Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Research article

Assessing intra- and inter-community trustworthiness in IoT: A role-based attack-resilient dynamic trust management model

Runbo Su [a,*], Arbia Riahi [b], Enrico Natalizio [c,a], Pascal Moyal [d], Amaury Saint-Jore [a], Ye-Qiong Song [a]

[a] *LORIA, CNRS, Université de Lorraine, France*
[b] *Cylab, CISS, Royal Military Academy, Belgium*
[c] *Technology Innovation Institute, United Arab Emirates*
[d] *IECL, INRIA, Université de Lorraine, France*

ARTICLE INFO

Keywords:
Security in IoT
Trust management (TM)
Attack on service
Malicious behavior detection
Implementation

ABSTRACT

IoT is regarded as the key technology for boosting the Industry 4.0 revolution. However, the introduction of high-intelligence devices and complex services raises new challenges for security in IoT. In this paper, a role-based attack-resilient trust management (TM) model for community-driven IoT is proposed at two different levels. First, the intra-community TM enables the IoT nodes within the same community 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 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.

1. Introduction

Since ‘Industry 4.0’ was first proposed in Germany [1], 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’ [2], 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, till now, more than 90% of companies are vulnerable to cyber threats according to the research from Positive Technologies [3], meaning the trustworthiness of IoT devices and services remains uncertain. 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 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 in IoT systems. In this regard, trust management plays a crucial role as it analyzes nodes’ behaviors over time.

[*] Correspondence to: Team Simbiot, LORIA, Université de Lorraine, France.
E-mail addresses: runbo.su@loria.fr, runbo.su@univ-lorraine.fr (R. Su).

https://doi.org/10.1016/j.iot.2024.101213
Received 29 January 2024; Received in revised form 2 May 2024; Accepted 3 May 2024

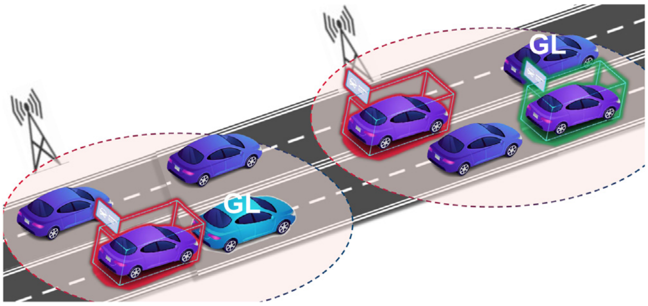


Fig. 1. Cluster-based vehicle groups (GL = group leader).

Existing trust models mainly considered applying centralized and distributed TM [4]. Centralized ones conduct TM by a single entity, and models suffer from one single failure point issue. In a distributed architecture, nodes are self-organizing, but indirect trust assessment poses a huge problem of computation complexity, and trust initialization remains difficult for newcomer nodes due to the lack of global opinion in terms of trust. Both of them are not optimal for addressing the scalability issues and improving the service-oriented activities in IoT systems [5]. For this reason, more and more IoT systems are considering introducing community-driven trustworthiness assessment, where heterogeneous IoT nodes are grouped into different communities by interests and utilities [6,7]. For instance, controller-based smart industrial plants in IIoT (Industrial Internet of Things) [8], where factories are monitored by interconnected smart industrial controllers for dedicated production missions. Another example is illustrated in Fig. 1 by clustered vehicle groups in IoV (Internet of Vehicles) [9], where a group leader is elected to supervise other vehicles in the group. From this perspective, in the literature, a model assessing the trustworthiness in a way that intra-community nodes' trust can be locally governed, communities disturbingly communicate, and evaluate the trust of each other, is still missing. Furthermore, most of the existing models use a static service evaluation scheme without differentiating service types and considering the service composition process, and due to that, only a few studies consider evaluating the trustworthiness of different roles, namely the SP and SR. This also leads to vulnerability to attacks on services since most of the existing models rarely consider both sides. Thus, a framework covering the trust evaluation of SP and SR is needed. In this context, we propose a four-phase role-based trust management framework assessing intra- and inter-community trustworthiness in IoT. More importantly, the validation of the existing TM models is all based on theoretical analysis or simulation results, and no implementation with real devices is conducted to support their proposed model [5]. For this, in our work, we carry out an implementation by using ROS (Robot Operating System) 2 for real-world testing. Our contributions are as follows:

- We design a hybrid trust architecture containing intra- and inter-community levels suitable for community-driven IoT systems. In the first level, nodes are evaluated on the basis of their roles as Service Providers (SP) or Service Raters (SR). The second level enables exchange between managers for inter-community trust assessment.
- We develop countermeasures addressing various attacks on service in a way that nodes are assessed during the different phases, namely trust initialization, service provision, and final decision-making (i.e., node classification). The simulation results show effectiveness against the considered attacks.
- We utilize ROS 2 to realize an implementation with real-world devices. Our objective is to show the proposed model's feasibility and partly validate our work through a specific scenario.

The rest of this paper is organized as follows. Section 2 describes the background and reviews the related contributions to TM models. Section 3 discusses the architecture of the proposed model, which supports community-driven IoT. Section 4 details intra- and inter-community TM. The simulation results and performance analysis are presented in Section 5. Section 6 details the implementation realized and the preliminary results obtained. Section 7 draws the conclusion and outlines our future work.

2. Background and motivation

In this section, we give the motivation for our work by introducing the background and reviewing related work in detail. This also allows us to study the constraints and the techniques to design a TM model that can meet the requirements of a community-driven IoT and overcome its related issues.

2.1. SOA (service-oriented architecture) in IoT and related threats

In IoT, intelligent IoT nodes can participate collaboratively in complex IoT services on the basis of SOA [10] illustrated in Fig. 2. The 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.

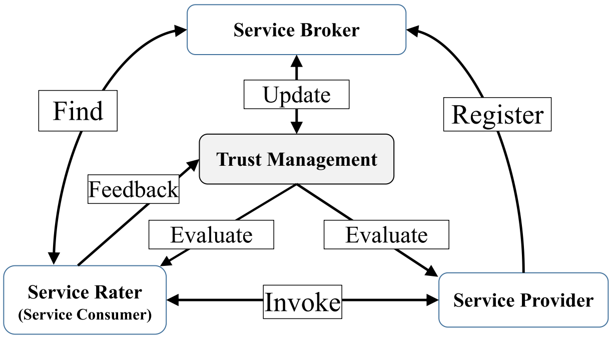


Fig. 2. SOA-based TM.

Table 1
Categories of attacks on services [11–14].

Src.	Attack		Target	Ref.
SR	URA	BSA	Rating	[11]
		BMA		[12]
		SPA		
	NCA		Service, Rating	
SP				
	IBA	CBA		[13]
		OOA		
		SBA		[14]

By integrating the TM into the SOA, Fig. 2 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 SR and SP 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 behaving poorly in the provision of service s_1 may be outstanding at performing service s_2. Thus, securing IoT requires the TM to interact with aforementioned service activities to assign to nodes and services accurate trust values. Furthermore, a poor SP may perform excellently as an honest SR, and a less capable SR may provide consistently satisfactory service. To make decisions for such nodes, a global opinion based on their workloads of SR and SP is also needed to determine their overall nature.

In regard to attacks on service, Table 1 classifies them by attack source and target:

- Unfair Rating Attacks (URA) aim at creating disorder in the evaluation system by delivering dishonest ratings and consist of three kinds:
  - Ballot Stuffing Attack (BSA): The attacker highly recommends malicious nodes to increase their reputation.
  - Bad Mouthing Attack (BMA): The attacker sends negative feedback to decrease the trust score of a good service provider.
  - Self-promoting Attack (SPA): The attacker provides positive ratings for itself, intending to be selected for service provision.
- Newcomer Attack (NCA): The attacker re-enters the system with a new identity to refresh its trust score. The attack source can be a service rater or provider when NCA occurs.
- Inconsistent Behavior Attacks (IBA) are from the malicious service provider and can also be of three distinct types:
  - Conflicting Behavior Attack (CBA): The attacker performs differently with different nodes.
  - On–off Attack (OOA): The attacker switches its behavior between good and bad over time to maintain its trust score above a certain threshold.
  - Selective Behavior Attack (SBA): The attacker performs well and badly between services in an alternative manner.

Building countermeasures against attacks on services to prevent adverse effects is an essential task in SOA-based TM to show robustness and effectiveness in treating these attacks. It should be noted that our model focuses on addressing attacks on service, and we thus assume that attacks on communication such as DoS (Denial of Service) have been addressed by other security-related communication schemes [15].

2.2. Existing TM models and limitations

With the purpose of monitoring IoT nodes' behavior in terms of service, some works applied reward and penalty schemes to help the trust evaluation. The work in [16] designed a game-based trust framework to reward cooperative service provision and

Table 2
Summary of related TM models.

Ref.	Com.	SOA	Attacks on services						
			BSA	SPA	BMA	NCA	CBA	OOA	SBA
[16]	–	x	–	–	–	✓	–	✓	x
[17]	–	–	–	–	–	–	–	✓	–
[14]	–	x	✓	–	✓	–	–	✓	–
[18]	–	–	–	–	–	–	–	✓	–
[19]	–	–	✓	✓	✓	–	–	✓	–
[20]	x	–	–	–	–	–	–	✓	–
[21]	x	–	✓	–	✓	–	–	–	–
[22]	✓	–	–	–	–	–	–	–	–
Our work	✓	✓	✓	✓	✓	✓	✓	✓	✓

Com. = Community-driven, ✓= Supported, x = Partially Supported.

address misbehavior in distributed crowdsourcing IoT systems. Authors in [17] presented a distributed service score mechanism where the unsatisfactory service will be punished by doubled distrust. The TM model in [14] proposed a centralized mechanism for measuring and updating reputations and recommendations in a collective context. In [18], authors designed a decentralized trust model named DECAY, focusing on demotivating passive behavior and promoting active participation. A model in [19] suggested regulating IoT service interactions using reputation systems and blockchain technology, where the reward–penalty scheme is posed through a customizable architecture. These studies have considered motivating excellent services and discouraging poor ones. However, these works remain challenging for community-driven IoT systems due to their TM architectures, either fully centralized or fully distributed. Authors in [20] 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 in this work, but the lack of countermeasures against malicious SR makes the TM model vulnerable since it proposed a leader selection scheme. [21] introduced an intelligent hierarchical approach to create a hybrid TM environment, where the architecture consists of Master Node that manages cluster nodes and Super Node that handles the allocation of the cluster for MN. This model proposed an algorithm to eliminate rating outliers. However, the dishonest SR detection and isolation mechanisms 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. [22] proposed a trust model by using clustering analysis, where node locations form the cluster, and one master node per cluster will be periodically updated based on the trust value using regression model-based clustering. On the one hand, this work can be applied to community-driven IoT. On the other hand, it does not support SOA and no trust-related attacks are addressed.

Table 2 gives a summarized comparison between the above-reviewed TM models. From the perspective of role-based assessment, [14] viewed nodes act as SR and SP at the same time in a collective service scenario, the quality of the SP side is insufficiently solved in [21,22], and works in [17,18,20,22] only slightly investigated the quality of SR side. More importantly, a global opinion based on SR and SP workloads to determine the node's overall nature is still missing in existing studies. Furthermore, most of the aforementioned TM models addressed a few attacks on services, meaning comprehensive countermeasures are still missing. Most of them treated IoT services homogeneously, and none of them discussed service composition. In regard to TM architecture, some of them are either fully distributed or fully centralized, which puts their suitability for community-driven IoT systems in question. To overcome the above-mentioned limitations, in this work, we design a novel TM model. Since [20,21] are partly suitable for community-driven IoT systems and they proposed countermeasures against trust-related attacks, they are used in for a comparative purpose.

3. Proposed framework

This section presents the proposed hybrid TM architecture, which is suitable to evaluate intra- and inter-community trustworthiness in IoT.

As analyzed in Section 2, considering that fully centralized or fully distributed architectures are not sufficiently advantageous in terms of applicability and security, we design a hybrid architecture to support and improve the TM in IoT. Fig. 3 demonstrates the proposed architecture of the proposed trust framework with two levels: intra- and inter-community TM. 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.

4. TM model

We present the intra-community TM model first by detailing each phase. Then, we explain different AC cases, namely the newcomer node, the returner node back to the original community, and the node moving to the new community. Lastly, we focus on the TM at the inter-community level.

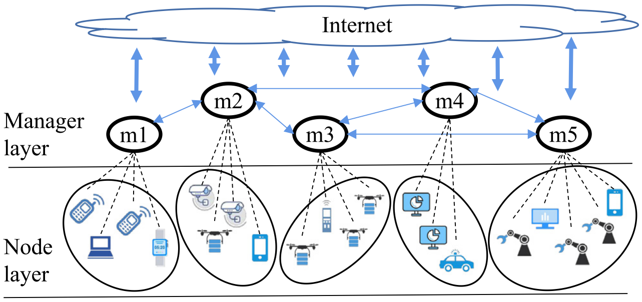


Fig. 3. Proposed TM architecture.

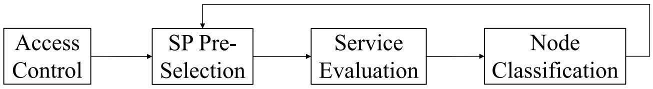


Fig. 4. Four-phase intra-community trust assessment.

4.1. Intra-community TM

4.1.1. Overview of four-phase intra-community TM

Fig. 4 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 [23], 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 Sections 4.1.2–4.1.5, and in particular, different AC cases will be discussed in Section 4.1.6.

4.1.2. Access control

Since recording the node by its asserted attributes is more advantageous in terms of security [24], we employ the ABAC (Attribute-Based Access Control) for the AC phase, which is regarded as a logical methodology for AC in IoT [25]. 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** ($A^{\text{soc}}$): 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 [26]. In our model, we consider three main object relationships: parental (PR) and co-work (CWR) relationships. Nodes belonging to the same manufacturer have higher PR, as their characteristics in terms of software specification are somehow approximated. On the other hand, 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 [27]. In the proposed model, the context attribute contains a hash value that enables verifying if the comer node is a newcomer.

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

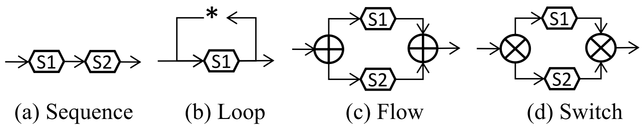
$$DS_i^{\text{fct}} = \begin{cases} 1, & \text{if } \ S_i^{\text{fct}} \neq \infty \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

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

$$DS_i^{\text{soc}} = \mu^{PR} PR_i + \mu^{CWR} CWR_i, \tag{2}$$

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

$$PR_i = \frac{1}{|CN|} \sum_{k \in CN} v_{ik}^{PR}, \tag{3}$$



(a) Sequence (b) Loop (c) Flow (d) Switch

Fig. 5. Constructs for workflow.

$$CWR_i = \frac{1}{|CN|} \sum_{k \in CN} \frac{|S_i^{fct} \cap S_k^{fct}|}{|S_i^{fct} \cup S_k^{fct}|}, \tag{4}$$

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

$$DS_i^{\mathrm{ctx}} = \begin{cases} 0.5, & \text{if } i \text{ is newcomer} \\ 1, & \text{otherwise} \end{cases} \tag{5}$$

With three sub-DS values calculated by (1)–(5), the DS of node i is defined as

$$DS_i = (\omega^{\mathrm{fct}} \cdot DS_i^{\mathrm{fct}} + \omega^{\mathrm{soc}} \cdot DS_i^{\mathrm{soc}})^{1/DS_i^{\mathrm{ctx}}}, \tag{6}$$

where $\omega^{\mathrm{fct}} + \omega^{\mathrm{soc}} = 1$. 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 [28], 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.

4.1.3. Service provider (SP) selection

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 [29], which often occurs in industrial context [30]. 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. 5 illustrates commonly-used service composition constructs, including sequence → (s1, then s2), loop * (s1 several times), flow ⊕ (s1 and s2), and switch ⊗ (s1 or s2). Consider so-called Service Grade SG_i^n representing the quality of the node i's service of the type n, (7) gives the calculation of OSG in case of workflow, we fix r for loop * representing the times of repetitions.

$$OSG_i = \begin{cases} \prod_{n \in S_{\mathrm{req}}} SG_i^n, & \text{for } \rightarrow \\ (SG_i^n)^r, & \text{for } * \\ 1 - \prod_{n \in S_{\mathrm{req}}} (1 - SG_i^n), & \text{for } \oplus \\ max_{n \in S_{\mathrm{req}}} (SG_i^n), & \text{for } \otimes \\ SG_i^n, & \text{if } n = S_{req} \text{ (single service)} \end{cases} \tag{7}$$

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

$$SS_i = \begin{cases} QSP_i \cdot OSG_i, & \text{for a workflow} \\ DS_i, & \text{if newcomer.} \end{cases} \tag{8}$$

The examination of the candidate SP in (8) concerns two sides: QSP gives the opinion from a more general view, e.g., the stability (refer to , 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.

4.1.4. Service evaluation

As authors analyzed in [31], the trustworthiness measurement of interactions between IoT nodes remains challenging when the service badly performs or the service is poorly evaluated. For this reason, in our work, we consider four trust metrics in this phase to comprehensively measure nodes' trustworthiness in terms of service, namely TS, QSR, QSP, and SG. QSR and QSP describe how much the node can be trusted as SR and SP, TS can give 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, such as the mission success rate or network performance in [28,32], and service recommendation in [33].

Trust score (TS). We set:

$$TS_i = \frac{|RS_i|}{|RS_i + PS_i|} \cdot QSR_i + \frac{|PS_i|}{|RS_i + PS_i|} \cdot QSP_i, \tag{9}$$

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 (10) and (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 (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

$$QSR_i = \varphi \cdot CQSR_i + (1-\varphi) \cdot LQSR_i \ , \tag{10}$$

where $\varphi \in [0.5, 1)$, and $CQSR$ 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 (10) is computed as follows:

$$CQSR_i = 1 - \frac{1}{|R_{i-}|} \sum_{j \in R_{i-}} |f_{ij} - \bar{f}_j|^{1/l} \ , \tag{11}$$

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 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 SIoT, 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).

$$QSP_i = \varepsilon \cdot CQSP_i + (1-\varepsilon) \cdot LQSP_i \ , \tag{12}$$

where ε is set in $[0.5, 1)$ to weigh 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 (12) is computed as follows:

$$CQSP_i = \frac{1}{|R_{-i}|} \sum_{j \in R_{-i}} \theta_{ji} \cdot \lambda_{ji} \cdot QSR_j \cdot f_{ji} \ , \tag{13}$$

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 (14) and (15): for all $j \in R_{-i}$,

$$\theta_{ji} = \operatorname{sinc}(1 - f_{ji}) \cdot \operatorname{sinc}(\Delta f_{ji})^{\Delta t_{ji}}, \tag{14}$$

$$\lambda_{ji} = 1 - |f_{ji} - \bar{f}_i|^{1/l}, \tag{15}$$

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

$$\operatorname{sinc}(x) = \begin{cases} 1, & \text{for } x = 0 \\ \frac{\sin(\pi x)}{\pi x}, & \text{for } x \neq 0, \end{cases} \tag{16}$$

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 (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 (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:

$$SG_i^n = \kappa \cdot CSG_i^n + (1-\kappa) \cdot LSG_i^n, \tag{17}$$

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

$$CSG_i^n = \frac{1}{|R_{-i}^n|} \sum_{j \in R_{-i}^n} QSR_j \cdot f_{ji}^n, \tag{18}$$

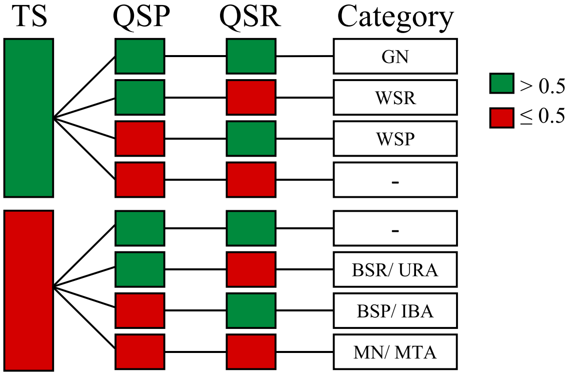


Fig. 6. Node classification scheme.

where R^n_{-i} is the set of nodes that rated the service of type n provided by i and f^n_{ji} 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 and alert other communities in case of need, e.g., when a node moves to another community. Hence, the misbehavior aiming at service types from malicious SP, namely SBA, will be authorized to 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 for the SP selection and will eventually be eliminated from the community.

Complexity of computing TS, QSP, QSR, and SG. The main computation of intra-community TM is calculating trust values in (9), (10), (12), and (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 designed computation scheme remains efficient as g is a linear function with respect to N.

4.1.5. Node classification

As stated in Section 2, 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, attacks on service are divided into two categories by attack source, namely SP and SR, the node classification should take into consideration a scheme to enable identifying 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. 6 enables the community manager to categorize nodes into 6 groups:

- 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.

4.1.6. Three AC cases

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

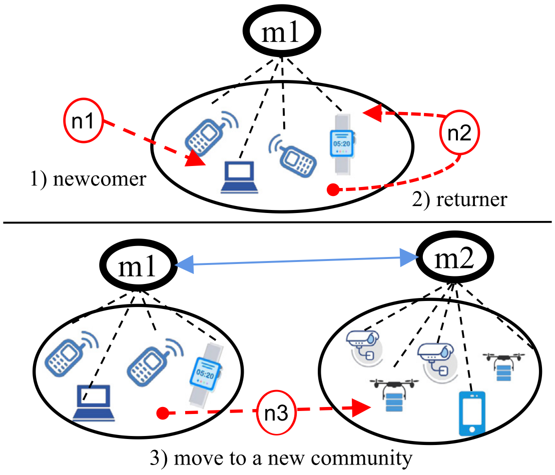


Fig. 7. Three AC cases and related checking mechanisms.

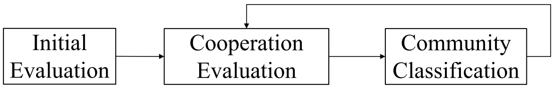


Fig. 8. Three-phase inter-community trust assessment.

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. 7 demonstrates three cases, namely newcomer, returner, and the node moving to the new community. To determine the initial trust and collect related information for attributes [34], 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. 10 shows how the community manager employs differently in different AC cases.

4.2. Inter-community TM

4.2.1. Overview of phases

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), such as the production line given in [35], different industrial factories are dedicated for specific production missions, to optimize the supply chain and achieve a common task in the production line, they have to cooperate in a sequential manner. 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 [36], 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. 8. 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.

4.2.2. 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

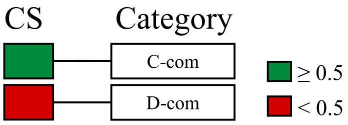


Fig. 9. Community classification scheme.

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:

$$IS_{xy} = \frac{|S_x \cap S_y|}{|S_x|}, \tag{19}$$

where S_x and S_y denote respectively the sets of all functional services needed in communities x and y. 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, considering an industrial context, given y and x have high IS, if a sudden failure of an essential service type comes to the factory y, it can immediately ask the other factory x to help by sending nodes providing this service.

4.2.3. *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):

$$CO_{xy} = \frac{|GN_{yx}|}{|MN_{yx}|}, \tag{20}$$

where GN_{yx} represents the good nodes (evaluated by the current community x according to the node classification scheme given in Fig. 6) from y to x, and MN_{yx} are all the nodes that moved from y to the new community x.

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

$$CS_{xy} = \begin{cases} IS_{xy}, & \text{before any interactions} \\ \eta^{CO} \cdot CO_{xy} + \eta^{IS} \cdot LCS_{xy}, & \text{otherwise,} \end{cases} \tag{21}$$

where LCS_{xy} represents the last CS_{xy} value, $\eta^{IS} + \eta^{CO} = 1$. $LCS_{xy} = IS_{xy}$ for the first evaluation. IS_{xy} 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.

The main computation of inter-community TM comes from (21). When the community x evaluates y, the computation complexity of calculating trust values is $\mathcal{O}(v(K))$, where $K = |S_x| \times |S_y|$, showing the proposed scheme remains computationally efficient as v is a linear function with respect to K.

4.2.4. *Community classification*

With the CS value, Fig. 9 classifies the evaluated community into two groups: convenient community (C-com) and distant community (D-com).

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 remarks on its cooperativeness.

Table 3
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

Table 4
Configuration of intra-community TM simulation.

Conf.	Description
Manager	Single one
Service types	s1∼s4
Node type	see in Table 5
Population of nodes	(t1∼4) × 4 = 16; (NC) × 2; (Re) × 2

NC = Newcomer, Re = Returner.

5. Simulation

In this section, we simulated trust evaluation within a single community and between different communities to verify the effectiveness of intra- and inter-community TM of the proposed framework.

As illustrated in Table 3, 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 5.1 and 5.2.

5.1. *Performance evaluation of intra-community TM*

This subsection is split into 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 4 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, →, *, ⊕, and ⊗), where →, ⊕, and ⊗ contain 2 service types each (excluding $s4$ since it is 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.

5.1.1. *Access control*

Two aspects of the AC phase performance will be analyzed: the calculation of DS values and the demonstration of re-entry to the original community/moving to a new community case.

DS calculation. As defined in Table 5, NC1 and Re1 possess the same functional services and PB, likewise for NC2 and Re2. Table 6 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.

Re-entry to the original community/moving to a new community case. As shown in Fig. 10, 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).

In Fig. 10(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. 10(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.

Table 5
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.

Table 6
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

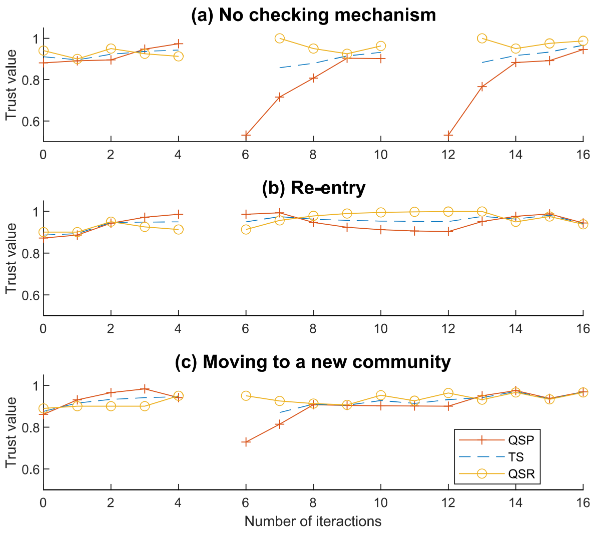


Fig. 10. Changes in trust values in three scenarios in the AC phase.

5.1.2. SP selection

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

Ranking SP by SS. Fig. 11 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.

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.

The performance of SP selection. 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. 12, 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.

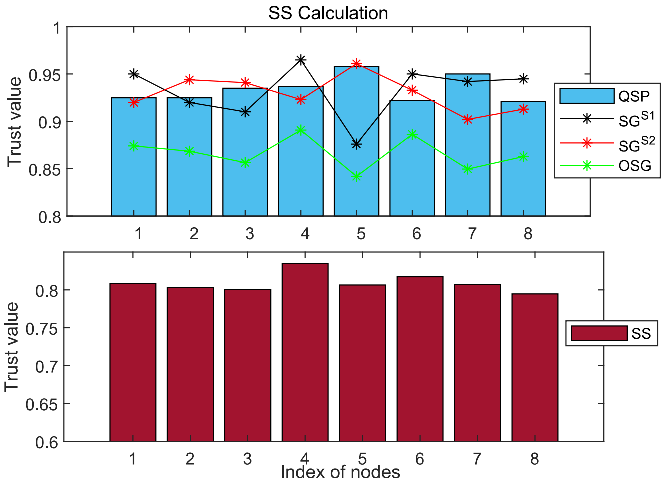


Fig. 11. SS calculation based on candidate nodes' OSG and QSP.

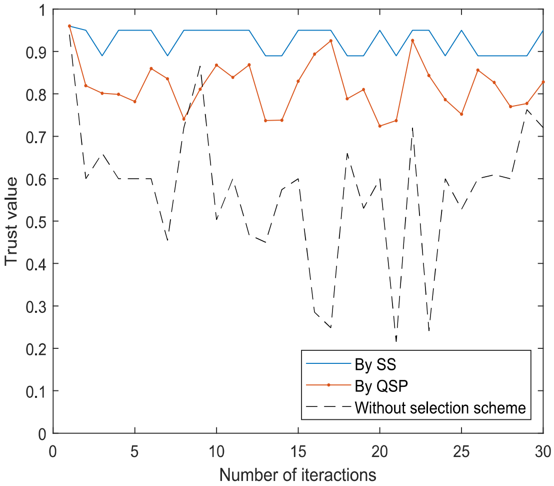


Fig. 12. Comparison of trust values between three scenarios of the SP selection.

Table 7
Feedback values for the attacker after the attack is launched.

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

5.1.3. 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 7 illustrates average feedback values from the attacked nodes and other nodes to evaluate the attacker.

In Fig. 13, 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.

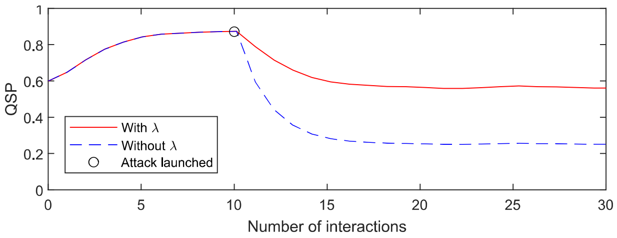


Fig. 13. Changes in QSP values with λ and without λ in presence of CBA attack.

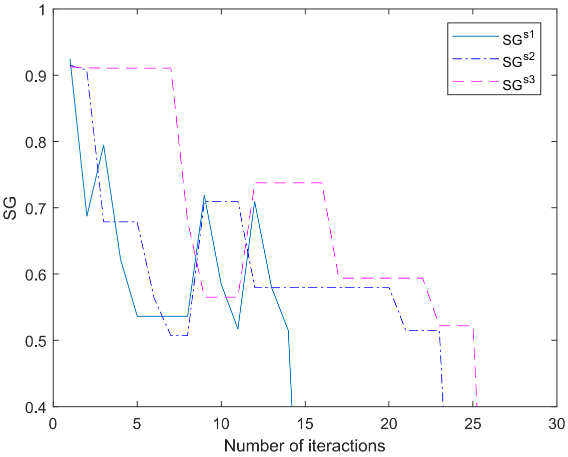


Fig. 14. Changes in SG values regarding service types in the presence of SBA attack.

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. 14 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} and SG^{s2} when misbehaving in $s3$. Eventually, SG values all drop under 0.5. As defined in , 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 [37]: 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. 15 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 , nodes can only gain a reputation through providing good services in a continuous way.

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 for good/bad services). As shown in the upper part of Fig. 16, 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

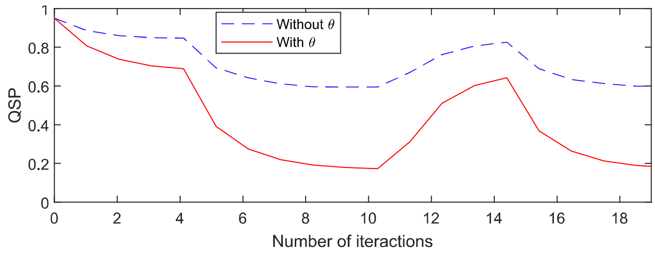


Fig. 15. Changes in QSP values with θ and without θ in the presence of OOA attack.

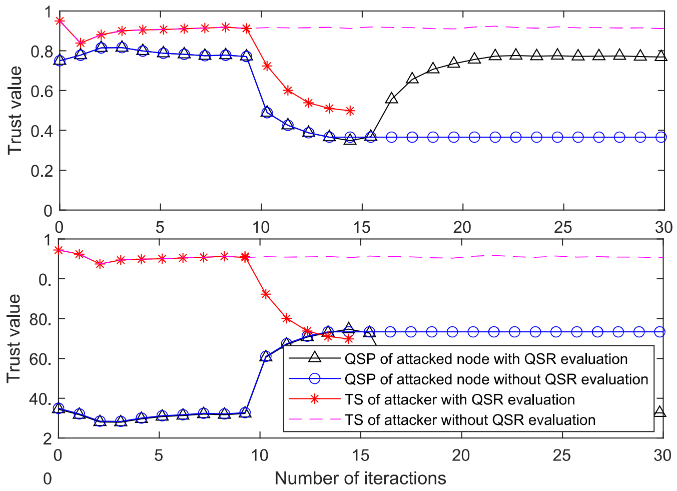


Fig. 16. Changes in trust values of both attacked and attacker nodes with QSR evaluation and without this evaluation in the presence of BMA and BSA attacks.

nodes' QSP drops after the isolation of the attacker node, in the lower part of Fig. 16, 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 and discussion. In this part, we compare the proposed model with the TMCoI-SIOT model [20] and CITM-IoT model [21] (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 address the aforementioned three attacks and 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. Since 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. Next, we focus on an F-score analysis by varying the percentage of malicious nodes (pm) to demonstrate the global performance evaluation, namely precision, recall, and critical success index (CSI) [38]. To better evaluate the behavior of the attacked node, we only illustrate the changes in the trust value of the attacked SP.

Precision, recall, and CSI are defined as follows:

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

$$\text{CSI} = \frac{tp}{tp + fp + fn}$$

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, these F-scores enable verification measures of the accuracy, sensitivity, and effectiveness of the proposed trust scheme.

Fig. 17 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

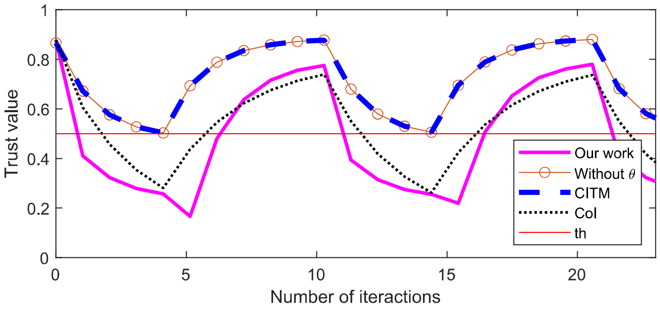


Fig. 17. Changes in trust values of the attacker node in the presence of OOA attack.

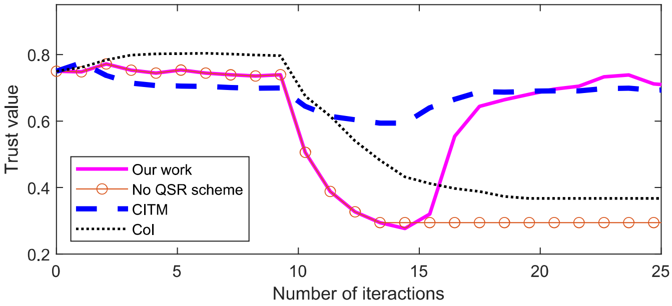


Fig. 18. Changes in trust values of the attacked node in the presence of BMA attack.

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. 18 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. 19, 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, all three F-scores are given zero in the CITM model, i.e., no OOA attackers are detected, and this is justified by the lack of mechanism against OOA in the CITM model. We can also notice that F-scores are close to zero when the population of malicious nodes exceeds 50%. Similarly, Fig. 20 visualizes F-scores' performance under the BMA attack by varying the population of malicious nodes. We can observe that precision, recall, and CSI values 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 remains low 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. 19 and 20 decrease significantly when the population of malicious nodes reaches 50%.

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 4.1.6. 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 summarize, 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.

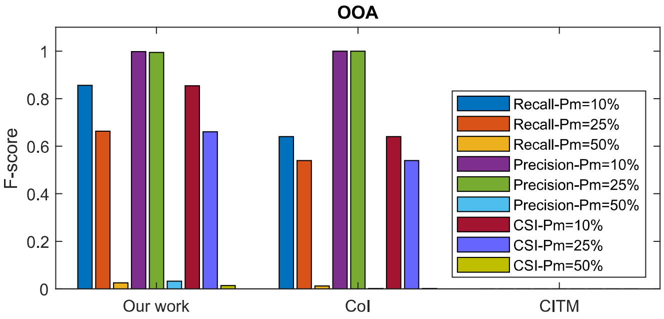


Fig. 19. Performance of F-scores in the presence of OOA attack.

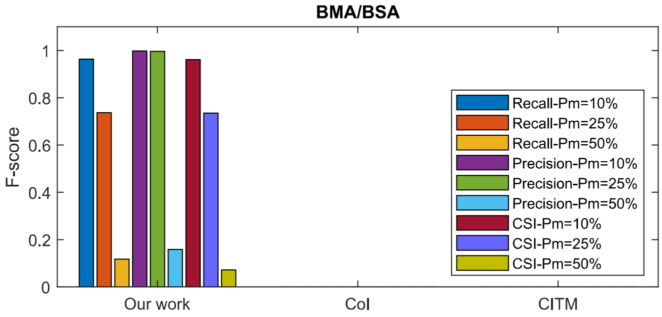


Fig. 20. Performance of F-scores in the presence of BMA attack.

5.2. Inter-community TM

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. On the other hand, nodes from p2 are forced to misbehave in p1.

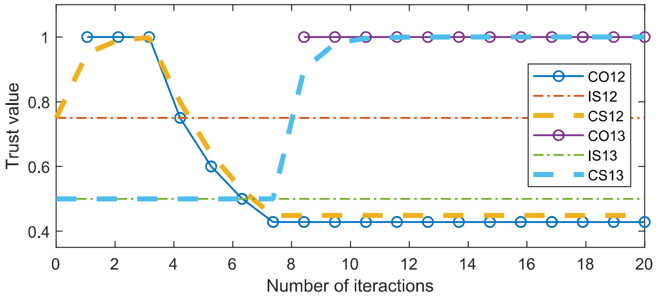


Fig. 21. Changes in IS, CO, and CS values of p1 evaluating p2 and p3.

Since we set p2 to perform in an uncooperative manner with p1, as can be seen in Fig. 21, 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. 9, 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 4.2.4.

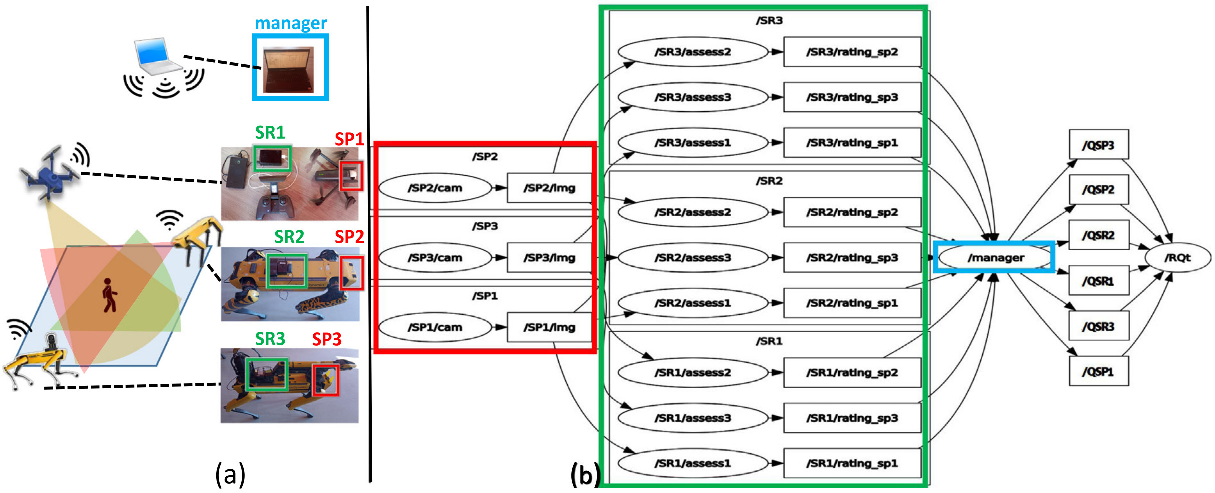


Fig. 22. Implementation by using ROS 2, where SR, SP, and trust manager are highlighted by corresponding colors: (a) Scenario and implemented hardware; (b) Software-level architecture generated by RQt in ROS 2.

6. Implementation

6.1. Scenario, implemented hardware and software

As illustrated in the left part of Fig. 22(a), a three-robot scenario is considered for implementation, where robots accomplish a common mission using their cameras to cooperatively monitor a human. The image transmission frequency is fixed at every 500 ms. As the preliminary implementation of the designed trust model, a HOG (Histograms of Oriented Gradient) [39] 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, other parameters remain the same as we defined in the simulation. Next, we are going to detail the implemented hardware and software.

Implemented hardware, as shown in Fig. 22(a), consists of an aerial drone (Anafi, ™Parrot[1]) and two quadruped ground robots (Spot, ™Boston Dynamics[2]). 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 ([3]). 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[4], and a 5 GHz Wifi access point[5] is set to enable all above-mentioned hardware's communication.

Compared with other Robotics Software Frameworks (RSF), ROS 2, an open-source software platform for robotics based on DDS (Data Distribution Service) [40], is more suitable for manipulation of nodes in IoT systems and for data exchange [41]. For this reason, ROS 2 is deployed for the implementation of our proposed model. The software architecture is depicted in Fig. 22(b) by a ***Node Graph***, which is composed of nodes,[6] 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 rating 3 SPs. Before computing QSR and QSP values, the manager will launch an approximate synchronization of nine ratings produced by the nine `access` nodes. After that, the node `manager` calculates the trust values by employing algorithms in Section 4.1.4. Finally, 3 QSR and 3 QSP values will be output by the `manager`.

6.2. Preliminary results on implementation

Via RQt in ROS 2, nodes' trust values are illustrated in Fig. 23 and Fig. 24, where OOA and BMA are launched, respectively. It can be noticed that SRs and SPs are working properly at the beginning in both figures, where QSP and QSR values are close to 1, meaning the well-performed ones receive high trustworthiness.

Next, the target human is forced to quickly move to produce environmental perturbation, at 18 and 30 s, respectively, which explains the changes in trust values before the 40 s in Fig. 23. It can also be seen that between 50 and 90 s, the camera SP3 of the

[1] https://www.parrot.com/fr/drones/anafi

[2] https://bostondynamics.com/products/spot/

[3] All utilized Raspberry pi card are of model 3-B+, https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/

[4] https://docs.ros.org/en/humble

[5] Model RT AX92U https://www.asus.com/fr/networking-iot-servers/wifi-routers/asus-gaming-routers/rt-ax92u/

[6] It must be noted that the node in ROS 2 totally differs from a node in IoT representing a device.

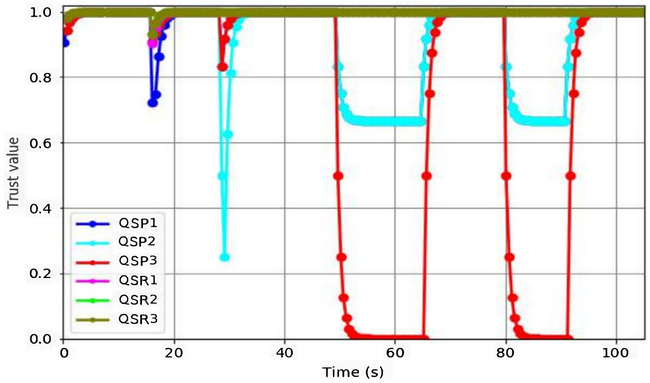


Fig. 23. Changes in QSR and QSP in the presence of OOA, launched by SP3 at the 50 s. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

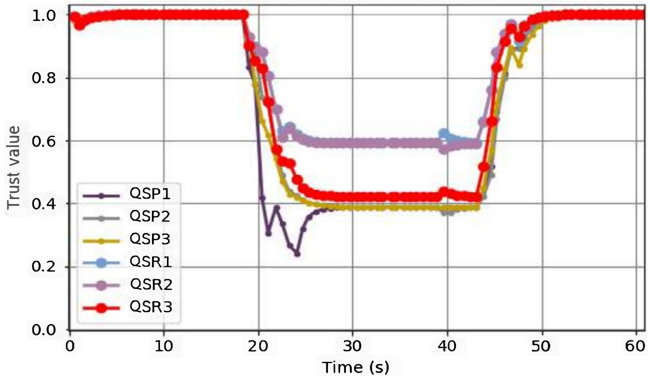


Fig. 24. Changes in QSR and QSP in the presence of BMA, launched by SR3 at the 18 s. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

quadruped robot is dedicated to an OOA attacker by switching between good and bad over time. Thus, the red curve representing the OOA attacker's $\mathtt{QSP3}$ decreases to 0 while $\mathtt{QSP1}$ and $\mathtt{QSP2}$ are also slightly lowered. As the gap between the trust values of the attacker and the well-behaved ones is sufficiently large, the OOA attacker is identified.

Results of one other type of attack are visualized in Fig. 24, where SR3 is fixed as the BMA attacker between 20 and 45 s to give 0.5 as the rating for all received services, no matter what SPs' real performances are. The attacker tries to ruin the reputation of good SPs by rating them negatively. The red curve representing the BMA attacker's $\mathtt{QSR3}$ goes down to 0.4, where $\mathtt{QSR1}$ and $\mathtt{QSR2}$ remain at 0.6. On the one side, this figure clearly demonstrates that the BMA attacker SR3 can be differentiated from non-malicious ones and it will also be effective for BSA since it misbehaves symmetrically as BMA does (Refer to Fig. 16). On the other side, all normal SPs are influenced harmfully in a way that their QSP values decrease 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 IoT system with more SRs and SPs, such negative effects caused by dishonest SR will be significantly reduced.

6.3. Discussion of implementation experience

As in the ***Node Graph*** illustrated in Fig. 22(b), it can be observed that each SP can only periodically (every 500 ms) share the same image with all SR, meaning that SP cannot perform differently with different SR. This indicates the manner in which CBA can be addressed at the practice level since CBA can be prevented systematically in our implementation by ROS 2 configuration. Furthermore, according to our implementation experience, we noticed a potential challenge if the number of devices further increases regarding the synchronization of the received rating on the manager side (although we addressed this issue by fixing the image transmission frequency in our implementation). This is because, in real IoT scenarios, not every node holds the same transmission frequency due to the heterogeneous nature of IoT nodes and networks. Consequently, the synchronization of rating collection becoming much more complicated to treat, especially when the number of devices increases.

7. Discussion and concluding results

This paper presents a role-based attack-resilient dynamic TM model containing both intra- and inter-community trust evaluation, which is suitable for community-driven 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. For this, we developed a three-phase mechanism to measure the cooperativeness between communities. 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. Unlike the existing works that insufficiently considered role-based assessment, our proposed model enables the evaluation of both SP and SR roles and also provides a global opinion combining both sides' qualities and workload to determine the nodes' nature in terms of service. On the other hand, due to the architecture design of our model, the trustworthiness within the community is monitored in a locally centralized manner, meaning 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, as 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 scheme, 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 OOA, BMA/BSA, and CBA. We can also observe that the negative perturbation from the malicious SR is much more harmful than the malicious SP side. For future work, we are interested in considering NCA and SBA in the implementation, as well as testing the proposed trust model more comprehensively by extending the size of the system with more IoT nodes with a more mobile scenario.

CRediT authorship contribution statement

Runbo Su: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Arbia Riahi:** Writing – review & editing, Methodology, Conceptualization. **Enrico Natalizio:** Writing – review & editing, Validation, Methodology, Conceptualization. **Pascal Moyal:** Writing – review & editing, Supervision, Formal analysis, Conceptualization. **Amaury Saint-Jore:** Validation, Software. **Ye-Qiong Song:** Writing – review & editing, Methodology, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This research was partially funded by the Federation Charles Hermite and by the ANR grant MATCHES, ref ANR-18-CE40-0019.

References

[1] Andreja Rojko, Industry 4.0 concept: Background and overview, Int. J. Interact. Mob. Technol. 11 (5) (2017).

[2] Ling Li, China's manufacturing locus in 2025: With a comparison of "Made-in-China 2025" and "industry 4.0", Technol. Forecast. Soc. Change 135 (2018) 66–74.

[3] 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/.

[4] Sana Alam, Shehnila Zardari, Shaheena Noor, Shakil Ahmed, Haralambos Mouratidis, Trust management in Social Internet of Things (SIoT): A survey, IEEE Access 10 (2022) 108924–108954.

[5] Avani Sharma, Emmanuel S. Pilli, Arka P. Mazumdar, Poonam Gera, Towards trustworthy Internet of Things: A survey on trust management applications and schemes, Comput. Commun. 160 (2020) 475–493.

[6] Ghani Ur Rehman, Anwar Ghani, Muhammad Zubair, Muhammad Imran Saeed, Dhananjay Singh, SOS: Socially omitting selfishness in IoT for smart and connected communities, Int. J. Commun. Syst. 36 (1) (2023) e4455.

[7] Lina Xu, Rem Collier, Gregory M.P. O'Hare, A survey of clustering techniques in WSNs and consideration of the challenges of applying such to 5G IoT scenarios, IEEE Internet Things J. 4 (5) (2017) 1229–1249.

[8] Sasikumar Asaithambi, Logesh Ravi, Hossam Kotb, Ahmad H. Milyani, Abdullah Ahmed Azhari, Senthilkumar Nallusamy, Vijayakumar Varadarajan, Subramaniyaswamy Vairavasundaram, An energy-efficient and blockchain-integrated software defined network for the industrial Internet of Things, Sensors 22 (20) (2022) 7917.

[9] Arun Kumar Sangaiah, Amir Javadpour, Forough Ja'fari, Weizhe Zhang, Shadi Mahmoodi Khaniabadi, Hierarchical clustering based on dendrogram in sustainable transportation systems, IEEE Trans. Intell. Transp. Syst. (2022).

[10] Naseem Ibrahim, 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.

[11] Ray Chen, 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.

[12] Yan Lindsay Sun, Zue Han, Wei Yu, K.J. Ray Liu, 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.

[13] Runbo Su, Arbia Riahi Sfar, Enrico Natalizio, Pascal Moyal, Ye-Qiong Song, Ensuring trustworthiness in IoIT/AIoT: A phase-based approach, IEEE Internet Things Mag. 5 (2) (2022) 84–88.

[14] Yosra Ben Saied, Alexis Olivereau, Djamal Zeghlache, Maryline Laurent, Trust management system design for the Internet of Things: A context-aware and multi-service approach, Comput. Secur. 39 (2013) 351–365.

[15] Ayesha Altaf, Haider Abbas, Faiza Iqbal, Abdelouahid Derhab, Trust models of Internet of Smart Things: A survey, open issues, and future directions, J. Netw. Comput. Appl. 137 (2019) 93–111.

[16] Runbo Su, Arbia Riahi Sfar, Enrico Natalizio, Pascal Moyal, Ye-Qiong Song, 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.

[17] A. Meena Kowshalya, M.L. Valarmathi, Trust management for reliable decision making among social objects in the Social Internet of Things, IET Netw. 6 (4) (2017) 75–80.

[18] Brennan Huber, Farah Kandah, DECAY: Dynamic evaluation and component analysis for enhancing trust management, in: 2024 IEEE International Conference on Consumer Electronics, ICCE, IEEE, 2024, pp. 1–6.

[19] Ammar Ayman Battah, Youssef Iraqi, Ernesto Damiani, A trust and reputation system for IoT service interactions, IEEE Trans. Netw. Serv. Manag. 19 (3) (2022) 2987–3005.

[20] Oumaima Ben Abderrahim, Mohamed Houcine Elhdhili, 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.

[21] Mohammad Dahman Alshehri, Farookh Khadeer Hussain, Omar Khadeer Hussain, Clustering-driven intelligent trust management methodology for the Internet of Things (CITM-IoT), Mob. Netw. Appl. 23 (3) (2018) 419–431.

[22] S. Feslin Anish Mon, S. Godfrey Winster, R. Ramesh, Trust model for IoT using cluster analysis: A centralized approach, Wirel. Pers. Commun. 127 (1) (2022) 715–736.

[23] Claudio Marche, Luigi Serreli, Michele Nitti, Analysis of feedback evaluation for trust management models in the Internet of Things, IoT 2 (3) (2021) 498–509.

[24] Pintu Kumar Sadhu, Venkata P. Yanambaka, Ahmed Abdelgawad, Internet of things: Security and solutions survey, Sensors 22 (19) (2022) 7433.

[25] Stavros Salonikias, Antonios Gouglidis, Ioannis Mavridis, Dimitris Gritzalis, Access control in the industrial Internet of Things, in: Security and Privacy Trends in the Industrial Internet of Things, Springer, 2019, pp. 95–114.

[26] Luigi Atzori, Antonio Iera, Giacomo Morabito, Michele Nitti, The Social Internet of Things (SIoT)–when social networks meet the Internet of Things: Concept, architecture and network characterization, Comput. Netw. 56 (16) (2012) 3594–3608.

[27] Charith Perera, Arkady Zaslavsky, Peter Christen, Dimitrios Georgakopoulos, Context aware computing for the Internet of Things: A survey, IEEE Commun. Surv. Tutor. 16 (1) (2013) 414–454.

[28] 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, in: VEHITS, SciTePress, INSTICC, 2024, pp. 276–283.

[29] Asrin Vakili, Nima Jafari Navimipour, Comprehensive and systematic review of the service composition mechanisms in the cloud environments, J. Netw. Comput. Appl. 81 (2017) 24–36.

[30] Akseer Ali Mirani, Gustavo Velasco-Hernandez, Anshul Awasthi, Joseph Walsh, Key challenges and emerging technologies in industrial IoT architectures: A review, Sensors 22 (15) (2022) 5836.

[31] Claudio Marche, Michele Nitti, Can we trust trust management systems? IoT 3 (2) (2022) 262–272.

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

[33] Shaozhong Zhang, Dingkai Zhang, Yaohui Wu, Haidong Zhong, Service recommendation model based on trust and QoS for social Internet of Things, IEEE Trans. Serv. Comput. (2023).

[34] Runbo Su, Arbia Riahi Sfar, Enrico Natalizio, Pascal Moyal, Ye-Qiong Song, 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.

[35] Xianming Huang, Intelligent remote monitoring and manufacturing system of production line based on industrial Internet of Things, Comput. Commun. 150 (2020) 421–428.

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

[37] Wafa Abdelghani, A Multi-Dimensional Trust-Model for Dynamic, Scalable and Resources-Efficient Trust-Management in Social Internet of Things (Ph.D. thesis), (2020TOU30231) Université Paul Sabatier - Toulouse III ; Université de Sfax (Tunisie), 2020.

[38] Gashirai K. Mbizvo, Andrew J. Larner, F*, an interpretable transformation of the F measure, equates to the critical success index, 2023, Preprints.

[39] Navneet Dalal, Bill Triggs, Histograms of Oriented Gradients for Human Detection, in: Cordelia Schmid, Stefano Soatto, Carlo Tomasi (Eds.), International Conference on Computer Vision & Pattern Recognition, CVPR '05, vol. 1, IEEE Computer Society, San Diego, United States, 2005, pp. 886–893.

[40] Steven Macenski, et al., Robot Operating System 2: Design, architecture, and uses in the wild, Science Robotics 7 (66) (2022).

[41] Pablo Iñigo-Blasco, et al., Robotics software frameworks for multi-agent robotic systems development, Robot. Auton. Syst. 60 (6) (2012) 803–821.