuds travailleurs effectuent des actions de manière indépendante, c'est-à-dire qu'il n'y a pas d'inter-affection entre eux. La théorie des jeux a été prise en considération pour modéliser la confiance interindividuelle dans les systèmes SO-IoT. Toutefois, les recherches existantes montrent qu'il reste encore plusieurs limitations à résoudre. Premièrement, la majorité des solutions de gestion de la confiance fondées sur la théorie des jeux se concentrent sur un ensemble simple d'actions (par exemple, coopérer/défaillance), de sorte que les actions du demandeur et des travailleurs sont homogènes, ce qui ne correspond pas à la réalité de SO-IoT. Deuxièmement, le modèle comportemental stratégique complexe des attaquants malveillants n'est pas pris en considération, ce qui signifie que l'attaquant reste indétectable lorsqu'il modifie ses actions pour tromper le système d'évaluation. Troisièmement, la distinction entre le comportement intéressé et le mauvais comportement n'est pas faite, le premier provenant de nœuds non-malveillants et causant moins de dommages. Enfin, la coopération entre les nœuds SO-IoT, c'est-à-dire le demandeur (évaluateur) et les travailleurs, n'est pas suffisamment examinée. Plus précisément, comment leur coopération peut être encouragée à l'aide du modèle de jeu. Dans ce contexte, nous proposons un modèle théorique de jeu utilisant le SBG pour surmonter les limitations. Nous nous concentrons sur l'évaluation des interactions entre le nœud évaluateur et les nœuds évalués dans un système SO-IoT distribué.

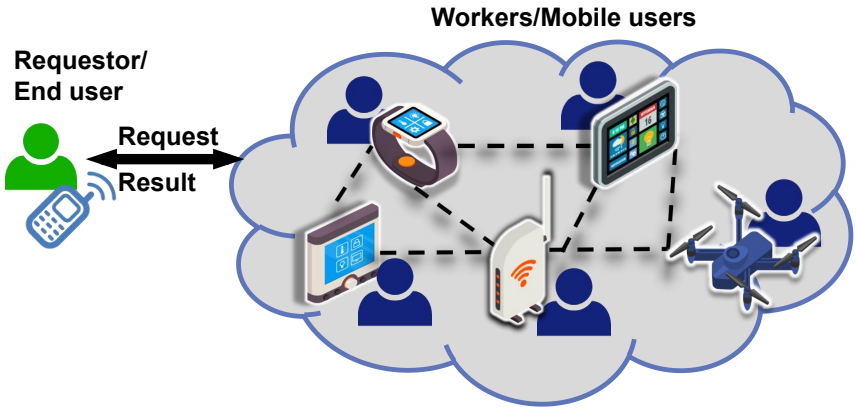


Figure 6 : Architecture considérée pour évaluer la confiance interindividuelle dans SO-IoT

La figure 6 illustre le nœud évaluateur de tâches et les nœuds travailleurs dans un système SO-IoT distribué. Dans cette partie, nous avons proposé un jeu stochastique bayésien (SBG)

pour traiter le comportement erroné basé sur l'altruisme byzantin et rationnel (BAR) dans l'évaluation de la confiance interindividuelle, où les types de comportement des travailleurs du service peuvent être déduits raisonnablement, et le demandeur peut effectuer des actions optimales en conséquence en prenant le gain à long terme en considération. Pour valider et évaluer les performances du modèle proposé, nous avons simulé différents scénarios et effectué une comparaison avec d'autres approches. Les résultats numériques ont montré l'efficacité et la faisabilité de la solution proposée.

Confiance dans IoV en évaluant les messages V2X

Un certain nombre de messages V2X (Vehicle-to-Everything) sont normalisés par l'Institut européen de normalisation des télécommunications (ETSI) [79], tels que CAM (Cooperative Awareness Message) et CPM (Collective Perception Message). Étant donné que la sécurité routière et l'efficacité du trafic reposent sur l'hypothèse que des messages V2V corrects et précis sont partagés, garantir la fiabilité de ces messages V2X devient une tâche essentielle dans la sécurité de l'IoV (Internet des véhicules). Toutefois, le fait de contenir des informations relatives à la sécurité rend les messages V2X susceptibles de faire l'objet d'attaques internes malveillantes de la part de véhicules compromis après l'étape d'authentification de l'infrastructure à clé publique (PKI), tels que les véhicules fantômes (Ghost Vehicles = GV), atteignant passivement ou activement un état "fantôme" en termes de communication, de position, etc.

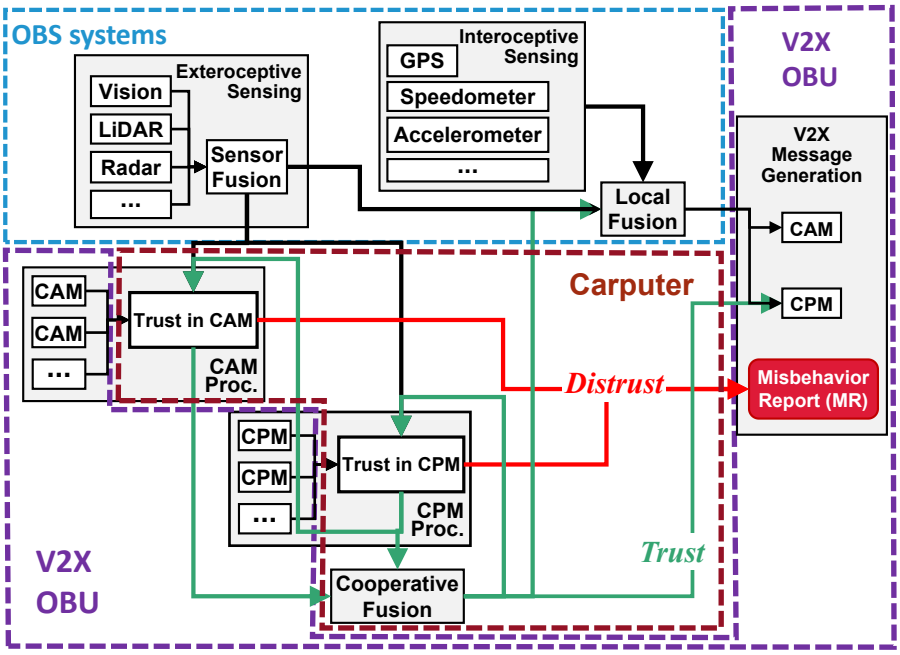


Figure 7 : Les flux fonctionnels qui démontrent comment le modèle de confiance interagit avec l'OBS (On-Board Sensor) et l'OBU (On-Borad Unit) de V2X messages.

La figure 7 montre le diagramme de flux fonctionnel traitant les messages CAM et CPM. En intégrant CPS (Collective Perception Service) dans le simulateur Veins, notre travail vise à proposer un modèle d'évaluation de la confiance dans l'IoV contre plusieurs types de GV basés sur CAM et CPM afin d'accroître la sécurité. Les résultats de la simulation fournissent une analyse préliminaire de la faisabilité du modèle proposé et montrent son efficacité en termes d'évaluation de la fiabilité des messages V2X. Étant donné que la CPS est rarement prise en considération dans

les travaux existants sur la communication IoV, nous avons intégré la CPS dans le simulateur Veins pour permettre les communications CPM entre les véhicules. En outre, nous avons proposé un cadre de confiance qui prend en compte deux attaques de confiance basées sur CAM, à savoir OOA et NCA, et quatre attaques GV basées sur CPM, à savoir Constant, Décalage constant, Aléatoire et Décalage aléatoire. Une simulation de scénario à trois véhicules a été réalisée dans le cadre des attaques susmentionnées afin de valider le modèle de confiance proposé. Les résultats de la simulation montrent l'efficacité du modèle proposé en tenant compte de deux types de comportements erronés basés sur les CAM et de quatre types de comportements erronés basés sur les CPM. Étant donné que le trafic réel reste complexe, les scénarios de simulation et les modèles d'attaque seront explorés et discutés dans nos travaux futurs par le biais d'une simulation à plus grande échelle impliquant un plus grand nombre d'entités de l'IoV.

Bibliography

[1] Runbo Su et al. "PDTM: Phase-based dynamic trust management for Internet of things". In: *2021 International Conference on Computer Communications and Networks (ICCCN)*. IEEE. 2021, pp. 1–7. DOI: 10.1109/ICCCN52240.2021.9522234.

[2] Runbo Su et al. "Ensuring Trustworthiness in IoIT/AIoT: A Phase-Based Approach". In: *IEEE Internet of Things Magazine* 5.2 (2022), pp. 84–88. DOI: 10.1109/IOTM.001.2100190.

[3] Runbo Su, Yujun Jin, and Ye-Qiong Song. "A lightweight cooperative trust model for IoV". In: (2023).

[4] Runbo Su et al. "A Game Theoretical Model addressing Misbehavior in Crowdsourcing IoT". In: *2023 20th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE. 2023, pp. 195–203. DOI: 10.1109/SECON58729.2023.10287527.

[5] Runbo Su and Amaury Saint-Jore. "A Role-based Trust Model assessing IoA services: First Results on Real MAS Implementation by using ROS 2". In: EWSN '23. ACM, 2024, pp. 293–296. DOI: 10.5555/3639940.3639978.

[6] Runbo Su., Yujun Jin., and Ye-Qiong Song. "Assessing Trustworthiness of V2X Messages: A Cooperative Trust Model Against CAM- and CPM-Based Ghost Vehicles in IoV". In: *Proceedings of the 10th International Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS*. INSTICC. SciTePress, 2024, pp. 276–283. ISBN: 978-989-758-703-0. DOI: 10.5220/0012605200003702.

[7] Runbo Su, Arbia Riahi Sfar, and Pascal Moyal. "Game theoretical analysis of strategy changes and influence factors in Crowdsourcing IoT systems". Abu Dhabi, United Arab Emirates, Apr. 2024. URL: https://hal.science/hal-04564953.

[8] Runbo Su et al. "Assessing intra- and inter-community trustworthiness in IoT: A role-based attack-resilient dynamic trust management model". In: *Internet of Things* (2024), p. 101213. ISSN: 2542-6605. DOI: https://doi.org/10.1016/j.iot.2024.101213.

[9] Oliver Schilke, Martin Reimann, and Karen S Cook. "Trust in social relations". In: *Annual Review of Sociology* 47 (2021), pp. 239–259.

[10] D GAMBETTA. "Can we trust trust?" In: *Trust: Making and Breaking Cooperative Relations* (1988).

[11] Harvey S James Jr. "The trust paradox: a survey of economic inquiries into the nature of trust and trustworthiness". In: *Journal of Economic Behavior & Organization* 47.3 (2002), pp. 291–307.

[12] Karen Cook. *Trust in society*. Russell Sage Foundation, 2001.

[13] Roger C Mayer, James H Davis, and F David Schoorman. "An integrative model of organizational trust". In: *Academy of management review* 20.3 (1995), pp. 709–734.

[14] Piotr Sztompka. *Trust: A sociological theory*. Cambridge university press, 1999.

[15] Andreja Rojko. "Industry 4.0 concept: Background and overview." In: *International Journal of Interactive Mobile Technologies* 11.5 (2017).

[16] Ling Li. "China's manufacturing locus in 2025: With a comparison of "Made-in-China 2025" and "Industry 4.0"". In: *Technological Forecasting and Social Change* 135 (2018), pp. 66–74.

[17] *Positive Technologies: 91% of Industrial Companies Open to Cyber-Attacks*. 2021. URL: https://www.ptsecurity.com/ww-en/about/news/positive-technologies-91-of-industrial-companies-open-to-cyber-attacks/.

[18] Iqbal H Sarker et al. "Internet of things (iot) security intelligence: a comprehensive overview, machine learning solutions and research directions". In: *Mobile Networks and Applications* 28.1 (2023), pp. 296–312.

[19] Matt Blaze, Joan Feigenbaum, and Jack Lacy. "Decentralized trust management". In: *Proceedings 1996 IEEE Symposium on Security and Privacy*. IEEE. 1996, pp. 164–173.

[20] Avani Sharma et al. "Towards trustworthy Internet of Things: A survey on Trust Management applications and schemes". In: *Computer Communications* 160.June (2020), pp. 475–493. ISSN: 1873703X. DOI: 10.1016/j.comcom.2020.06.030. URL: https://doi.org/10.1016/j.comcom.2020.06.030.

[21] Ayesha Altaf et al. "Trust models of internet of smart things: A survey, open issues, and future directions". In: *Journal of Network and Computer Applications* 137 (2019), pp. 93–111.

[22] Arbia Riahi Sfar et al. "A roadmap for security challenges in the Internet of Things". In: *Digital Communications and Networks* 4.2 (2018), pp. 118–137.

[23] Zinal D Patel. "A review on service oriented architectures for internet of things (IoT)". In: *2018 2nd international conference on trends in electronics and informatics (ICOEI)*. IEEE. 2018, pp. 466–470.

[24] GR Sagar and N Jayapandian. "Internet of things: service-oriented architecture opportunities and challenges". In: *Inventive Communication and Computational Technologies: Proceedings of ICICCT 2019* (2020), pp. 71–78.

[25] Naseem Ibrahim and Brandon Bench. "Service-oriented architecture for the internet of things". In: *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE. 2017, pp. 1004–1009.

[26] Zainab M Aljazzaf, Miriam AM Capretz, and Mark Perry. "Trust-based service-oriented architecture". In: *Journal of King Saud University-Computer and Information Sciences* 28.4 (2016), pp. 470–480.

[27] Timam Ghosh et al. "AI-Based Communication-as-a-Service for Network Management in Society 5.0". In: *IEEE Transactions on Network and Service Management* (2021).

[28] Nektaria Kaloudi and Jingyue Li. "The ai-based cyber threat landscape: A survey". In: *ACM Computing Surveys (CSUR)* 53.1 (2020), pp. 1–34.

[29] Murat Kuzlu, Corinne Fair, and Ozgur Guler. "Role of artificial intelligence in the Internet of Things (IoT) cybersecurity". In: *Discover Internet of Things* 1.1 (2021), pp. 1–14.

[30] Hassan I Ahmed et al. "A survey of IoT security threats and defenses". In: *International Journal of Advanced Computer Research* 9.45 (2019), pp. 325–350.

[31] Yanli Yu et al. "Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures". In: *Journal of Network and computer Applications* 35.3 (2012), pp. 867–880.

[32] Ray Chen and Jia Guo. "Dynamic hierarchical trust management of mobile groups and its application to misbehaving node detection". In: *2014 IEEE 28th international conference on advanced information networking and applications*. IEEE. 2014, pp. 49–56.

[33] Yan Lindsay Sun et al. "A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks". In: *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*. IEEE. 2006, pp. 1–13.

[34] Yosra Ben Saied et al. "Trust management system design for the Internet of Things: A context-aware and multi-service approach". In: *Computers & Security* 39 (2013), pp. 351–365.

[35] Amitanand S Aiyer et al. "BAR fault tolerance for cooperative services". In: *Proceedings of the twentieth ACM symposium on Operating systems principles*. 2005, pp. 45–58.

[36] Amira Bradai, Walid Ben-Ameur, and Hossam Afifi. "Byzantine resistant reputation-based trust management". In: *9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*. IEEE. 2013, pp. 269–278.

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

[38] Safaa Hriez et al. "A novel trust-aware and energy-aware clustering method that uses stochastic fractal search in IoT-enabled wireless sensor networks". In: *IEEE Systems Journal* (2021).

[39] Guntur Dharma Putra et al. "Trust management in decentralized iot access control system". In: *2020 IEEE international conference on blockchain and cryptocurrency (ICBC)*. IEEE. 2020, pp. 1–9.

[40] Upul Jayasinghe et al. "Machine learning based trust computational model for IoT services". In: *IEEE Transactions on Sustainable Computing* 4.1 (2019), pp. 39–52.

[41] Jia Guo, Ray Chen, and Jeffrey JP Tsai. "A survey of trust computation models for service management in internet of things systems". In: *Computer Communications* 97 (2017), pp. 1–14.

[42] A Meena Kowshalya and ML Valarmathi. "Trust management for reliable decision making among social objects in the Social Internet of Things". In: *IET Networks* 6.4 (2017), pp. 75–80.

[43] 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: *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE. 2017, pp. 747–752.

[44] Mohammad Dahman Alshehri, Farookh Khadeer Hussain, and Omar Khadeer Hussain. "Clustering-driven intelligent trust management methodology for the internet of things (CITM-IoT)". In: *Mobile networks and applications* 23.3 (2018), pp. 419–431.

[45] Pintu Kumar Sadhu, Venkata P Yanambaka, and Ahmed Abdelgawad. "Internet of things: Security and solutions survey". In: *Sensors* 22.19 (2022), p. 7433.

[46] Stavros Salonikias et al. "Access control in the industrial internet of things". In: *Security and privacy trends in the industrial internet of things*. Springer, 2019, pp. 95–114.

[47] Safwa Ameer, James Benson, and Ravi Sandhu. "An attribute-based approach toward a secured smart-home IoT access control and a comparison with a role-based approach". In: *Information* 13.2 (2022), p. 60.

[48] Luigi Atzori et al. "The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization". In: *Computer networks* 56.16 (2012), pp. 3594–3608.

[49] Charith Perera et al. "Context aware computing for the internet of things: A survey". In: *IEEE communications surveys & tutorials* 16.1 (2013), pp. 414–454.

[50] Asrin Vakili and Nima Jafari Navimipour. "Comprehensive and systematic review of the service composition mechanisms in the cloud environments". In: *Journal of Network and Computer Applications* 81 (2017), pp. 24–36.

[51] Akseer Ali Mirani et al. "Key Challenges and Emerging Technologies in Industrial IoT Architectures: A Review". In: *Sensors* 22.15 (2022), p. 5836.

[52] Claudio Marche and Michele Nitti. "Can We Trust Trust Management Systems?" In: *IoT* 3.2 (2022), pp. 262–272.

[53] Bong Gu Kang, Kyung-Min Seo, and Tag Gon Kim. "Model-based design of defense cyber-physical systems to analyze mission effectiveness and network performance". In: *IEEE Access* 7 (2019), pp. 42063–42080.

[54] Wei Ma et al. "Machine learning empowered trust evaluation method for IoT devices". In: *IEEE access* 9 (2021), pp. 65066–65077.

[55] Xianming Huang. "Intelligent remote monitoring and manufacturing system of production line based on industrial Internet of Things". In: *Computer Communications* 150 (2020), pp. 421–428.

[56] Ahmed Saidi, Khelifa Benahmed, and Nouredine Seddiki. "Secure cluster head election algorithm and misbehavior detection approach based on trust management technique for clustered wireless sensor networks". In: *Ad Hoc Networks* 106 (2020), p. 102215.

[57] Wafa Abdelghani. "A multi-dimensional trust-model for dynamic, scalable and resources-efficient trust-management in social internet of things". Theses. Université Paul Sabatier - Toulouse III ; Université de Sfax (Tunisie), Dec. 2020. URL: `https://tel.archives-ouvertes.fr/tel-03215718`.

[58] Marina Sokolova, Nathalie Japkowicz, and Stan Szpakowicz. "Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation". In: *Australasian joint conference on artificial intelligence*. Springer. 2006, pp. 1015–1021.

[59] Navneet Dalal and Bill Triggs. "Histograms of Oriented Gradients for Human Detection". In: *International Conference on Computer Vision & Pattern Recognition (CVPR '05)*. Ed. by Cordelia Schmid, Stefano Soatto, and Carlo Tomasi. Vol. 1. San Diego, United States: IEEE Computer Society, June 2005, pp. 886–893. DOI: `10.1109/CVPR.2005.177`. URL: `https://inria.hal.science/inria-00548512`.

[60] Parrot. *Anafi*. `https://www.parrot.com/fr/drones/anafi`. Accessed: 03.15.2024.

[61] Boston Dynamic. *Sport*. `https://bostondynamics.com/products/spot/`. Accessed: 03.15.2024.

[62] *Raspberry Pi*. https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/. Accessed: 03.15.2024.

[63] ROS2. *Humble*. https://docs.ros.org/en/humble. Accessed: 03.15.2024.

[64] *Wifi Access*. https://www.asus.com/fr/networking-iot-servers/wifi-routers/asus-gaming-routers/rt-ax92u/. Accessed: 03.15.2024.

[65] Steven Macenski and et al. "Robot Operating System 2: Design, architecture, and uses in the wild". In: *Science Robotics* 7.66 (May 2022). (Visited on 03/01/2024).

[66] Pablo Iñigo-Blasco and et al. "Robotics software frameworks for multi-agent robotic systems development". en. In: *Robotics and Autonomous Systems* 60.6 (June 2012), pp. 803–821. ISSN: 09218890. DOI: 10.1016/j.robot.2012.02.004. (Visited on 02/21/2024).

[67] *Node Graphe in Ros2*. https://docs.ros.org/en/foxy/Tutorials/Beginner-CLI-Tools/Understanding-ROS2-Nodes/Understanding-ROS2-Nodes.html. Accessed: 03.15.2024.

[68] Kun Wang et al. "Toward trustworthy crowdsourcing in the social internet of things". In: *IEEE Wireless Communications* 23.5 (2016), pp. 30–36.

[69] Victor Naroditskiy et al. "Crowdsourcing contest dilemma". In: *Journal of The Royal Society Interface* 11.99 (2014), p. 20140532.

[70] Qin Hu et al. "Solving the crowdsourcing dilemma using the zero-determinant strategies". In: *IEEE Transactions on Information Forensics and Security* 15 (2020), pp. 1778–1789.

[71] Chuanxiu Chi et al. "Multistrategy repeated game-based mobile crowdsourcing incentive mechanism for mobile edge computing in Internet of Things". In: *Wireless Communications and Mobile Computing* 2021 (2021), pp. 1–18.

[72] Montdher Alabadi and Zafer Albayrak. "Q-learning for securing cyber-physical systems: a survey". In: *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. IEEE. 2020, pp. 1–13.

[73] Dipyaman Banerjee and Sandip Sen. "Reaching pareto-optimality in prisoner's dilemma using conditional joint action learning". In: *Autonomous Agents and Multi-Agent Systems* 15 (2007), pp. 91–108.

[74] Stefano V Albrecht, Jacob W Crandall, and Subramanian Ramamoorthy. "Belief and truth in hypothesised behaviours". In: *Artificial Intelligence* 235 (2016), pp. 63–94.

[75] Stefano V Albrecht. "Utilising policy types for effective ad hoc coordination in multiagent systems". PhD thesis. The University of Edinburgh, 2015.

[76] Benny Vejlgaard et al. "Coverage and capacity analysis of sigfox, lora, gprs, and nb-iot". In: *2017 IEEE 85th vehicular technology conference (VTC Spring)*. IEEE. 2017, pp. 1–5.

[77] Edgar Saavedra et al. "A Universal Testbed for IoT Wireless Technologies: Abstracting Latency, Error Rate and Stability from the IoT Protocol and Hardware Platform". In: *Sensors* 22.11 (2022), p. 4159.

[78] Arbia Riahi Sfar et al. "A game theoretic approach for privacy preserving model in IoT-based transportation". In: *IEEE Transactions on Intelligent Transportation Systems* 20.12 (2019), pp. 4405–4414.

[79] ETSI. *ETSI*. https://www.etsi.org/. Accessed: 03.28.2024.

[80] Efstathios Zavvos et al. “Privacy and Trust in the Internet of Vehicles”. In: *IEEE Transactions on Intelligent Transportation Systems* 23.8 (2021), pp. 10126–10141.

[81] Hassan Farran and David Khoury. “Performance Improvements of Vehicular PKI Protocol for the Security of V2X Communications”. In: *2023 46th TSP*. 2023, pp. 177–182.

[82] Sohan Gyawali and Yi Qian. “Misbehavior detection using machine learning in vehicular communication networks”. In: *2019-2019 IEEE ICC*. 2019, pp. 1–6.

[83] Sohan Gyawali et al. “Challenges and solutions for cellular based V2X communications”. In: *IEEE Communications Surveys & Tutorials* 23.1 (2020), pp. 222–255.

[84] Baofeng Ji et al. “Survey on the internet of vehicles: Network architectures and applications”. In: *IEEE Communications Standards Magazine* 4.1 (2020), pp. 34–41.

[85] ETSI. “103 415 V1. 1.1 -intelligent transport systems; Security; Pre-standardization study on Misbehaviour Detection; Release 2”. In: *ETSI, Oct.* (2020).

[86] ETSI. “302 637-2 V1.4.1 -intelligent transport systems; vehicular communications; basic set of applications; part 2: Specification of cooperative awareness basic service”. In: *ETSI, Apr* (2019).

[87] ETSI. “103 324 V2.1.1 -intelligent transport system (its); vehicular communications; basic set of applications; collective perception service; release 2”. In: *ETSI, Jun* (2023).

[88] Moreno Ambrosin et al. “Design of a Misbehavior Detection System for Objects Based Shared Perception V2X Applications”. In: *2019 IEEE ITSC*. 2019, pp. 1165–1172. DOI: 10.1109/ITSC.2019.8917066.

[89] Rens W Van Der Heijden, Thomas Lukaseder, and Frank Kargl. “Veremi: A dataset for comparable evaluation of misbehavior detection in vanets”. In: *14th SecureComm 2018, Proceedings, Part I*. Springer. 2018, pp. 318–337.

[90] Veins. *Veins*. https://veins.car2x.org/. Accessed: 12.24.2023.

[91] Simon Bachmeier, Benedikt Jaeger, and Kilian Holzinger. “Network Simulation with OMNeT++”. In: *Network* 37 (2020).

[92] Hamssa Hasrouny et al. “Trust model for secure group leader-based communications in VANET”. In: *Wireless Networks* 25 (2019), pp. 4639–4661.

[93] Assiya Akli and Khalid Chougdali. “A survey on Machine Learning for IoT Trust Management”. In: *2023 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)*. IEEE. 2023, pp. 59–65.

[94] Behrouz Pourghebleh, Karzan Wakil, and Nima Jafari Navimipour. “A comprehensive study on the trust management techniques in the Internet of Things”. In: *IEEE Internet of Things Journal* 6.6 (2019), pp. 9326–9337.

Résumé

À la différence de la confiance dans les sciences sociales, où les interactions entre les humains sont mesurées, la notion de confiance dans le domaine de la sécurité pour l'IdO (Internet des Objets) se concentre davantage sur les interactions entre les nœuds (objets) grâce à l'intégration des objets intelligents. De plus, comme les nœuds de l'IdO peuvent se regrouper au sein d'une communauté, de par des intérêts et fonctionnalités similaires, l'évaluation de la confiance, aussi dans un contexte groupe-groupe, individu-individu ou encore groupe-individu, est également importante. Cependant, la gestion des limitations induites par certaines menaces (e.g., trust-related attack and BAR Model) et de la vulnérabilité inhérente à l'architecture de la gestion de confiance reste un défi. Cette thèse étudie la notion de confiance sous trois angles dans le cadre de l'internet des objets orienté services (SO-IoT) : la confiance groupe-groupe, la confiance groupe-individu et la confiance individu-individu. Dans un premier temps, un modèle dynamique basé sur les rôles est développé pour évaluer la confiance intra-groupe et inter-groupe, en améliorant les activités orientées services et en abordant les questions de sécurité aussi bien au sein des communautés qu'entre elles. Une approche localement centralisée en quatre phases est employée. Celle-ci se concentre sur les moyens de contrer les attaques ciblant les services au sein du groupe. Un mécanisme en trois phases est également conçu pour mesurer la coopération entre les groupes. Une implémentation basée sur le système ROS 2 a été mise en œuvre afin d'analyser les performances du modèle proposé, celles-ci s'appuyant sur des résultats préliminaires. Dans un second temps, pour traiter le mauvais comportement dans SO-IoT en termes de confiance individu-individu, un Jeu Stochastique Bayésien (JSB) est introduit dans notre modèle. Ce JSB prend en compte l'hétérogénéité des nœuds IoT et des schémas comportementaux complexes des fournisseurs de services, encourageant la coopération et pénalisant les actions stratégiques malveillantes. Enfin, le travail d'évaluation de la confiance de messages V2X dans l'IoV (Internet of Vehicles) démontre la possibilité de mettre en œuvre un modèle de gestion de confiance dans un environnement pratique de type IdO.

Mots-clés: Gestion de la confiance, la sécurité de l'IdO, Modélisation de la confiance, détection du comportement malveillant, l'IdO orienté services

Abstract

Unlike Trust in Social Science, in which interactions between humans are measured, thanks to the integration of numerous smart devices, Trust in IoT security focuses more on interactions between nodes. Moreover, As IoT nodes can somehow benefit from 'Group'/'Community' since they form by similar interests or functionalities, the assessment of Group-Individual and Inter-Individual Trust is also important. However, handling limitations brought by potential threats and inherent vulnerability due to TM architecture remains challenging. This thesis investigates Trust from three perspectives in the Service-Oriented Internet of Things (SO-IoT): Inter-Group Trust, Group-Individual Trust, and Inter-Individual Trust. Firstly, a role-based dynamic model is developed to assess intra- and inter-community(group), enhancing service-oriented activities and addressing security issues within and between communities. A locally centralized four-phase approach is employed, focusing on countermeasures against attacks on services within

the community. Additionally, a three-phase mechanism is devised to measure cooperativeness between communities. An implementation based on the ROS 2 system was implemented to analyze the performance of the proposed model based on the preliminary results. Secondly, to address misbehavior in SO-IoT in terms of Inter-Individual trust, a Stochastic Bayesian Game (SBG) is introduced, which considers the heterogeneity of IoT nodes, and complex behavioral schemes of service providers are incorporated, encouraging cooperation and penalizing malicious strategical actions. Lastly, the work of assessing the Trust of V2X messages in IoV demonstrates the possibility of implementing Trust Management in a concrete IoT environment.

Keywords: Trust Management, IoT security, Trust modeling, misbehavior detection, Service-Oriented IoT