D. Ferraris, C. Fernandez-Gago, Y. Assouyat, H. Labiod, W. Haiguang and J. Lopez, "Trust dynamicity for IoT: How do i trust your social IoT cluster?", *Internet of Things*, vol. 30, 2025.
http://doi.org/10.1016/j.iot.2025.101529
NICS Lab. Publications: https://www.nics.uma.es/publications

Trust Dynamicity for IoT: How do I Trust your Social IoT Cluster?

Davide Ferraris[a,*], Carmen Fernandez-Gago[a], Younes Assouyat[a], Houda Labiod[b], Wang Haiguang[c], Javier Lopez[a]

[a] *Network, Information and Computer Security Lab, University of Malaga, 29071, Malaga, Spain*
[b] *Huawei Technologies France, Shield Lab Paris, 18 quai du Point du Jour, 92100 boulogne-billancourt, France*
[c] *Huawei Shield Lab Singapore, 20 Science Park Road, Teletech Park, Singapore Science Park II*

Abstract

The Social IoT (SIoT) enhances the traditional Internet of Things (IoT) by integrating social relationships between device owners. This paper presents a dynamic trust framework specifically designed for SIoT environments, with the objective of providing security against malicious attacks targeting IoT devices. The framework offers a multi-dimensional analysis of trust, emphasizing the behaviours and contextual interactions of domestic devices. A prototype implementing the proposed framework is introduced and evaluated across three different use cases showing how to assess device reputation, enable dynamic device integration, and secure communication within device clusters. The evaluation results highlight the framework's ability to enhance the reliability of device interactions and ensure seamless interoperability among devices utilizing different trust models. This significant improvement in trust management contributes to more secure and efficient SIoT operations. The findings underscore the critical role of dynamic trust adaptation and interoperability in creating a cohesive and secure SIoT ecosystem.

Keywords: Trust, Dynamicity, IoT, Social IoT, Network, Interoperability

1. Introduction

The advent of the Internet of Things (IoT) has ushered in a new era of interconnected devices that collaborate to enhance various aspects of our daily lives.

[*] Email: ferraris@uma.es

An evolution of such an ecosystem is called Social IoT (SIoT), where it is also considered the social relationship among the owner of the IoT devices. Thus, within this expansive network, SIoT clusters have emerged as dynamic ecosystems where trust plays a pivotal role [1]. These clusters are intricate groupings of devices and entities that interact to achieve common goals, fostering a collaborative and interconnected environment.

In SIoT clusters, the concept of trust is not static, but dynamic and continually evolving based on real-time interactions, device behaviour, and contextual changes [2]. The dynamic nature of trust is particularly pronounced in environments where devices join or leave the network, update their software, or encounter varying levels of reliability [3]. Understanding and adapting to this dynamicity is crucial for ensuring the integrity and security of SIoT clusters [4].

Moreover, the diverse nature of the devices within these clusters introduces the need for trust interoperability [5]. Given that devices may employ different trust models, protocols, or standards, ensuring seamless interaction and collaboration becomes a significant challenge. Interoperability is essential for devices with disparate trust mechanisms to communicate effectively, thus fostering a cohesive and trustworthy SIoT ecosystem [6].

In this dynamic landscape, addressing the dual challenges of dynamic trust adaptation and trust interoperability becomes imperative. This introduction sets the stage for exploring the intricacies of trust dynamics within SIoT clusters and delving into strategies to achieve dynamicity and interoperability among various trust models. As we consider different scenarios, it becomes evident that the ability to adapt to the evolving landscape of trust is fundamental to unlock the full potential of SIoT clusters.

The solution we have designed is mainly composed of a trust model that identifies different trust levels. We take into consideration several parameters such as the risk, the previous trust value (if there are exits), and the context, which is a well-known parameter fundamental for both IoT and trust [7]. The trust model is based on [3] and is improved in order to consider more scenarios in addition to a smart home (i.e., smart city). Such a model will analyze the trust parameters in order to accept or keep an entity in a SIoT cluster. In fact, according to the capacity of the SIoT ecosystem, we enable the trust model to consider users with a social relationship to be connected through their IoT clusters, where trust will be the key that allows them to interact with the devices of a friend/partner. In this situation, context is crucial and trust metrics will determine whether a user can be trusted to interact with another friend/partner's device or not.

In this paper, we present a dynamic trust model that adapts to changes of trust over time in different SIoT clusters, considering also interoperability as a key factor.

The structure of the paper is as follows. In Section 2, we introduce the background and analyse the related work. Then, in Section 3, we explain the motivation and describe how we combine dynamicity and interoperability. Section 4 describes the novel trust framework and next we present its application to a smart home environments in Section 5. In Section 6 we validate the proposed prototype. Finally, in Section 7, we conclude the paper and describe the future work.

2. Background and Related Work

Trust management in the IoT has been an active area of research in recent years, as the increasing interconnections of devices and the growing volume of data generated by IoT devices pose significant security and privacy challenges [8]. Numerous researchers have investigated various aspects of trust management in IoT, focusing on trust dynamics, trust interoperability, and trust-based service management as we are going to see next.

2.1. Security and Trust in IoT

The IoT environment is a worldwide network of interconnected entities that can be located, usable and readable through the Internet. Such systems increase the complexity of a connected world and guaranteeing security is a difficult task. In fact, it is expected that these objects will have to interact with each other often under conditions of uncertainty. Mechanisms to resolve this lack of information are needed and trust can help address this need [9]. Related to trust, reputation is more objective and it can be a parameter for trust decision [10]. The heterogeneity and dynamicity of IoT have raised questions and led to some possible architectures being put forward. Roman et al. [11] identified four main architectures, each of them have their strengths and weaknesses. These architectures are centralised, collaborative IoT, connected Intranets of Things and distributed. In a centralised approach, a gateway such as a smart home hub manages a group of devices (mostly passive), with the primary control gateway and logic being in the hub itself. The major risk with this architecture is that, when the smart hub is compromised or is not working properly, the whole architecture fails.

In order to protect such architectures, several protocols and works have been proposed. A taxonomy of such challenges and possible solutions have been proposed in [12]. This work has been the basis of further research such as the one proposed by [13], where the authors consider the different layers of the ISO/OSI architecture focusing on the issues that can be encountered in each of them. They focused also on the communication protocols more used in IoT (i.e., 6LoWPAN, MQTT) and made clear the importance of placing security and trust at the centre. Another interesting work, such as the one published in [14], focused on the user perception of security in IoT by the final users. In fact, with the growing of IoT, users are becoming crucial players and need to be aware about the possible issues as we will discuss later.

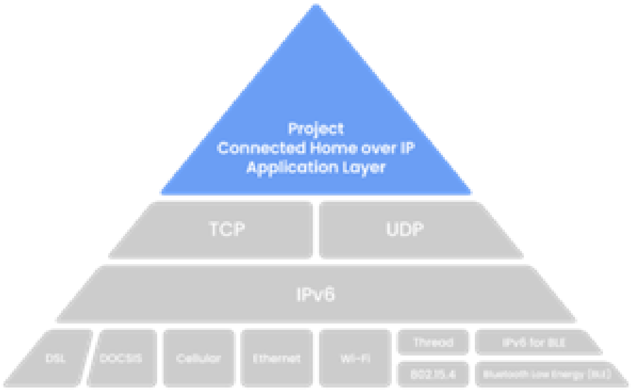


Figure 1: Matter: Application and network stack layers

In order to find such balance between user perspective and security enhancement, in [15] the authors proposed several solutions. One of them is Matter [16], which is a unified IP-based protocol proposed to securely connect smart devices and enable smart home ecosystems. The standard aims to address one major challenge facing the smart home industry such as the lack of interoperability between devices from different manufacturers focusing on the top layer proposed in Figure 1. The protocol defines the application layer that will be deployed on devices as well as the different link layers to help maintain interoperability. To provide security and privacy, Matter supports security by design and zero trust principles (i.e., AES-CCM128, X.509 certificate), rule-based access control, verification of software integrity and CSA certification [16]. However, such technology is yet a novel IoT standard, but some weaknesses have been identified leading adversaries to exploit in mostly targeted attacks. Nevertheless, Matter architecture enhances the possibilities to develop a trust model to overcome such issues and be consistent during communications (i.e., NS3 implementation [17]). All the previous works have been the basis for the work proposed in [18] where trust and security have been considered as crucial for the IoT environment and its extension known as SIoT.

2.2. Dynamic Trust Management in Multi-service IoT Environments

Trust Management plays a crucial role to maintain security and reliability in the IoT. It aims to maintain reliability in a system by ensuring the secure exchange of information and accomplishing various decision-making tasks (reliable service composition, secure routing, device authentication, access control, etc).

The state of art shows that most of the work aimed at facilitating reliable service composition and management in IoT [19]. However, issues related to inherent characteristics of IoT systems like heterogeneity, limited resource power, scalability or context awareness have not been adequately addressed.

Dynamicity has been considered by most of the works by emphasising less on the heterogeneity of devices and networks. Also, in IoT, devices can join or leave the network at any moment of time [3]. Although, a significant amount of work has been done to consider this issue, efforts are required to capture device dynamicity paying attention to other issues. Also, most of the works considers static weight assignment for different trust attributes or factors in trust computation. An effective and adaptive methodology must be used for dynamic assignment of weights reducing the chances of application dependence. Concerning heterogeneity of devices and networks, very few researchers considered heterogeneity while designing trust models. As for context awareness, it is essential to pay attention to context-awareness while computing trust in a multi-service IoT environment.

2.3. Trust Management in the SIoT

The architectures and the protocols discussed earlier have been the basement for the creation of an extension of IoT which is growing due to the possibility to interconnect devices belonging to different users having social relationships among them [20]. Thus, the concept of SIoT has gained significant traction in recent years, as researchers and industry experts explore the potential of enabling interconnected devices to interact and collaborate in a socially aware manner [21]. Trust management in SIoT is critical for ensuring secure and reliable data exchange among devices and entities. Numerous researchers have investigated various aspects of trust management in SIoT, focusing on trust dynamics, trust interoperability, and trust-based service management [22].

Several studies have explored the dynamic nature of trust in SIoT, emphasizing the need for adaptive trust management mechanisms. Chen et al. [23] proposed a trust management model based on fuzzy reputation, considering the time-varying nature of trust. Bao and Chen [24] presented a trust management approach that incorporates both social trust and Quality-of-Service (QoS) trust metrics, adapting trust evaluations based on contextual information.

Achieving trust interoperability across different SIoT clusters is essential for seamless communication and collaboration. Saied et al. [25] proposed a context-aware and multiservice trust management approach that facilitates trust exchange and recognition between different SIoT domains. Zhang et al. [26] developed a cross-domain trust management framework based on blockchain technology, enabling secure and verifiable trust exchange.

Trust plays a crucial role in service management within SIoT environments. Sun et al. [27] proposed a trust-based service selection framework for SIoT, considering both QoS and social trust factors. Wang et al. [7] developed a trust-based service recommendation mechanism for SIoT, utilizing social network analysis to identify trustworthy service providers. Other important works consider Context-aware trust models incorporating contextual information, such as location, time, and purpose of interactions, to make more informed trust decisions [7].

The works presented in this section represent a crucial interest of the research community in enhancing security solutions for IoT and SIoT. However, to the best of our knowledge, none of such works have presented a whole trust model that can enhance security and communication among devices in SIoT. With the consideration of interoperability and dynamicity issues, our purpose is to fill this gap.

3. Motivation

As we discussed earlier, considering both dynamicity and interoperability is essential when addressing trust within the IoT due to several significant reasons, but at the same time might pose challenges for the inclusion of these aspects. In order to design the proposed solution, we have identified the following ones:

- **Evolving IoT Ecosystem**: IoT environments are inherently dynamic, characterized by constantly changing device states, network conditions, and data context. Devices may enter or exit the network, update their software, or experience fluctuations in trustworthiness. To maintain trust in such a dynamic landscape, trust models and mechanisms need to adapt in real-time to these changes [28].

- **Heterogeneity of Devices**: IoT ecosystems comprise a diverse range of devices with varying capabilities and trust requirements. These devices may employ different trust models and protocols. Ensuring interoperability allows devices with distinct trust mechanisms to interact seamlessly, fostering trust across the IoT ecosystem [29, 30].

- **Multi-Domain IoT**: IoT applications often span multiple domains, such as healthcare, smart cities, smart homes or industrial automation. Each domain may utilize unique trust models and standards. Interoperability is vital for trust to extend across these domains while accommodating their specific trust requirements [31, 32].

- **Security and Privacy**: The dynamic nature of IoT environments introduces security and privacy challenges. Cyber threats and vulnerabilities can emerge at any time, necessitating dynamic adaptation of trust models to counter these threats. Interoperability must also consider privacy concerns and ensure data protection during trust-related interactions [33].

- **Resource-Constrained Devices**: Many IoT devices operate with limited computational resources, making dynamic trust adaptation a necessity. These resource-constrained devices may need to adjust their trust levels or security measures to optimize resource utilization while maintaining trustworthiness [34].

- **Real-Time Decision-Making**: IoT applications often require real-time decision-making based on trust assessments. Dynamic trust models that account for the evolving context and the device states are crucial to make accurate and timely decisions [3].

- **User Experience**: Dynamicity and interoperability are vital for creating a seamless and user-friendly IoT experience. Users expect their IoT devices and applications to work reliably and securely accross various scenarios, and trust plays a pivotal role in achieving this [35].

In summary, with the diversification of network functions and services, security authentication alone cannot meet the requirements of trust between network devices and services. Management of information interaction between IoT and, for instance, SIoT devices should be refined according to the trust classification. Dynamicity of trust is a prerequisite as trust may change over time and adaptive trust modelling and adaptive trust management mechanisms should be deployed.

In the following subsections, we will deeply analyze dynamicity and interoperability challenges according to the SIoT ecosystem.

4. Proposed Trust Framework

In this Section, we will first discuss about the idea of applying dynamicity and interoperability of trust in the SIoT. We will then present three possible states defining the evolution of each SIoT entity: join, stay and leave.

4.1. Dynamic, Adaptive and Interoperable SIoT Trust Model

First of all, we consider a SIoT network as a composition of clusters with IoT devices belonging to different owners with social relationship among them. Figure 2 depicts our view of a cluster, which is composed of smart devices belonging to an owner.

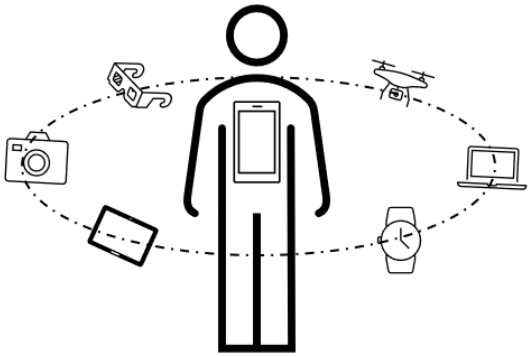


Figure 2: Cluster belonging to a smart devices owner

These models should continuously evaluate devices and network conditions, oversee data quality and context, and respond to evolving threats. Trust management systems must strike a balance between security and usability, providing a foundation for reliable communication and data integrity while adjusting to the ever-changing landscape of IoT environments. This adaptability is crucial for maintaining trust and security in the fast-paced and diverse IoT environment. These considerations are crucial for a paradigm such as the IoT where we implement such adaptive trust models enforcing other users with more connections (i.e., more devices are available to another users) or reducing the possibilities for a user of interacting with some devices or, as a final possibility, removing the user from the social circle.

In our trust model, we consider dynamicity and interoperability as two fundamental parts, which are strictly connected as shown in Figure 3. There, **1** represents the first state where trust computation is monitored and in the case a change occurs, it is represented by the transition **1a**. In this case, a modification of the value as occurred due to the stay model as we will see later and a new value of trust is determined. However, it is possible that the trust value must be translated into another value, for example if we have a trust value related to an evaluation model and we want to translate it into a decision model value, the diagram will pass from the state **Dyn** to the state **Int** by the transition **2**. Here, the models will be translated according to the Interoperable Trust Repository (ITR) presented in [5] which provide interoperability between evaluation and decision models. In this state number **3**, such translation happens and when it is finished, there is the transition **4** coming back to the Dyn state (**1**) where the trust value now is the one translated, so any modification will be performed according to the new value/-model. In the case a modification occurs after a trust monitoring, there will be a change of this value following the transition **1a** as explained earlier.

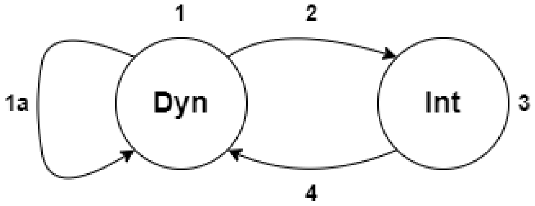


Figure 3: Relationship between Dynamicity and Interoperability in our model

4.2. Trust Management of SIoT Device States

In the following part, we will explain more deeply the states related to dynamicity and interoperability. Starting from dynamicity, in our trust model we define four possible states: Join, Stay, Leave and Quarantine. We discuss them in the following subsections. Their relationships are shown in Figure 4.

In this figure, we can see that the starting point will be the Join state and the final point will be after the Leave state. It should be possible to reach the final state if the output of the Join phase is to deny the new entity to join the network. Otherwise, the flow will reach the Stay state. From this state there are three possible outputs: to remain in Stay, move to Leave or move to Quarantine.

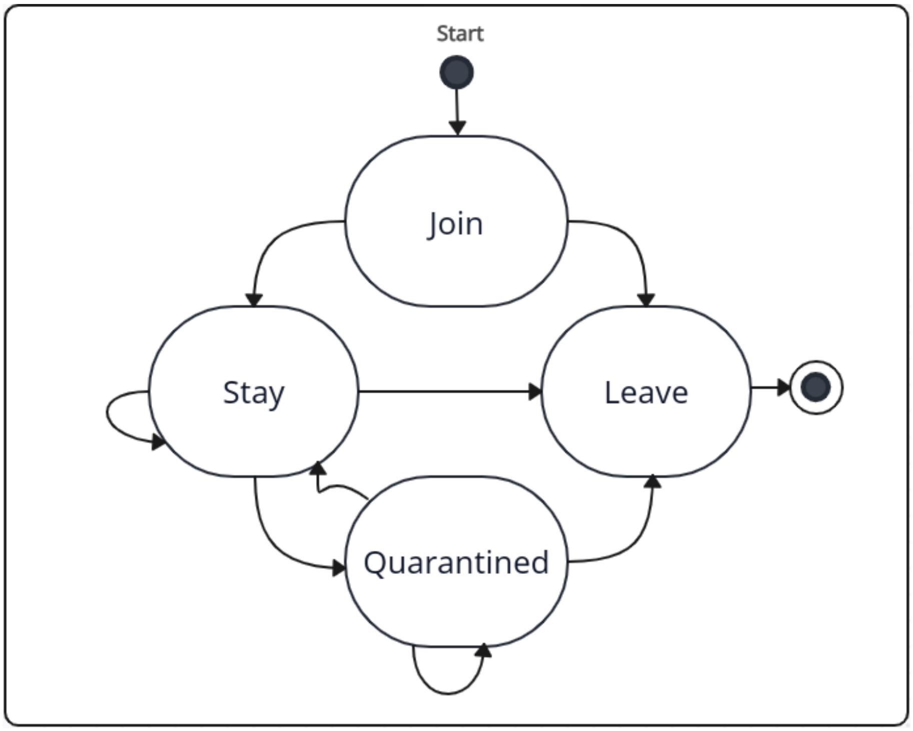


Figure 4: State Diagrams Relationships: Join, Stay, Leave and Quarantine

However, now we briefly explain how the trust model works and which parameters are important in order to compute the trust level in order to trust or not an entity. Such parameters have been extracted and extended from [3]. They are:

1. **Trust DB**: It is represented by an old trust value (i.e., reputation in the case of an evaluation model or trust level in the case of a decision model) in the case the device has been previously added to the network and left or put in quarantine . Otherwise, this value will be considered as the medium one (i.e., in the case of an evaluation model with a range 0-1, it will be given a value of 0.5) and will be computed with the following parameters.
2. **Context**: it represents the value of importance related to the scope of the joining device and its functionalities. We give to it values from 1 to 4.
3. **Risk**: This parameter is composed by three factors and Risk values are 1 to 9 for each parameters (Likelihood, Severity, Detectability):
    - *Severity*: it represents the impact of the event and its consequences to the device integrity and the consequences to the network.

   - *Likelihood*: it represents the probability that the event will occur.
   - *Detectability*: it represents the possibility to detect the event including the possibility of mitigate it.

4. **Threat DB**: it represents a repository containing the known attacks related to the device. In the case no threats are known there will be a lower value, on the other hand, if a powerful attack is known and not already solved, this will impact on the final decision.

Such parameters have been extrapolated by analyzing established trust and risk assessment frameworks in both IoT and broader cybersecurity contexts such as Dempster-Shafer theory of evidence [36].

All the parameters are strongly connected among them and they will be normalized according to the range chosen for trust value.

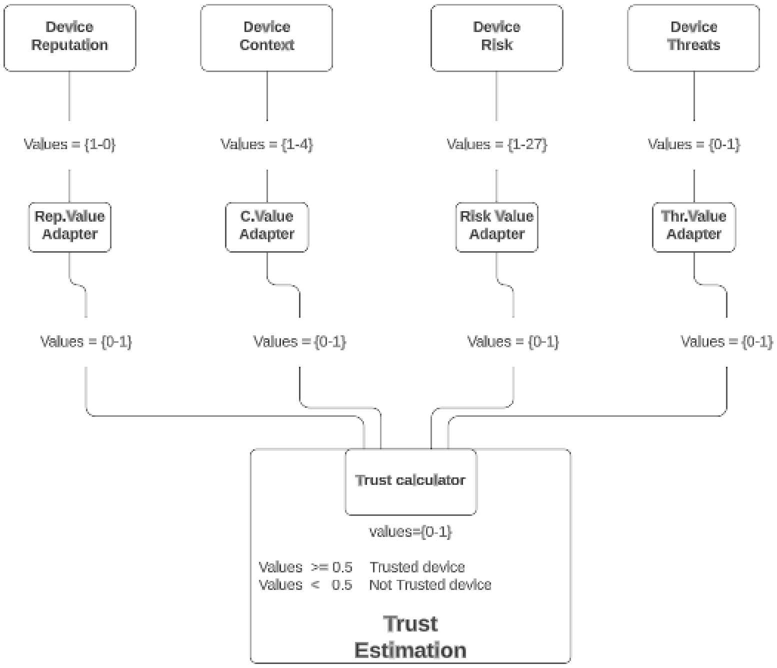


Figure 5: Computation and normalization of the four parameters

The trust calculator presented in Figure 5 can be represented with the following equation:

$$\text{Trust} = \text{Rep} * \text{Context} * \text{Threat} * \text{Risk}$$

Where * represents the operation related to the trust computation. It can be modified according to the importance that we can give to each parameter. As a default representation, the operation is a summation and then a division for the number of the parameters as we gave them the same importance in the default setting. However, it is possible to consider or not consider a parameter as we allow to adjust their importance. The rationale behind this choice aligns with principles found in well-known security and risk assessment methodologies, such as the Common Vulnerability Scoring System (CVSS) [37] and the FAIR (Factor Analysis of Information Risk) framework [38].

Finally, in order to restrict the possible outcomes, the minimum threshold allowing a device to join/stay or not in a cluster can be adjusted by the cluster owner. It depends on the trust perception of the other cluster's owner. In Figure 5, we consider it as 0.5, where values range from 0 to 1, but this value can be modified. In fact, such threshold is a subjective choice that depends on the owner's perception and it can be modified in order to reduce the trusted values (i.e., use a value above 0.5) or improve it (i.e., use a value below 0.5). We decided to use as threshold the values 0.5 to 1, to reflect other similar works, when a new device usually has the medium value when accessing a network [39].

In the following subsections, we will describe Join, Stay, Leave and Quarantine states.

4.2.1. Join

When a device expresses interest in joining the network, whether it is external or internal, it is assigned an ID. We consider the possibility to have networks similar to the one proposed in [3]. Coming back to the ID, said ID is unique and related to the fingerprint of the device, in the next phase, the history of the device's reputation is verified, obviously if the device has never interacted with the ecosystem, it should have no information and it is considered as a new device in case it already has a history, only the last reputation that the device has within the ecosystem will be taken into account, this reputation will be one of the four metrics necessary to generate the new reputation. There is a reputation threshold that the device must overcome to be part of the ecosystem, if so, the reputation history of the device in question is fed back, while a profile is created for such a device the profile will be saved in both databases history database of the ecosystem devices. If the test is not passed, the reputation achieved remains registered in the reputation database. The registered reputation is considered for future joining attempts of such a device. It should be noted that the reputation threshold can be assigned dynamically by the ecosystem or as a fixed value.

4.2.2. Stay

Once inside the ecosystem the device remains under supervision, any change in the behaviour of the device is detected by and the reputation of the device is reassessed to ensure that the device continues to meet the security requirements network, if not, the device must go down into quarantine or it is permanently removed from the ecosystem. This situation is similar to the Join state, but it adds several possibilities. The first one is related to the monitoring state. If something occurs, trust estimation is performed in order to adjust the trust value. In fact, this modification can be necessary either the device has behaved maliciously (i.e., trust value will be lower than before) or non-maliciously (i.e., trust value will be higher than be-fore).

In this model, security is implemented to enhance the possibilities of put them in quarantine or disconnect them from other devices. In order to distinguish between a network device and a quarantined device, we consider it in the trust database where it is also contained the information related to the possibility that all the devices have presented an anomaly in their behaviours. However, we can consider the quarantine as a temporary and intermediary stage between the state of network device and device expelled from the network.

4.2.3. Quarantine

We have considered this case in order to provide a period of time to the entities which are behaving in suspect or malicious ways. Such period can be used to analyze better the data and to solve possible issues due to malfunctions, malware or firmware problems.

The threat DB is important in order to choose if putting an entity in quarantine and removing it from such state. In fact, in the case the shifting has been performed for detecting a vulnerability, such change of the state can be reverted once such vulnerability has been solved. In this case, a firmware or software update should be necessary for the considered device.

Another important element is the Trust DB, in fact in the case of malicious or doubtful behaviour of an entity, the quarantine can be a good option in order to check whether the anomalies continue or not. In a positive case, the Leave state is reached.

Furthermore, in the case an entity has been put into quarantine because of a known threat or risk, the only way to exit from this state and come back to the stay state is, for example, updating the firmware and/or software to solve the issue “immunizing” the device from the exploit of the threat. In this case, the value of the Threat DB will be updated and improved. Such modification, will improve also the trust value, allowing the device to re-join the network. Otherwise, if the problem cannot be solved, the only possibility to move from the quarantine state, is to move to the leave state.

4.2.4. Leave

This is the simplest case, but it is worth to be mentioned. In fact, this state can happen according to different situations:

1. After a malicious behaviour, the device instead to be put into quarantine, is banned from the network (i.e., in the case the threat and risk are very high and the network itself can be compromised entirely from the device). In this case, the information must be stored in the Trust DB in order to block the possibility for the device to join again the network in the future.
2. The device will leave the network temporarily, and it will join the network again in the future. In this case too, the stored values will allow the device to join the network again. In fact, we assume that the device is leaving "willingly" the network, with a positive trust value.
3. The device is discarded (i.e., the owner bought a newer version), thus we are certain that in the future the device will not join again the network. However, it is better to store the information for a certain amount of time (decided by the owner) in the Trust DB.

5. Application for Smart Home

In this section, we will apply the proposed framework in a smart home environment proposing a basic architecture and also three sequence diagrams according to the join, stay and leave states.

5.1. Technical Solution

Considering a device registers to a smart home system, device attestation evidence shall be provided. Such new device is evaluated based on its context, reputation, risk and threat level. After such consideration, a trust level is assigned to the device in order to accept it in the network.

In Figure 6, we exemplify the architecture in which we will show the sequence diagrams according to the three states mentioned earlier.

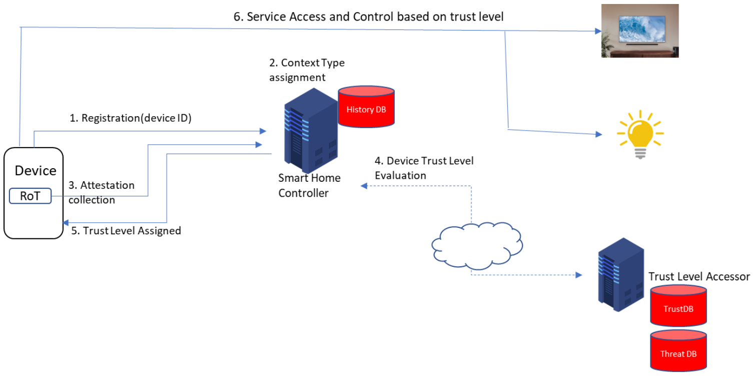

6. Service Access and Control based on trust level
2. Context Type assignment
History DB
1. Registration(device ID)
Device
RoT
3. Attestation collection
5. Trust Level Assigned
Smart Home Controller
4. Device Trust Level Evaluation
Trust Level Accessor
TrustDB
Threat DB

Figure 6: Smart Home Example and Communications

The device must be registered in the Smart Home Controller that will assign the context type to the selected device and checks if it has joined before the environment consulting the trust DB. After the device attestation and the trust evaluation, the device is assigned such trust value and the system will decide whether to add it or not to the network.

5.1.1. Join Example

In this part, we can check how the system works in the Join state. In Figure 7, we can see the steps.

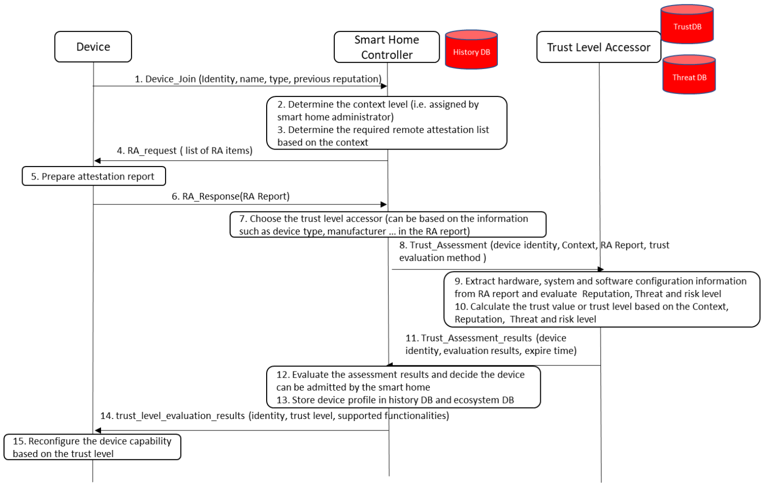

Device
Smart Home Controller
History DB
Trust Level Accessor
TrustDB
Threat DB
1. Device_Join (Identity, name, type, previous reputation)
2. Determine the context level (i.e. assigned by smart home administrator)
3. Determine the required remote attestation list based on the context
4. RA_request (list of RA items)
5. Prepare attestation report
6. RA_Response(RA Report)
7. Choose the trust level accessor (can be based on the information such as device type, manufacturer … in the RA report)
8. Trust_Assessment (device identity, Context, RA Report, trust evaluation method)
9. Extract hardware, system and software configuration information from RA report and evaluate Reputation, Threat and risk level
10. Calculate the trust value or trust level based on the Context, Reputation, Threat and risk level
11. Trust_Assessment_results (device identity, evaluation results, expire time)
12. Evaluate the assessment results and decide the device can be admitted by the smart home
13. Store device profile in history DB and ecosystem DB
14. trust_level_evaluation_results (identity, trust level, supported functionalities)
15. Reconfigure the device capability based on the trust level

Figure 7: Join Steps

When a new device attempts to join a smart home network, it initiates the process by providing its identity, name, type, and previous reputation to the Smart Home Controller. This initial step is crucial as it allows the controller to recognize and categorize the device based on its history and specifications. Following this, the Smart Home Controller assigns a context level to the new device, a task typically handled by the smart home administrator. This context level is vital as it dictates the security protocols and operational parameters that the device must adhere to within the network.

Once the context level is determined, the controller identifies the specific remote attestation (RA) items required from the device based on its assigned context. Subsequently, the controller sends a remote attestation request to the device, listing the required attestation items. The device then prepares a comprehensive attestation report containing all the requested information, ensuring that it meets the specified requirements.

The device responds by sending this attestation report back to the Smart Home Controller. The controller, equipped with this detailed report, proceeds to select an appropriate trust level accessor. This selection is based on the information provided in the RA report, such as the device type and manufacturer, ensuring that the trust evaluation is tailored to the specific characteristics of the device.

The trust level accessor then undertakes a thorough trust assessment, utilizing the device identity, context, RA report, and a chosen trust evaluation method. This process involves extracting detailed hardware, system, and software configuration information from the RA report. The accessor evaluates these details to ascertain the device's reputation, threat level, and overall risk.

Following this, the trust level accessor calculates the device's trust value or trust level, integrating the context, reputation, threat, and risk assessments into a cohesive evaluation. The results of this trust assessment, including the device identity, evaluation outcomes, and an expiry time for the assessment, are communicated back to the Smart Home Controller.

Armed with the trust assessment results, the Smart Home Controller evaluates whether the device meets the criteria for admission into the smart home network. This decision is critical in maintaining the network's security and operational integrity. If the device is deemed trustworthy, its profile is stored in both the history database and the ecosystem database, ensuring that its information is readily accessible for future reference.

The controller then sends the trust level evaluation results back to the device, detailing its identity, trust level, and the functionalities it is permitted to support within the network. Finally, the device reconfigures its capabilities based on the received trust level, ensuring that it operates within the defined parameters and adheres to the network's security protocols. This comprehensive process ensures that only trusted devices are integrated into the smart home ecosystem, maintaining a secure and efficient environment for all connected devices.

5.1.2. Stay Example

Then, as we described above, another important step is related to the stay part showed in Figure 8.

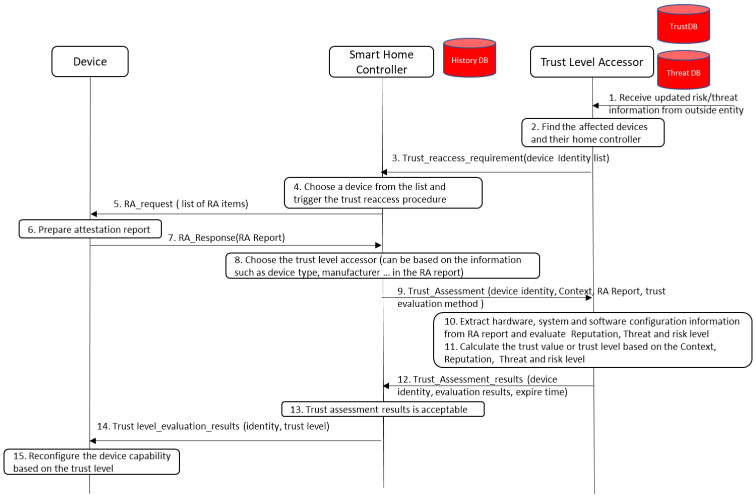

TrustDB
Device
Smart Home Controller
History DB
Trust Level Accessor
Threat DB
1. Receive updated risk/threat information from outside entity
2. Find the affected devices and their home controller
3. Trust_reaccess_requirement(device Identity list)
4. Choose a device from the list and trigger the trust reaccess procedure
5. RA_request (list of RA items)
6. Prepare attestation report
7. RA_Response(RA Report)
8. Choose the trust level accessor (can be based on the information such as device type, manufacturer ... in the RA report)
9. Trust_Assessment (device identity, Context, RA Report, trust evaluation method)
10. Extract hardware, system and software configuration information from RA report and evaluate Reputation, Threat and risk level
11. Calculate the trust value or trust level based on the Context, Reputation, Threat and risk level
12. Trust_Assessment_results (device identity, evaluation results, expire time)
13. Trust assessment results is acceptable
14. Trust level_evaluation_results (identity, trust level)
15. Reconfigure the device capability based on the trust level

Figure 8: Stay Steps

We know that in the context of maintaining the integrity and security of a smart home network, it is essential to continually reassess the trustworthiness of connected devices. In the Sequence Diagram for Stay, this ongoing process is triggered when the Trust Level Accessor receives updated risk and threat information from an external entity. This information might include new vulnerabilities or emerging threats that could impact the devices within the smart home ecosystem.

Upon receiving this updated information, the Trust Level Accessor identifies the devices that might be affected and pinpoints their respective home controllers. This identification process is crucial for ensuring that only the relevant devices undergo the reassessment procedure, optimizing resource utilization and minimizing disruptions.

The Smart Home Controller then initiates the trust reassessment process by sending a trust reassessment requirement, which includes a list of device identities, to the devices in question. This requirement compels each listed device to undergo a reevaluation of its trustworthiness, ensuring that it continues to meet the security standards of the smart home network.

The Smart Home Controller selects a specific device from the list and triggers the trust reassessment procedure. This selection might be based on various criteria, such as the device's previous reputation, its criticality within the network, or its susceptibility to the newly identified threats.

The selected device responds by preparing an attestation report, which is a detailed document outlining its current state. This report includes information on the device's identity, hardware, system, and software configurations. The attestation report is then sent back to the Smart Home Controller, which subsequently forwards it to the Trust Level Accessor.

Upon receiving the attestation report, the Trust Level Accessor determines the appropriate trust level accessor based on the information contained within the report. This could involve considering factors such as the device type, manufacturer, and the specific nature of the risks involved.

The trust level accessor then performs a comprehensive trust assessment. This involves extracting detailed information from the attestation report and evaluating the device's reputation, threat level, and overall risk. The accessor calculates the device's trust value or trust level by integrating these evaluations with the contextual information provided by the Smart Home Controller.

The results of this trust assessment, including the device identity, evaluation outcomes, and an expiry time for the assessment, are then communicated back to the Smart Home Controller. The controller evaluates these results to determine whether the device continues to meet the necessary trust standards.

If the trust assessment results are deemed acceptable, the controller sends the trust level evaluation results back to the device. These results detail the device's identity, its current trust level, and the functionalities it is permitted to support within the network. The device then reconfigures its capabilities based on the received trust level, ensuring that it operates within the defined parameters and adheres to the network's updated security protocols.

This iterative process of trust reassessment and reconfiguration ensures that the smart home network remains secure and resilient, continually adapting to emerging threats and maintaining the integrity of all connected devices.

5.1.3. Leave Example

In the lifecycle of devices within a smart home network, there comes a time when certain devices can be removed due to their inability to meet trust requirements. In order to model this part, we have presented the Leave state and we can see its sequence diagram in Figure 9. Here, such process starts when the Trust Level Accessor updates the Trust Assessment results, which include the device identity, current state, and evaluation results. This information is crucial for making informed decisions about the device's status within the network.

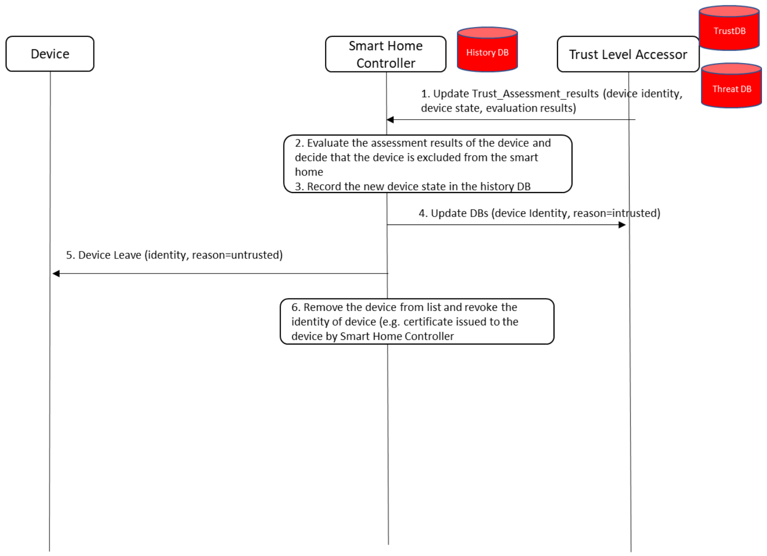

Device
Smart Home Controller
History DB
Trust Level Accessor
TrustDB
Threat DB
1. Update Trust_Assessment_results (device identity, device state, evaluation results)
2. Evaluate the assessment results of the device and decide that the device is excluded from the smart home
3. Record the new device state in the history DB
4. Update DBs (device Identity, reason=intrusted)
5. Device Leave (identity, reason=untrusted)
6. Remove the device from list and revoke the identity of device (e.g. certificate issued to the device by Smart Home Controller

Figure 9: Leave Steps

The Smart Home Controller receives these Trust Assessment results and proceeds to evaluate them. The evaluation involves a thorough analysis of the device's performance, security posture, and compliance with the network's trust policies. Based on this assessment, the Smart Home Controller may decide that the device no longer meets the necessary criteria to remain part of the smart home environment.

Once the decision is made to exclude the device from the network, the Smart Home Controller records this new state in the History Database. This step ensures that there is a complete and traceable record of the device's status and the reasons for its exclusion. This historical data can be useful for future audits, troubleshooting, and maintaining the integrity of the network.

Next, the Smart Home Controller updates the relevant databases, including the TrustDB and ThreatDB, with the device's identity and the reason for its untrusted status. This update is essential for maintaining an accurate and up-to-date view of all devices within the network, highlighting those that have been excluded due to trust issues.

The device then receives a notification of its exclusion from the network, specifying its identity and the reason for its untrusted status. This step formally informs the device that it is no longer part of the smart home environment, initiating its departure process.

Following this, the Smart Home Controller removes the device from the list of trusted devices and revokes its identity. This revocation typically involves invalidating any certificates or credentials that were issued to the device by the Smart Home Controller. By revoking these credentials, the controller ensures that the device can no longer communicate or interact with other devices within the network, effectively severing its ties to the smart home environment.

Through this structured and detailed process, the smart home network maintains its security and trustworthiness by ensuring that only compliant and secure devices are part of the ecosystem. This vigilance helps protect the network from potential threats and vulnerabilities posed by untrusted devices.

6. RTrustSim Description and Use Case Scenarios

RTrustSim is a trust management simulator designed for a social IoT environment, developed within the framework of Huawei's 1+8+N[1] paradigm. According to the model developed, we refer to the internal network as the 8 part and the external network as the N part. The framework is implemented in Java and C++, in accordance with the SOLID [2] principles to ensure a high degree of maintainability and scalability. The implementation of the simulator is not the objective of this paper, but rather the trust management within the RTrustSim simulator. Therefore, the code is not explained in the current paper.

The RTustSim simulator encompasses several modules for trust management, such as insertion of new devices, expulsion of devices, device management communications and interactions, statistics regarding device reputation within the cluster, cluster statistics, device history, event logging, device profile containing all device information justifying its treatment within the cluster, as well as a module designed to intentionally alter device behavior, among others. In Figure 10, the main modules of the simulator are displayed.

[1] https://medium.com/application-library-engineering-group/what-is-1-8-n-and-super-devices-976894ce0758

[2] https://www.digitalocean.com/community/conceptual-articles/s-o-l-i-d-the-first-five-principles-of-object-oriented-design

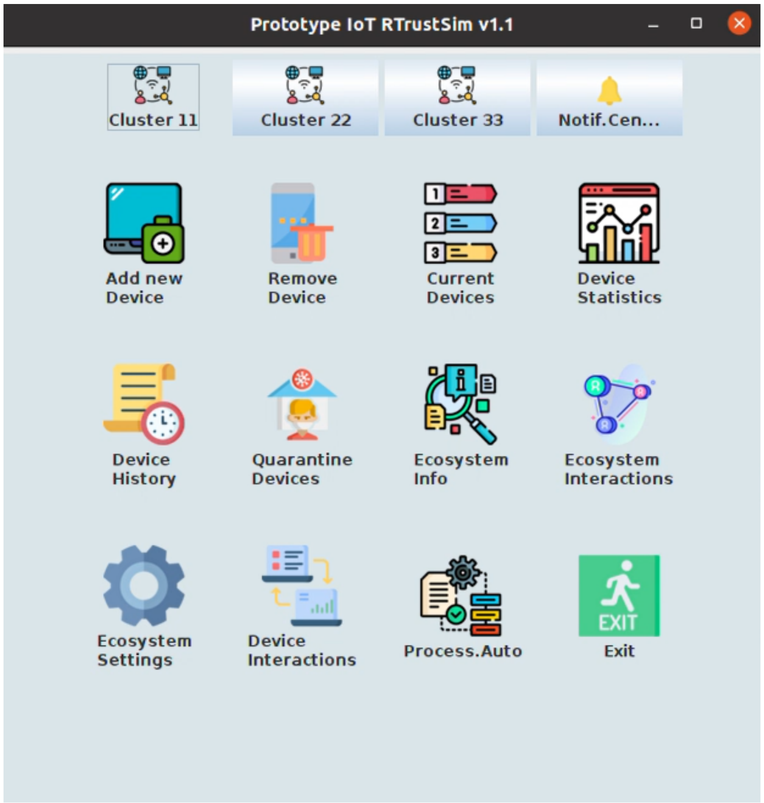


Figure 10: Main modules of RTrustSim

In this section, three different use cases will be considered. The first one represents a generic use case that leverages the device profile information to give a holistic insight into the manner in which the device is treated within the cluster. The cluster in this article refers to a set of devices with different levels of interactions, restricted by reputation and internal cluster policy. These devices are interconnected through one central device, typically a smartphone, responsible for managing interactions and the interoperability between devices within and between clusters. The decision of using the smartphone as central device is grounded in different reasons such as its wide connectivity, including Wi-Fi, Bluetooth, and mobile communication technologies, facilitating communication with various IoT devices. Additionally, the user-friendly interface, processing and local storage capabilities, and specific mobile applications make the smartphone the most appropriate device to serve as the central point of control and monitoring for multiple devices within the cluster. The second and third use cases focus on inter-cluster and intra-cluster interactions, respectively.

Although it is in its early stages of use, it is important to highlight that communication between devices is based on data from the NS3[3]) network simulator, which we have integrated into RTrustSim to obtain communication data as accurately as possible. Currently, the trust framework relies on NS3 as a tool for obtaining communication data in pcap format files, and the interpretation of such data is carried out manually using Wireshark.

6.1. Use Case 1: Generic use case

The main actors in this use case include the cluster owner, the central cluster device, which is in this case the Smartphone, and the devices participating in communication, such as the smart speaker with the identifier 3311 and the smart fridge with identifier 7766. As mentioned in the introduction of this section, the device profile generated internally in the cluster after the interactions between the cluster and the device is exploited. The generic scenario is split into two key points: parameters that define the level of trust of the device, and the interactions that determine the operability of the device within the cluster. The inclusion of a new device in the cluster is intrinsically linked to the initiative of the cluster owner. While the effective acceptance and operability of the latter is governed by the rules imposed by the cluster itself, although the cluster owner's influence on decisions is not disregarded. The cluster owner can adjust key parameters, such as the Minimum Joining and Remaining Threshold in the Cluster (MJRTC), directly affecting the device operations within the cluster.

During insertion of the new device into the cluster, its reputation undergoes an evaluation. Once a device becomes part of one of the clusters networks either network 8 or network N, it undergoes continuous behaviour evaluation. This process involves rapid detection and response to any alteration in the device behaviour. A change in the device behaviour results in an immediate change in its reputation value, subsequently altering the device's state.

The analysis of a device's profile stands out as a featured option to start the description of the general use case, as it encompasses all metrics that describe how the device will be managed within the cluster. Taking the example of a smart speaker, in this scenario, the smart speaker, also known as a virtual assistant, plays the role of a master. The virtual assistant positions itself as the device responsible for initiating, controlling, and managing communication with the other involved

[3] https://www.nsnam.org

devices.

6.1.1. Device parameters

Each device is identified by a unique device identifier, in this case the smart speaker is identified as 3311. To maintain conciseness in the presentation, certain parameters such as risks, threats and context inherent to the device's profile are omitted.

Thus, taking the example of an updated reputation value of 0.733 on a scale from 0 to 1, breaking down into two aspects: the original, linked to the creation model, and another adapted to the current cluster model to which it belongs. The creation date, set on Fri Oct 06 12:29:18 CEST 2023, and the last reputation update, recorded on Fri Jan 12 09:36:17 CET 2024, enhance the temporal context of the device. The state of the smart speaker, labeled as "Stay," confirms the authorization of operability within the cluster. Devices are categorized into four different states: Join, Stay, Quarantine, and Leave. This last state represents the definitive abandonment of the device. The device's reputation determines its specific state within the cluster.

Below, the functionalities that the device exercises within the cluster are revealed, each of these functionalities comes with a number that represents its importance, and Criticality Level of Operation (CLOP). Among these functions, the verification of the Smart Fridge's content stands out with a CLOP of 5 on a scale of 1 to 10, the preparation of the shopping list with a CLOP of 6, and the temperature control of the Smart Fridge with a CLOP of 7. This means that changing the temperature of the fridge is a more critical task compared to displaying its contents.

The device's reputation does not only affect its state but also influences the operations it performs within the cluster. A positive reputation implies fewer restrictions on the device's operations, keeping critical functions beyond the reach of devices with a lower reputation. Deterioration in reputation could result in limitations on the device's operations within the cluster. These limitations, in turn, could lead to a state of inactivity or complete paralysis of the device, known as quarantine.

6.1.2. Device interactions

In this section, we examine the potential interactions present in the device profile of the smart speaker with other devices within the cluster. This analysis serves as a basis to illustrate a communication scenario between two devices. In this case, our selection includes the smart speaker and the smart fridge, taking into consideration the established premises. It is noteworthy that the smartphone and the smart fridge have been successfully added to the cluster, achieving reputation values of 0.733 and 0.668, respectively. The minimum reputation considered in this communication is 0.668.

Throughout the communication between the smart speaker and the smart fridge, various essential operations and functionalities are conducted. These operations range from checking the refrigerator's content, with a CLOP of 5, to preparing the shopping list with CLOP of 6. These criticality levels represent the relative importance of each functionality and operations in the system's context, where higher levels emphasize functionalities considered critical for the overall performance and utility of the system form the cluster owner perspective. This scenario starts with an order issued by the cluster owner for the 'preparation of the shopping list' with a criticality level of 6, selected within a range of 0 to 10. The cluster verifies whether the minimum reputation between the smart speaker and the smart fridge allows the execution of this operation, in affirmative case, the cluster authorizes the operation between both devices.

During communication, a change in threat data associated with the Smart Speaker device (ID: 3311) is detected. After a reassessment of its reputation, it is determined that it has experienced a malicious behaviour, reaching a reputation of 0.45, below the MJRTC set at 0.5. The small difference of 0.05 indicates a mild behavioural change. As a result, the device is moved to quarantine, where it remains inactive and under behaviour supervision. If a positive change in behaviour is observed, the device can return to an active state inside the cluster. However, before coming back to the Stay state, the owner can perform antivirus scans. Another possibility, is the change of state after the firmware or software have been patched or updated. Nevertheless, if the difference between the MJRTC and reputation is substantial, the device is at risk of permanent exclusion from the cluster. Every device that interacts with the cluster will have a permanent reputation history in the cluster, facilitating thus evaluation in future attempts to join. Interactions extend beyond the confines of a single cluster, involving communication between devices located in diverse clusters. The adoption of the trust model occurs at the device level, leading to the presence of diversity in the implemented trust models, both within and between clusters.

Managing diversity in trust models emerges as an essential component to guarantee effective communication. In this context, the intermediate stage acts as a management mechanism, facilitating interoperability between different models and thereby contributing to the cohesion and efficiency of the system.

In Table 1, an intermediate stage is considered, which plays the role of a trans-

Decision Model 1	Intermediate Stage [0 1]	Decision Model 2
Low	[0 0]	Bad
Low	[0 0.25]	Not Bad
Medium	[0.33 0.5]	Good
High	[0.66 0.75]	Very Good

Table 1: Example of Decision Model Transformation Matrix

lator between two simplified decision models. The main difficulty lies in the lack of interoperability between both models, complicating the determination of which minimum reputation values between devices would allow the establishment of communication and determine which level of restrictions applies in the said communication. Additionally, the absence of direct comparability between values adds complexity to the process of establishing communication. The intricacy of the intermediate phase can fluctuate based on the trust models adopted by devices engaged in communication.

6.2. Use Case 2: Preventive health monitoring in Social IoT ecosystem

In the realm of preventive health monitoring, we consider the case of a patient with diabetes, Katty, and her father, Bob, who intends to regularly monitor and control Katty's health to detect possible anomalies that may require medical intervention. It is relevant to highlight that the primary goal is to prevent additional complications by anticipating any irregularities that may arise. Continuous monitoring seeks to prevent health problems and promote proactive healthcare related to Katty's diabetes. Additionally, it aims to provide doctors with a comprehensive understanding of her health.

Due to the sensitivity of the data exchanged between devices and considering the severity of its impact on patients' health, the trust placed in the devices involved is crucial. The trustworthiness on these devices is essential to ensure the effectiveness of preventive monitoring and secure information exchange, significantly contributing to proactive medical care and preventing potential complications associated with Katty's diabetes.

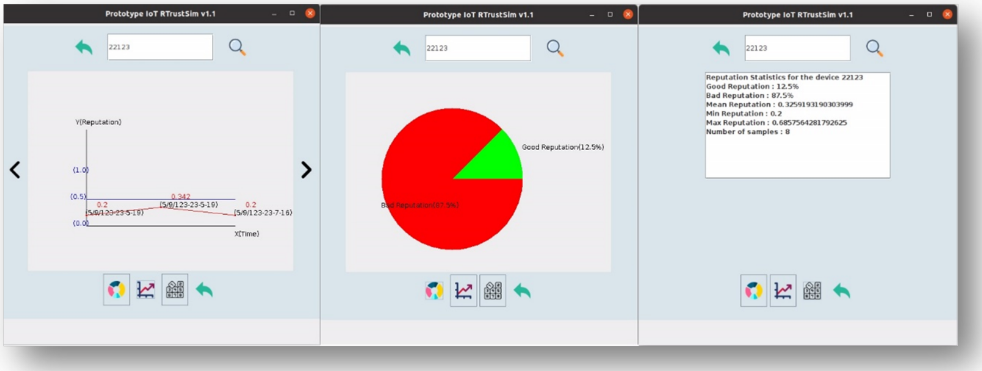
The scenario's circumstances are carried out within RTrustSim as follows. When Bob queries the Smart Speaker for a comprehensive report on his daughter's health, it triggers an attempt to establish communication between the Smart Speaker and the IoT devices within Katty's cluster. In the simulator context, this is simulated as an attempt to communicate between Cluster 1, which is Bob's Cluster and Cluster 2 owned by Katty.

The establishment of the communication triggers an internal evaluation of the device reputation within each cluster. The verification occurs as follows: first, the device's reputation must be equal to or higher than the MJRTC to achieve active state within the cluster. Subsequently, the second verification involves comparing the reputation with the CLOP, which, in this case, must be equal to or higher. Upon passing this verification with a reputation exceeding both MJRTC and CLOP, communication is established using Bob and Katty's smart mobile devices as intermediaries between the other devices in both clusters involved in the communication.

Given the diversity in the models adopted by the clusters, introducing an intermediate stage becomes imperative to foster understanding and interoperability between the clusters. In this phase, the aim here is to translate each of these models into an intermediate model that facilitates seamless interoperability with all the other models and makes possible the process of reverting from the unified model to the original one. This must be accomplished without any loss of accuracy or information. Since loss of accuracy could result in trusting malicious devices instead of quarantining them. RTrustSim offers a comprehensive report on all the interactions that occur inside the cluster.

The information is visualized through a window detailing the functionalities exchanged between devices during communication. Within the simulator, functions are provided to modify the metrics forming the reputation of devices. The goal of these options is to induce a change in device behaviour.

Assuming that during the communication between devices from both clusters, a change in the behaviour of one of Katty's devices is identified. Katty's and Bob's mobile devices are notified, and simultaneously, the reputation of the affected device is reevaluated. The new reputation turns in a transition from active state to inactive state also called quarantine, in which the device is placed under supervision, seeking a positive change in device behaviour. The objective of these measures is to ensure that only trusted devices have the right to operate and provide information. Furthermore, the criticality level of operations performed through these devices is linked to the level of trust placed on them. The dynamicity here is manifested in the proactive detection of device behaviour change and the measuring the impact of this change on the device reputation. The RTrustSim, as we can see in Figure 11, an overview based on device reputation history is provided throughout its instance in the cluster, encompassing statistical parameters that assist in accurately assessing the device performance decision-making.



```
Reputation Statistics for the device 22123
Good Reputation : 12.5%
Bad Reputation : 87.5%
Mean Reputation : 0.3259193190303999
Min Reputation : 0.2
Max Reputation : 0.6857564281792625
Number of samples : 8
```

Figure 11: Device Statistics

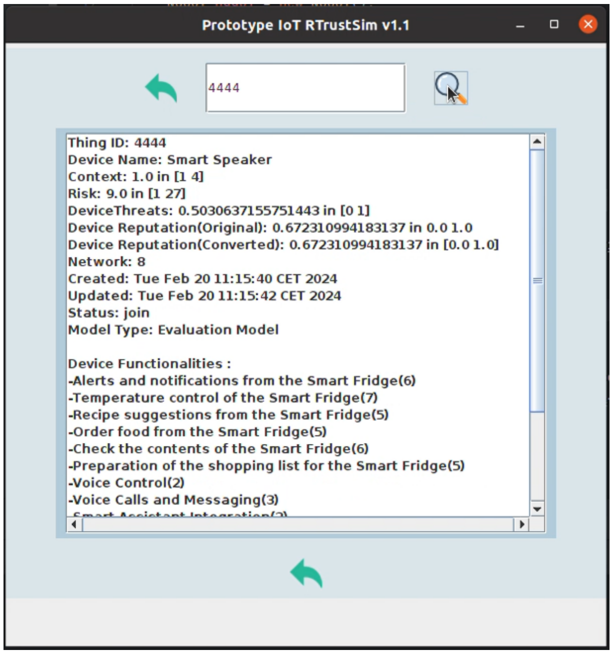
6.3. *Use Case 3: Dynamic Device Integration and Behaviour Evaluation in SIoT Clusters*

The word dynamics in the use case title refers to the variable criteria that the cluster uses to integrate a new device, based on the metrics that define its reputation and the thresholds established by the cluster to evaluate the behaviour of the device, while the SIoT is always referred to the relationship between devices within or between clusters.

This use case unfolds within the RTrustSim simulator, focusing on the integration of a new device into a cluster of devices within the SIoT environment.

It is essential to keep in mind that the decision to add a new device to the cluster always depends on the needs or request of the cluster owner, Bob. In the simulator, this process is carried out by inserting a device identifier and specifying its type. The simulator conducts an evaluation of the device reputation, based on factors such as risk, threats, context, and the device local reputation history. This comprehensive analysis contributes to informed decision-making regarding the inclusion of the new device in the cluster. Once the MJRTC is surpassed by the device reputation, the new device, in this case, a Smart Speaker with a reputation of 0.6 within a limited range of values between 0 and 1, is considered operational within the cluster. In this scenario, Bob decides to query the Smart Speaker about the expiration date of products in the fridge. In the RTrustSim simulator, this operation is simulated by establishing trusted communication between the smart fridge and the Smart Speaker. To execute this operation, the simulator performs a two-stage verification. Firstly, it examines the state of the devices involved in the communication, considering their respective reputations. Secondly, it verifies the execution permissions within the cluster by comparing the reputation with the CLOP, acting as a threshold that must be surpassed to enable the operation. In this case, both the Smart Speaker and the smart fridge are in the Join and Stay states, indicating that both devices are in an operational state. The minimum reputation in this case is that of the Smart Speaker, which is equal to 0.6. This minimum reputation exceeds the threshold set by CLOP. As the final decision, the cluster core orders the execution of the communication between the two devices. This communication involves the creation of a centralized star network on the smart mobile device, and Bob receives the response through the Smart Speaker's speaker. To provide dynamism to the scenario, the simulator allows a provoked change in one or several metrics that comprise device reputation. This change can manifest at the risk, threats, context, and/or reputation levels. The ability to simulate alterations in these metrics allows observing how they influence the response and performance of the device within the cluster, providing a deeper understanding of its behaviour in different situations and scenarios. Returning to the previous scenario of communication between the Smart Speaker and the smart fridge, a change in the risk associated with the Smart Speaker occurs during this exchange. This change in behaviour is identified, leading to a reassessment of the device reputation in question, which is in this case the Smart Speaker. The new assigned reputation is 0.5, which is equal to the MJRTC, indicating that the device is still considered operational. However, since the reputation of the Smart Speaker is below the CLOP required to perform operations of a medium level of criticality within the network, the device cannot execute the query operation regarding the expiration date of products in the fridge. This scenario causes a disruption in communication, persisting until proven otherwise by the device behaviour, and its reputation reaches or exceeds the threshold set by CLOP. This dynamic underscores the importance of maintaining reliability and fostering security standards within the cluster, ensuring consistent and efficient operation in the Social Internet of Things environment. Once the device becomes part of the cluster, a device profile is generated encompassing required parameter for delineating the subsequent treatment of the device within the cluster. The device profile is dynamic since any change in behavior generates an update of the device profile. Figure 12 illustrates an example of the device profile.

7. Conclusion and Future Work



4444

```
Thing ID: 4444
Device Name: Smart Speaker
Context: 1.0 in [1 4]
Risk: 9.0 in [1 27]
DeviceThreats: 0.5030637155751443 in [0 1]
Device Reputation(Original): 0.672310994183137 in 0.0 1.0
Device Reputation(Converted): 0.672310994183137 in [0.0 1.0]
Network: 8
Created: Tue Feb 20 11:15:40 CET 2024
Updated: Tue Feb 20 11:15:42 CET 2024
Status: join
Model Type: Evaluation Model

Device Functionalities :
-Alerts and notifications from the Smart Fridge(6)
-Temperature control of the Smart Fridge(7)
-Recipe suggestions from the Smart Fridge(5)
-Order food from the Smart Fridge(5)
-Check the contents of the Smart Fridge(6)
-Preparation of the shopping list for the Smart Fridge(5)
-Voice Control(2)
-Voice Calls and Messaging(3)
-Smart Assistant Integration(2)
```

Figure 12: Device Profile

This paper proposes a dynamic trust framework specifically designed for SIoT environments, addressing the critical need for secure and reliable interactions among different IoT devices belonging to different users. Our framework integrates a comprehensive trust calculation model that considers four essential elements: Threat, Reputation, Context, and Risk. This model does not only enhance the accuracy of the trust assessments but also adapts to the dynamic nature of IoT ecosystems. In fact, the interaction of devices within the same cluster or between different clusters implies the need of a maximum level of interoperability to ensure the correct understanding among the variety of trust modules that can be implemented within the cluster.

Through the evaluation of three distinct use cases, we have shown the practical applicability and effectiveness of our framework. The use cases indicate significant improvements over already existing elements such as device reputation assessment. In fact, our framework provides a more nuanced approach to evaluating device reputation by incorporating contextual factors and real-time threat assessments, leading to more informed trust decisions compared to traditional models that often rely solely on historical data. Moreover, we guarantee dynamic device integration with the ability to seamlessly integrate new devices into the SIoT ecosystem while maintaining trust integrity. This aspect represents a substantial advancement over existing frameworks, which frequently struggle with the challenges posed by heterogeneous devices and varying trust models. Finally, we facilitate secure interactions across different SIoT clusters as our framework addresses interoperability issues that have been inadequately tackled in prior research, thereby fostering a more cohesive and trustworthy SIoT environment.

In summary, our contributions do not only fill existing gaps in the literature concerning trust management in SIoT but also provides a robust foundation for future research and practical implementations. The dynamic trust model we propose is adaptable, scalable, and capable of evolving alongside the rapidly changing landscape of IoT technologies.

For future work, considering the fact that the integration of ns3 in this version is still in its early stages, the following versions of RtrustSim will be improved taking advantage of more metrics from ns3 in order to provide a better understanding of device behaviours. Additionally, we could explore the possibility of expanding the trust framework to operate in various sectors and environments, such as Industrial IoT (IIoT), where trust in devices is a significant concern. Furthermore, the trust framework can be used in Digital Twin Network (DTN) applications.

Acknowledgments

We thank Huawei Technologies for their support.

References

[1] Nitti, M., Girau, R., & Atzori, L. (2013). Trustworthiness management in the social internet of things. IEEE Transactions on knowledge and data engineering, 26(5), 1253-1266.

[2] Atzori, L., Iera, A., Morabito, G., & Nitti, M. (2012). The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. Computer networks, 56(16), 3594-3608.

[3] Ferraris, D., Fernandez-Gago, C., Daniel, J., & Lopez, J. (2019). A segregated architecture for a trust-based network of internet of things. In 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC) (pp. 1-6). IEEE.

[4] Abdelghani, W., Zayani, C. A., Amous, I., & Sèdes, F. (2016). Trust management in social internet of things: a survey. In Social Media: The Good, the Bad, and the Ugly: 15th IFIP WG 6.11 Conference on e-Business, e-Services, and e-Society, I3E 2016, Swansea, UK, September 13–15, 2016, Proceedings 15 (pp. 430-441). Springer International Publishing.

[5] Fernandez-Gago, C., Ferraris, D., Roman, R., & Lopez, J. (2024). Trust interoperability in the Internet of Things. Internet of Things, 26, 101226.

[6] Marche, C., & Nitti, M. (2020). Trust-related attacks and their detection: A trust management model for the social IoT. IEEE Transactions on Network and Service Management, 18(3), 3297-3308.

[7] Wang, Y., Chen, R., Cho, J. H., Swami, A., Lu, Y. C., Lu, C. T., & Tsai, J. J. (2016). CATrust: Context-aware trust management for service-oriented ad hoc networks. IEEE Transactions on Services Computing, 11(6), 908-921.

[8] Čolaković A., & Hadžialić M. (2018). Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues. Computer networks, 144, 17-39.

[9] Sharma, A., Pilli, E. S., Mazumdar, A. P., & Gera, P. (2020). Towards trustworthy Internet of Things: A survey on Trust Management applications and schemes. Computer Communications, 160, 475-493.

[10] Jøsang, A., Ismail, R., & Boyd, C. (2007). A survey of trust and reputation systems for online service provision. Decision support systems, 43(2), 618-644.

[11] Roman, R., Zhou, J., & Lopez, J. (2013). On the features and challenges of security and privacy in distributed internet of things. Computer networks, 57(10), 2266-2279.

[12] Karie, N. M., Sahri, N. M., Yang, W., Valli, C., & Kebande, V. R. (2021). A review of security standards and frameworks for IoT-based smart environments. IEEE Access, 9, 121975-121995.

[13] Schiller, E., Aidoo, A., Fuhrer, J., Stahl, J., Ziörjen, M., & Stiller, B. (2022). Landscape of IoT security. Computer Science Review, 44, 100467.

[14] Sereda, B., & Jaskolka, J. (2022). An evaluation of IoT security guidance documents: A shared responsibility perspective. Procedia Computer Science, 201, 281-288.

[15] Domínguez-Bolaño, T., Campos, O., Barral, V., Escudero, C. J., & García-Naya, J. A. (2022). An overview of IoT architectures, technologies, and existing open-source projects. Internet of Things, 20, 100626.

[16] Connectivity Standards Alliance (CSA), Matter, 2022, URL: https://csa-iot.org/all-solutions/matter/.

[17] Capuzzo, M., Delgado, C., Famaey, J., & Zanella, A. (2021, June). An ns-3 implementation of a battery-less node for energy-harvesting internet of things. In Proceedings of the 2021 Workshop on ns-3 (pp. 57-64).

[18] Fei, W., Ohno, H., & Sampalli, S. (2023). A Systematic Review of IoT Security: Research Potential, Challenges, and Future Directions. ACM Computing Surveys, 56(5), 1-40.

[19] Sagar, S., Mahmood, A., Wang, K., Sheng, Q. Z., Pabani, J. K., & Zhang, W. E. (2023). Trust–SIoT: towards trustworthy object classification in the social internet of things. IEEE Transactions on Network and Service Management.

[20] Alam, S., Zardari, S., Noor, S., Ahmed, S., & Mouratidis, H. (2022). Trust management in social internet of things (SIoT): a survey. IEEE Access, 10, 108924-108954.

[21] Becherer, M., Hussain, O. K., Zhang, Y., den Hartog, F., & Chang, E. (2024). On Trust Recommendations in the Social Internet of Things–A Survey. ACM Computing Surveys.

[22] Sagar, S., Mahmood, A., Sheng, Q. Z., Pabani, J. K., & Zhang, W. E. (2022). Understanding the trustworthiness management in the social internet of things: a survey. arXiv preprint arXiv:2202.03624.

[23] Chen, D., Chang, G., Sun, D., Li, J., Jia, J., & Wang, X. (2011). TRM-IoT: A trust management model based on fuzzy reputation for internet of things. Computer Science and Information Systems, 8(4), 1207-1228.

[24] Chen, R., Bao, F., & Guo, J. (2015). Trust-based service management for social internet of things systems. IEEE transactions on dependable and secure computing, 13(6), 684-696.

[25] Saied, Y. B., Olivereau, A., Zeghlache, D., & Laurent, M. (2013). Trust management system design for the Internet of Things: A context-aware and multi-service approach. Computers & Security, 39, 351-365.

[26] Ali, G., Ahmad, N., Cao, Y., Khan, S., Cruickshank, H., Qazi, E. A., & Ali, A. (2020). xDBAuth: Blockchain based cross domain authentication and authorization framework for Internet of Things. IEEE access, 8, 58800-58816.

[27] Mohammadi, V., Rahmani, A. M., Darwesh, A., & Sahafi, A. (2021). Trust-based Friend Selection Algorithm for navigability in social Internet of Things. Knowledge-Based Systems, 232, 107479.

[28] Chae, B. K. (2019). The evolution of the Internet of Things (IoT): A computational text analysis. Telecommunications Policy, 43(10), 101848.

[29] Ali, D. H. (2015). A social Internet of Things application architecture: applying semantic web technologies for achieving interoperability and automation between the cyber, physical and social worlds (Doctoral dissertation, Institut National des Télécommunications).

[30] Pang, J., Huang, Y., Xie, Z., Han, Q., & Cai, Z. (2020). Realizing the heterogeneity: A self-organized federated learning framework for IoT. IEEE Internet of Things Journal, 8(5), 3088-3098.

[31] Ferraris, D., & Fernandez-Gago, C. (2020). TrUStAPIS: a trust requirements elicitation method for IoT. International Journal of Information Security, 19(1), 111-127.

[32] Memarian, S., Farahani, B., & Nazemi, E. (2020). Social internet of things: interoperability and autonomous computing challenges. In 2020 International Conference on Omni-layer Intelligent Systems (COINS) (pp. 1-7). IEEE.

[33] Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. Computer networks, 76, 146-164.

[34] Sehgal, A., Perelman, V., Kuryla, S., & Schonwalder, J. (2012). Management of resource constrained devices in the internet of things. IEEE Communications Magazine, 50(12), 144-149.

[35] Bergman, J., & Johansson, I. (2017). The user experience perspective of Internet of Things development.

[36] Shafer, G. (1992). Dempster-shafer theory. Encyclopedia of artificial intelligence, 1, 330-331.

[37] Mell, P., Scarfone, K., & Romanosky, S. (2006). Common vulnerability scoring system. IEEE Security & Privacy, 4(6), 85-89.

[38] Freund, J., & Jones, J. (2014). Measuring and managing information risk: a FAIR approach. Butterworth-Heinemann.

[39] Ferraris, D., Bastos, D., Fernandez-Gago, C., & El-Moussa, F. (2021). A trust model for popular smart home devices. International Journal of Information Security, 20, 571-587.