A two-tier blockchain framework to increase protection and autonomy of smart objects in the IoT

A two-tier Blockchain framework to increase protection and autonomy of smart objects in the IoT

Enrico Corradini[1], Serena Nicolazzo[3], Antonino Nocera[2], Domenico Ursino[1], and Luca Virgili[1]

[1] Department of Information Engineering, Polytechnic University of Marche
[2] Department of Electrical, Computer and Biomedical Engineering, University of Pavia
[3] Daisy Lab, Polytechnic University of Marche
e.corradini@pm.univpm.it; serena.nicolazzo.sn@gmail.com; antonino.nocera@unipv.it;
d.ursino@univpm.it; l.virgili@pm.univpm.it

Abstract

In recent years, the Internet of Things paradigm has become pervasive in everyday life attracting the interest of the research community. Two of the most important challenges to be addressed concern the protection of smart objects and the need to guarantee them a great autonomy. For this purpose, the definition of trust and reputation mechanisms appears crucial. At the same time, several researchers have started to adopt a common distributed ledger, such as a Blockchain, for building advanced solutions in the IoT. However, due to the high dimensionality of this problem, enabling a trust and reputation mechanism by leveraging a Blockchain-based technology could give rise to several performance issues in the IoT. In this paper, we propose a two-tier Blockchain framework to increase the security and autonomy of smart objects in the IoT by implementing a trust-based protection mechanism. In this framework, smart objects are suitably grouped into communities. To reduce the complexity of the solution, the first-tier Blockchain is local and is used only to record probing transactions performed to evaluate the trust of an object in another one of the same community or of a different community. Periodically, after a time window, these transactions are aggregated and the obtained values are stored in the second-tier Blockchain. Specifically, stored values are the reputation of each object inside its community and the trust of each community in the other ones of the framework. In this paper, we describe in detail our framework, its behavior, the security model associated with it and the tests carried out to evaluate its correctness and performance.

Keywords: Internet of Things; Blockchain; Protection; Autonomy; Reliability; Trust; Reputation

1 Introduction

In recent years, the Internet of Things (IoT) paradigm has reached maturity and today is becoming increasingly pervasive in everyday life. This process has been made possible by new research approaches that enable objects

to be smart, autonomous and reliable. However, as the IoT grows, new challenges arise. In fact, the IoT is characterized by a large number of (often smart) objects with various constraints/features, such as: *(i)* limited storage and computing capability; *(ii)* great dynamism, due to the high number of nodes that join or leave the IoT at any time; *(iii)* criticality and sensitiveness of used services and applications. In this scenario, the protection of objects, on the one hand, and the possibility/need to guarantee them a great autonomy, on the other hand, represent two crucial issues to be addressed. As for *protection*, in [64] we presented a first approach to address this problem when it comes to privacy. In that case, we proposed to partially hide object features, but allowing their full use to support communication between objects. Nevertheless, the problem of providing a scalable, reliable and protected framework for IoT devices remains open. As for *autonomy*, making objects independent from each other during their interactions requires the capability of adding/removing contacts recognizing what features/services are provided by other objects [73, 84]. At the same time, in this context, the possibility of assessing the ability of an object to *concretely* and *correctly* provide the needed feature/service is fundamental. This is especially true when we consider the high vulnerability of smart objects to failures and/or cyber attacks, which could alter the way they behave. In the considered scenario, characterized by a high autonomy level of objects, leveraging compromised peers for services or for data retrieval may lead to the corruption of the whole IoT.

This reasoning highlights that autonomy and protection are two strongly interrelated aspects. In this scenario, the definition of trust and reputation mechanisms appears crucial [80, 19, 51, 20, 69, 28, 42]. However, most of the approaches proposed in recent literature describe strategies leveraging centralized services (such as whatchdogs) or particularly empowered smart objects, dedicated to data gathering from other objects and to the computation of trust and reputation values. Although these solutions may achieve pretty satisfactory results in some cases, they somehow force the fully distributed and autonomous nature of IoT to include "global" monitoring points.

To achieve a fully distributed solution in this setting, each smart object should be able to build a pretty complete representation of other objects' behavior in the IoT. However, as a prerequisite, it should also be able to unequivocally link a sequence of actions (defining a behavior) to each object. This would require the definition of an authentication mechanism to map each action (e.g., a transaction) to the object making it. One of the key aspects to be taken into account when addressing this issue is that the IoT is totally distributed. For this reason, classical Public Key Infrastructure models cannot be adopted because they refer to a common root of trust (CA root), which, for the reason stated above, is not easily achievable in this context. Indeed, the IoT should be totally distributed and composed of heterogeneous objects possibly belonging to independent domains. To address this issue, in the past literature, many authors have started to propose the use of the Blockchain technology in the IoT as a means to have a shared and reliable environment among all objects [24, 14, 31, 70, 57, 49, 75, 76, 79].

The application of Blockchain-based strategies to add trust and reputation facilities in the IoT without requiring any special actor (e.g., sophisticate smart objects) involved, poses a lot of interesting research challenges that must be faced to build a complete solution. One of the main problems is related to the high computational power required for deploying a Blockchain-based solution in the IoT context. Smart objects are intrinsically very heterogeneous and, therefore, provide a wide range of computation capacity spanning from fully equipped

powerful devices (such as smart cars, new generation smartphones, etc.) to very simple, with minimal computational capacity, smart sensors (e.g., smart meters, medical sensors, fitness trackers, etc.). In such a scenario, including the Blockchain technology can be very tricky because solutions must include the possibility of both exploiting fully equipped and powerful devices and supporting very simple and computationally limited ones. Moreover, if we observe this problem from the Blockchain perspective, handling the big volume of transactions generated by smart objects introduces important flaws in terms of both scalability and environmental costs [18, 77]. To partially face these issues several researchers focused on the definition of lightweight Blockchains for the IoT. Typically, these approaches work on the reduction of the information necessary to mine and validate transactions published in the ledger by proposing alternative consensus algorithms [30]. However, also the simple monitoring of the public ledger (to detect trust and reputation transactions, for instance) can be a heavy and expensive task for smart objects with minimal computation capacity in presence of a very high volume of transactions.

For this reason, some authors proposed to reduce the transaction volume to consider in the public ledger by adopting approaches based on the use of validity windows [78]. In this way, smart objects must only work with the transactions available inside the chosen window. Depending on the analyzed application scenario, reducing the size of transaction history may introduce important drawbacks; indeed, for instance, if such a ledger should be used to store trust and reputation information of smart objects at the end of a validity window, each object can have a fresh start as its reputation will be restored. To avoid this issue, historic data can be aggregated and made available inside each validity window; however, also this aggregation task can be very expensive and unfeasible for IoT objects if the volume of transactions is big [49].

This paper aims at providing a contribution in this setting. Indeed, it proposes a two-tier Blockchain framework to increase the protection and autonomy of smart objects in the IoT. Following the intuition proposed in [64], we consider smart objects as organized in communities. Hence, the first, local, tier is used to manage the trust measures of each smart object inside the community it belongs to and exploit a solution leveraging both a lightweight Blockchain and a validity window to control transaction volume. By organizing objects into communities, we can control the size of the local Blockchain in order to avoid excessive loads for smart objects. The second, global, tier is used to record aggregated data related to the individual communities, as well as the trust value that each community assigns to the other ones.

By definition, communities are built by looking at both the heterogeneity and the redundancy of provided features/services (so that multiple objects in the same community can offer the same feature/service). In a community, a smart object may require information to another smart object of the same community about the features/services offered by it. In order to estimate the latter's reliability, and ultimately its reputation in the community, our approach adopts a solution based on a probing mechanism. In particular, nodes are tested using probing queries about features/services they can provide. Their answers are then compared with those received by other nodes capable of offering the same features/services. This comparison allows the computation of the reliability of the tested object in providing the features/services declared. All transactions made to assess the reliability of smart objects in a community are stored in a Blockchain with a dedicated smart contract.

After a certain time window, our framework computes the reputation of each object inside its community. At the end of this process, smart objects that do not meet the minimum reputation level are removed from

the community. Then, for each community, a transaction with the list of its smart objects, along with their reputation, is stored in the Global Blockchain. In this way, the Local Blockchain is reset, following the approach described in [78], and all transactions occurring in the time window just passed are no longer considered.

Our approach also ensures protection when smart objects from different communities interact with each other. The procedure used in this case is similar to the one seen above. The results of a test performed by a smart object on another are stored in the Local Blockchain of the community the trustor object belongs to. Also in this scenario, after a certain time window, these transactions are aggregated and used to compute the trust of a community in another one. The trust values of each community in the other ones are stored in the Global Blockchain. Therefore, this last contains the reputation of each smart object in its community, as well as the trust of each community in the other ones it interacted with in the past. If there has never been an interaction between two communities, our approach assumes that each of them assigns a default trust value to the other one.

To perform the tasks described above, we use smart contract technology in the Blockchain. Indeed, Blockchain smart contracts are already being used to manage, control and secure IoT devices [53]. In particular, they can provide decentralized authentication rules and logic to implement single and multi-party authentication for an IoT device. They have been adopted to guarantee trustworthy and authorized identity registration, ownership tracking and monitoring of products, goods, and assets [68]. Their applications in IoT are discussed in [24], where the authors describe how Blockchain smart contracts can facilitate and support autonomous workflow and service sharing among IoT devices.

The previous description highlights that our approach is based on the requirement of having multiple objects providing the same feature/service in a community. Actually, this characteristic is common in several application scenarios. For instance, in contexts related to smart cities, the different available services are typically controlled through subgroups of smart meter/sensors dedicated to the specific domain, which the service belongs to. Such subgroups provide an adequate redundancy degree in order to avoid service outages. As a further example, consider the case in which a network of smart meters for fire detection is deployed in a forest [89]. This type of network typically includes sensors to measure temperature, humidity, wind speed and have a sufficient redundancy degree to handle fault tolerance. Again, we can think of the context of smart grids. These are power grids enabling a two-way flow of electricity and data across different smart sensors using the digital communication technology. This flow allows the system to detect, react and pro-act to problems that may occur or to usage changes. Of course, the security level required by the monitoring systems associated with smart grids needs ad hoc configurations. The latter typically involve clusters of smart objects collaborating with each other in a high fault tolerance setting. Another possible use case with these characteristics can be the video surveillance of critical public spaces through a network of smart drones capable of capturing snapshots of the monitored areas. In this case, the drones move freely in the environment and their number is often sufficient to ensure the coverage of the monitored areas. In this context, it is not unlikely that they can provide more than one proof of the same portion of an area in a given time interval. Snapshots can be compared through a proper similarity function that also takes into account different angles and perspectives.

Our approach also considers indoor scenarios, such as a smart home context. In these cases, the lower level of criticality and the smaller size of physical spaces make the need to provide redundancies less obvious.

However, the next generation smart devices are typically equipped with several different sensors; think, for instance, of smartphones, smartwatches and smart televisions. Such devices may actually represent backups to standard domotic sensors (e.g., smart meters), again allowing for the construction of heterogeneous groups of smart objects with some redundancy level. Finally, we mention an edge interesting use case consisting of a community of objects belonging to the personal area network of a user. For example, consider the simple user gait monitoring service. This is typically done through personal devices such as fitbands. However, the same functionality can be also provided by a smartphone (in many cases, even multiple smartphones if a person has both a work phone and a personal one with her). Still, it can also be provided through all those devices equipped with an Inertial Measurement Unit (IMU). Moreover, some recent approaches (such as the one described in [26]) show that the WiFi signal reflected by the human body generates unique, albeit small, variations in the receiver's wireless channel metrics, due to the well-known multipath effect of wireless signals. Thus, in principle, also WiFi can be used to measure a person's pace.

All these example application cases show that the presence of multiple objects providing the same feature/service in a community is common in the modern IoT scenario.

Moreover, through a deep experimental campaign, carried out leveraging real-life smart object data and Ethereum transactions, we prove that our approach is feasible and allows for the detection of compromised nodes in a relative small amount of time strictly related to the chosen probing frequency. Of course, as shown in Section 6.2, the fraction of probing interactions among objects can be suitably tuned to avoid downgrading the overall performance of the system, on the one hand, and to guarantee a satisfactory security level for smart objects, on the other hand.

The outline of this paper is as follows. In Section 2, we examine related literature. In Section 3, we describe the reference IoT model. In Section 4, we illustrate the proposed framework. In Section 5, we describe our security model. In Section 6, we present the set of experiments performed to test our approach. Finally, in Section 7, we draw our conclusions and have a look at possible future developments of our research efforts.

2 Related Work

Considering that the IoT paradigm has spread in a massive way in these last years, minimizing human intervention for the installation and management of its devices has been one of the main research direction in this context. This leads to the necessity of finding smarter and smarter autonomous decision-making processes, so that devices are able to vary their configuration dynamically throughout their working duration, selecting the best protocol to use, the best routes and the best nodes to communicate with [6, 72, 48, 3].

In this new perspective, the typical security goals of confidentiality, integrity and availability introduce additional problems. Indeed, the classical countermeasures to face privacy and security threats have to be rethought taking into account the many restrictions and limitations, in terms of components and devices, computational and power resources, and even the heterogeneous and distributed nature of the IoT [60, 91, 90, 2, 53, 4, 65, 54, 59].

In this context, trust architecture design and reputation evaluation play a crucial role, enhancing object security and reliable data collection and management. However, as stated above, due to its peculiarities (i.e.,

large number of entities with limited computation ability, and the highly dynamic nature of the network), existing solutions for sensor or P2P networks strive to be directly applicable to the IoT [80, 19, 51, 17, 20, 69, 28, 42, 83]. In particular, some works leverage cryptographic primitives or authentication mechanisms, such as TinySec [51], Key Session Scheme [20], SPINS [69], INSENS [28], and SERP [42]. However, they are computational demanding. Moreover, they are not secure against internal malicious nodes having the valid cryptographic keys. On the other hand, some of the nodes may have hardware fault (i.e., radio/sensor), and using only cryptographic mechanisms does not guarantee that these nodes are excluded from the network. Hence, a behavior-based or experience-based trust management framework is more suitable for our case.

In [19], the authors propose a scheme in which, using cryptographic primitives, each node has a unique and trustworthy identity. In addition, a reliable component evaluates the performance of the nodes based on an old trust degree and the indirect information from a third node. In [22], the authors present a trust architecture, called IoTrust, with a cross-layer authorization protocol. The main drawbacks of these two approaches are that: *(i)* they are based on a reliable level that computes node reputation score; *(ii)* they defend only against modification attacks, replay attacks, and message dropping attacks. An agent-based trust model for a WSN is presented in [21]. It uses a watchdog scheme to observe the behavior of nodes and broadcast their trust ratings. The sensor nodes receive the trust rating and make decisions about cooperation with other nodes. In [82] a reputation-based scheme called DRBTS is proposed. It provides beacon nodes with a method to monitor each other and supply information so that sensor nodes can choose who to trust, based on a quorum voting approach. However, in order to trust a beacon's information, a sensor must get votes for its trustworthiness from at least half of the set of neighbors it has in common with the beacon node. Also in these two cases, there are particular nodes (agent or beacon nodes) in charge of computing trust. On the other hand, our approach is completely decentralized and characterized by heterogeneous nodes exchanging different types of data.

We rely on Blockchain technology to reach a completely decentralized approach. In recent years, a lot of research has been carried out to deploy a Blockchain network in an IoT scenario in different environments, from industry to smart city and smart home. There could be a lot of advantages with this integration, such as managing device configuration, storing sensor data, enabling micro-payments and, above all, enhancing and securing IoT functionalities [24, 14, 31, 70, 57, 49, 75, 76, 79]. Among the above cited approaches, [70, 57] leverage Blockchain technology to provide forms of trust in an IoT network. In particular, the authors of [70] propose an approach for bridging trust between secure domains by leveraging Blockchain technology. They use an Obligation Chain containing obligations generated by Service Consumers, which are first locally accepted by Service Providers and, then, shared to the rest of the network. As this kind of approach is based on Islands of Trust, where the trust is provided and regulated by a full local PKI and CA, it cannot be used in our scenario, where a trust measure has to be computed for every node of the network. The authors of [57] focus on Long Range Wide Area Network (LoRaWAN), presenting a Blockchain based approach useful to verify that the data of a transaction existed at a specific time in the network. This architecture is specific for a LoRaWAN domain. Moreover, it simply uses Blockchain in a straightforward way, without providing a solution to label each node with a trust score.

A focal point is that, although Blockchain technology provides decentralized security and privacy, it involves significant energy, delay, and computational overhead, not suitable for most resource-constrained IoT devices.

So a lightweight instantiation of a Blockchain, suited for the IoT, is needed. The authors of [31] propose a new solution designing a private immutable ledger, which acts similarly to Blockchain, but is managed centrally. This approach consists of three layers namely: cloud storage, overlay, and smart home. Each smart home is equipped with an always online, high resource device, responsible for handling all communications internal and external to the home. These resource-rich devices also preserve a private and secure Blockchain, used for controlling and auditing communications. Also approaches such as Bitcoin-NG [38], LightChain [58] and Multichain [50] try to optimize the scalability of Blockchain proposing a leadership selection and a cross-chain mechanism, respectively. In particular, the authors of [58] propose a lightweight Blockchain framework, which is resource-efficient and suitable for power-constrained Industrial Internet of Things (IIoT) scenarios.

Another interesting solution has been proposed in [27]. Here, the authors address two well-known issues of the Blockchain technology, i.e., the low transaction rates and the environmental impact. They developed a hierarchical architecture, where the initial framework is split in a hierarchy of Blockchains (namely, sub-chains), managed independently. This architecture can exploit heterogeneous Blockchains and different hierarchies. Indeed, the authors use a Geospatial division to substitute the Proof-of-Work (PoW) with the Proof-of-Location (PoL); this last is based on the ability to generate a location certificate providing a location proof to create the next block. However, this approach cannot cope with the huge amount of transactions typical of an IoT context. Indeed, it is likely that, after a not too long time, the Blockchains of a network will be easily filled with many transactions. Clearly, this leads to an enormous usage of memory and computational resources that are not present in most IoT environments.

In our approach we consider a two-tier framework: a point-to-point local tier and a community-oriented global tier. Using a Blockchain to attest trust in the first local layer could be very computational demanding if this is performed for all transactions of the whole IoT. Hence, we use a Local Blockchain for each community to build the local tier in which each transaction is saved. At the global tier, transactions are aggregated and saved periodically in a Global Blockchain. In this way, our approach overcomes the limits of the applicability of Blockchain technology to an IoT scenario.

3 The reference IoT Model

In this section, we illustrate the model adopted to represent and handle the entities characterizing our framework. In our model, the main actor is the smart object. It has associated a profile with: *(i)* an identifier; *(ii)* a set of features characterizing it; *(iii)* a set of services it offers; *(iv)* the information needed for the communication with other smart objects (such as the MAC address, the IP address, etc.). The smart objects of the IoT can be partitioned into communities according to some rules (see Section 4.1 for the rules we adopted in this paper). Each smart object belongs to exactly one community. Smart objects can communicate with each other. This communication relies on suitable transactions involving a source smart object and a target one. Transactions can be performed to require the features/services declared by the target smart object (we call them *ordinary* transactions) or to test what it declared in order to evaluate its reliability (we call them *probing* transactions). Furthermore, transactions can be classified into *intra-community*, if they involve smart objects of the same community, or *inter-community*, if they involve smart objects belonging to different communities.

Parameter	*Meaning*
o_{i_k}	The smart object o_i of the community C_k.
C_k	A generic community of our framework.
tr_{i_k}	A trustor object belonging to C_k.
te_{j_q}	A trustee object beloning to C_q.
req_{i_j}	A probing transaction from tr_{i_k} to te_{j_q}.
P_{i_j}	A portion of smart objects able to answer req_{i_j}.
$\widehat{P_{i_j}}$	The "pruned" P_{i_j}.
out_j	The output to req_{i_j} provided by te_{j_q}.
$\overline{out_{i_j}}$	The average output to req_{i_j} provided by the smart objects of $\widehat{P_{i_j}}$.
T_{i_j}	The trust of tr_{i_k} in te_{j_q} after a probing transaction.
$\mathcal{F}$	A similarity function between out_j and $\overline{out_{i_j}}$.
τ	The tolerance admitted by the similarity function.
TrS_j	The set of trustors for te_{j_q}.
$\widehat{TrS_j}$	The "pruned" TrS_j.
R_j^ω	The reputation of te_j in C_q after the time window ω.
α	The weight of the importance of past data in R_j^ω.
$\overline{T_j^\omega}$	The average trust in te_j after the time window ω.
$\overline{t_q}$	The smart object in C_q supporting tr_{i_k} in its probing task.
$\overline{t_k}$	The smart object in C_k supporting te_{j_q} in its answer to tr_{i_k}.
$T_{k_q}^\omega$	The trust of C_k in C_q after the time window ω.
β	The weight of the importance of past transactions in the computation of $T_{k_q}^\omega$.
p	The probing probability.
$\Lambda_{k_q}^\omega$	A function evaluating the role of past transactions in the computation of $T_{k_q}^\omega$.
$\overline{T_{k_q}^\omega}$	The average trust values of the smart objects of C_k in the smart objects of C_q.
$\mathcal{I}_q^\omega$	A parameter denoting how much C_q has changed in the time window ω.
δ	A damping factor denoting the initial trust of a community.
$\mathcal{R}_{i_j}^\omega$	The reliability assigned by tr_{i_k} to te_{j_q} before starting a new transaction.

Table 1: The main abbreviations used throughout this paper

Each community has associated a Local Blockchain; it registers information about the transactions having a smart object of that community as trustor. The overall IoT has associated a Global Blockchain; it registers aggregated information produced periodically starting from the probing transactions registered in the Local Blockchains. As we will see in the following, the information stored in the Global Blockchain regards: *(i)* the list of smart objects belonging to each community and their reputation scores inside their communities; *(ii)* the trust of each community in the other ones of the IoT. In order to improve the readability of this paper, in Table 1 we report the main symbols used in it.

4 Technical description of our approach

In this section, we present the core of our approach. In particular, we describe our strategy to build the local and global Blockchain tiers to support the definition of a trust and reputation solution for smart objects. This section is organized as follows: In Subsection 4.1, we provide the general overview of the proposed scheme. In

Subsection 4.2, we discuss the computation of reliability measures for smart objects inside a community. In Subsection 4.3, we extend this activity to smart objects belonging to different communities.

4.1 General overview of the proposed scheme

As said in the Introduction, our goal is designing a framework to allow the protection of smart objects in an IoT scenario and, at the same time, the promotion of their autonomy. The autonomous interaction between smart objects occurs through mechanisms allowing each of them to understand what features/services can be provided by the smart objects it is in contact with [10]. The increasing of autonomy poses important challenges in terms of smart objects reliability. To address these challenges, we introduce in our framework suitable trust and reputation measurement techniques, which allow smart objects to assess the reliability of the smart objects they are in contact with, in order to "consciously" filter the information received. Following the standard approach accepted in the literature [1], our solution leverages two main tools to assess the reliability of smart objects. The first consists in the capability of verifying the ability of smart objects to provide the features/services they have declared. The second, instead, consists in the possibility of considering objects as belonging to a society in which information about measured objects' reliability can be propagated and is, hence, made available to all members. To enable the possibility for smart objects to assess whether their peers are reliable in providing the features/services they advertise, as done in [16], we adopt an approach based on probing transactions. As thoroughly explained in the Introduction, to support the evaluation mechanisms mentioned above, in particular to certify probing transactions, our approach uses a Blockchain-based solution. Regarding this choice, we highlight that, due to the distributed and decentralized nature of the Blockchain paradigm, approaches combining Blockchains and IoT are increasingly attracting interest in both the research and industrial context [66, 33, 74, 36, 81]. Many promising IoT applications that use a Blockchain-based layer to improve the autonomy and security of the involved smart objects have already been proposed. To give some examples, we can mention approaches in the contexts of device configuration management, sensor data storing or micro-payments [32, 85, 56]. However, it is also well known that there are many problems regarding the use of the Blockchain technology in the IoT [24, 14, 31, 70, 57, 49, 75, 76, 79]. They are mainly related to the high number of nodes involved and the large amount of data generated, as well as the low computational power of many smart objects.

Our approach addresses these issues by leveraging a two-tier Blockchain. In particular, we assume that smart objects are grouped into suitable communities according to different criteria. Within these communities, smart objects can adopt control mechanisms aimed at identifying anomalous behaviors and making interactions as secure as possible. We point out that our approach is orthogonal to how communities are formed. To this end, we could use any approach, such as the one proposed in [64]. This requires that smart objects in a community should present a certain level of redundancy of the features/services offered. This property is also fundamental in our approach. Indeed, in order to enable mechanisms to evaluate the ability of a smart object to provide a given declared feature/service, it is necessary to have an alternative source as reference (see below for details). Therefore, the first Blockchain tier is internal to a single community and is intended as a local public ledger in which the probing transactions inside a community are stored. The second tier is global and concerns the

whole IoT scenario; this level reports only *aggregated* information about the different communities.

As explained in the Introduction, the local tier could be implemented using a fully IoT-based solution that uses lightweight strategies to provide a public shared ledger. IoT devices alone cannot keep up with the computational power and energy demands of traditional Blockchains. In fact, most of them are based on the Proof-of-Work paradigm, which is not suitable for the IoT context. Nevertheless, several approaches to build lightweight Blockchains for IoT have been proposed in the scientific literature [66, 33, 74, 36, 81]. Among others, a very promising and up-to-date project is IOTA[1]. This is one of the most popular Blockchain-based ecosystems (at the time of writing this paper, the cryptocurrency underlying this system is ranked 24^{th} in the market capitalization). Furthermore, it has an important developer base and also supports smart contracts, thanks to the QUBIC protocol [63].

IOTA is based on a micro-transaction infrastructure for the IoT context. It represents a more energy-efficient technology than classic Blockchains, because it increases the transaction speed and makes it possible to perform transactions without paying any fees. The foundation of IOTA is the adoption of an acyclic directed graph called Tangle [81, 71]. In it, there are no blocks and each new transaction references the previous two ones in order to gain network consensus. Even with these tricks, the data to store can grow rapidly because there are many interconnected devices. To address this issue, IOTA proposes two solutions depending on the overall environment. The first consists in the creation of special entities, called *Permanodes*, which keep all Tangle data. The second involves the *snapshotting* of data, i.e., storing only the balances of the local addresses and deleting everything else. In this way, it is possible to group together several transactions of the same address in a log, which requires less storage. Of course, the first solution is the most expensive one because it implies the creation of a new entity with more resources in terms of computational power and energy than common IoT devices. For this reason, in our case, the second solution seems more suitable, since it requires less storage and has many possible configurations (such as global and local *snapshotting* [81]).

Although we have described IOTA in more detail, we repeat that our approach is orthogonal to the specific solution adopted to have a lightweight Blockchain in the IoT.

Instead, the global tier can be implemented on any Blockchain network, e.g., Ethereum[2] or HyperLedger[3]. This tier is only used to store aggregated data involving multiple communities. The frequency of use of the global tier is very low, compared to the one of local tiers. Therefore, the cost to access it is negligible for the smart objects in our framework. Since there are no stringent requirements for the global tier, as there are for the local ones, in the following, due to space limitations, we will not discuss this topic in detail.

Figure 4.1 shows the general architecture of our approach. As can be seen from this figure, smart objects are organized in heterogeneous communities. There is no restriction on the interaction between smart objects of different communities. In fact, in our approach, smart objects can continue to interact as freely as in any other IoT, thus preserving one of the main features of this kind of network [43, 8, 9].

To control and limit the transaction volume to be analyzed, the normal communication bursts among smart objects are conceptually divided into time windows. Within each window, in addition to normal interactions,

[1] `www.iota.org`

[2] `www.ethereum.org`

[3] `www.hyperledger.org`

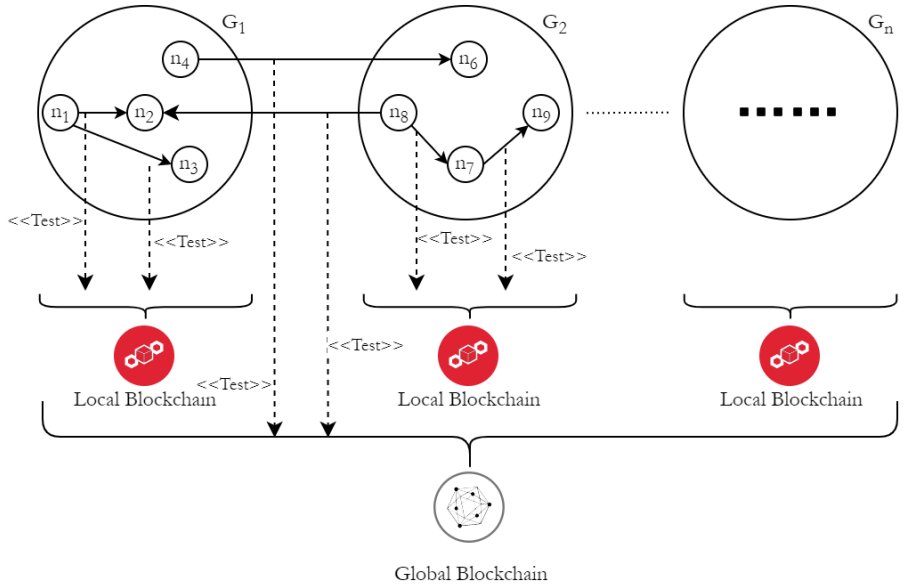


Figure 1: General architecture of our approach

probing transactions are performed. These tests are randomly generated; in particular, each smart object can decide to test another one belonging to its community with a certain probability. The test performed must be compliant with the features/services offered by the tested smart object.

In order to verify the reliability of the tested smart object, the tester requires the support of other smart objects belonging to its community and providing the same feature/service. With regard to this, based on the feature/service redundancy hypothesis characterizing the smart objects of each community, we assume that it is always possible to identify a subset of smart objects providing the same features/services of the tested one. They can be involved in the verification task.

Tests are used to compute the reputation of smart objects within their communities. All transactions associated with tests are recorded in the Local Blockchain of the community in order to make them provable and, therefore, reliable. After a defined time window, the reputation of each smart object in its community is computed by aggregating the results of the tests it has undergone. This also allows a limitation of the growth of the number of transactions in the Local Blockchain. The computation of the reputation values can be performed directly on the Blockchain using a dedicated smart contract. After this task, the list of community members, together with the corresponding reputation scores, is published in the Global Blockchain. Smart objects having a reputation score below a certain threshold are automatically removed from the community.

Our approach also guarantees protection in case of interaction between members of different communities. In order to deal with possible inter-community attacks, it provides mechanisms to test the reliability of smart objects even in case of communications between members of different communities. In particular, when two smart objects belonging to different communities contact each other, one of them can undergo the other one to a test with a certain probability. In order to compute the test result, the tester object requires to the tested one the features/services it declares. Furthermore, it performs the same request to other objects belonging to

the same community as the tested object. The test result is saved in the Local Blockchain of the community which the tester object belongs to. Analogously to what happens for local tests, after a defined time window, the transactions associated with the tests of smart objects belonging to external communities are aggregated. In this way, we get a trust value of a community in each external community with which at least one of its objects interacted. Also the trust values between communities are saved in the Global Blockchain.

Therefore, the Global Blockchain stores the reputation of each smart object in its community, as well as the trust of each community in the other ones it interacted with in the past. If there has been no interaction between two communities, our approach assumes that each of them has a default trust value in the other one, equal to the minimum trust value allowed.

Thanks to all information stored in the Global Blockchain, when a smart object o_{i_k} of a community C_k wants to interact with a smart object o_{j_q} of a community C_q, $C_q \neq C_k$, o_{i_k} can compute the reliability of o_{j_q} taking into account the reputation of o_{j_q} within C_q and the trust of C_k in C_q.

4.2 The Local Blockchain tier: assessing trust and reputation inside communities

In this section, we illustrate the tasks carried out by our approach to evaluate trust and reputation inside communities. In particular, in Subsection 4.2.1, we present the computation of the trust between two smart objects belonging to the same community; instead, in Subsection 4.2.2, we describe the computation of the reputation of a smart object inside its community.

4.2.1 Measuring trust in point-to-point interactions

In an IoT scenario, some malicious owners may exist. They could use their misbehaving devices for self-interests, for instance to perform some attacks to ruin the reputation of other IoT devices. For this reason, trust and reputation management is a key issue in IoT, and many researches on this topic can be found in the past literature [86, 13, 11, 12, 22, 19, 21, 80]. Typically, in the evaluation of trust and reputation, the following factors are considered [1]:

- *The quality of service provided by the device.* Also known as QoST (Quality of Service Trust), it is the ability of an IoT device to provide a service with a certain level of quality. QoST generally refers to performance and may depend on several parameters, such as competence, cooperativeness, reliability, task completion capability, and so forth.
- *The trust derived from the relationship between two objects or between an object and its owner*, also known as *Social Trust*. It may depend on parameters like intimacy, honesty, privacy, centrality, connectivity, etc. It is prevalent in Social IoT systems, where IoT devices must be evaluated based not only on QoST but also on the behavior of their owners [7].

A challenge-response approach is generally used for QoST computation. The idea is to estimate the reliability of the response obtained in a challenge between a trustor and a trustee. Generally, the trust interaction between two smart objects can be represented by means of a triplet `<trustor, trustee, feature/service>`. The field `feature/service` denotes the subject of the evaluation and is closely related to the application context. As an

example, this can be a service offered (like the news of the day) or a simple measurement of a quantity that the trustee can return to the trustor.

The social trust, instead, refers to the social behavior of an object, and possibly its owner, in its interaction with each other object in its community or, more generally, in the IoT.

In our system, we adopt a mixed solution that considers both the ability of an object to answer a probing query and the information on the same feature/service that can be obtained from other IoT objects answering the same query.

As said before, in our IoT framework, smart objects are grouped into communities. These are built taking care to guarantee the heterogeneity of features/services provided by its objects, and the redundancy in the provisioning (i.e., more objects can offer the same feature/service in one community).

Each node in a community can activate a probing activity towards another node in the same community by requesting the provision of a feature/service that the latter has declared to provide.

So, given a feature/service, say req_{i_j}, requested by a trustor tr_i to a trustee te_j, it is possible to identify a partition P_{i_j} of smart objects in the community able to provide an answer to req_{i_j}. Then, a "pruning" is performed on P_{i_j} to select the smart objects that are most likely to return an output close to the one returned by te_j. We call $\widehat{P_{i_j}}$ the partition P_{i_j} after this pruning. For example, if the required feature/service regards temperature measurement, P_{i_j} contains all the smart objects in the community able to measure temperature. Since this parameter is related to the context where a smart object operates, $\widehat{P_{i_j}}$ contains only those objects of P_{i_j} having a context compatible with te_j [29].

To measure the trust T_{i_j} of tr_i in te_j, we consider the deviation between the output out_j returned by te_j and the average output $\overline{out_{i_j}}$ provided by the smart objects of $\widehat{P_{i_j}}$. In particular, T_{i_j} can be computed as:

$$T_{i_j} = 1 - \mathcal{F}(out_j, \overline{out_{i_j}}, \tau) \tag{1}$$

Here, $\mathcal{F}$ is a dissimilarity function that returns real values in the range [0,1]. The greater the dissimilarity between out_j and $\overline{out_{i_j}}$ and the higher the value returned by $\mathcal{F}$. Clearly, $\mathcal{F}$ depends on the parameter we are measuring and the range of values it can assume. It also takes into account the tolerance τ allowed by the parameter. Also τ can assume values included in the real range [0,1].

Observe that the definition of $\mathcal{F}$ is orthogonal to our approach and may depend on several factors related to the parameter to measure or the service required. In case of services, $\mathcal{F}$ can take into account parameters such as Quality of Service (QoS) or Quality of Experience (QoE) [35, 39]. In case of a numerical output, linked for instance to a measurement, a possible definition of $\mathcal{F}$ could be the following:

$$\mathcal{F}(out_j, \overline{out_{i_j}}, \tau) = \frac{|out_j - \overline{out_{i_j}}|}{max(out_j, \overline{out_{i_j}})} \cdot (1 - \tau)$$

From the implementation point of view, our approach is based on a permissioned Blockchain in which there are some smart contracts dedicated to the computation and propagation of trust and reputation values. In particular, our approach saves probing transactions in a Local Blockchain. For this purpose, for each transaction, it activates a dedicated smart contract that implements the steps reported in Algorithm 1. The same steps are shown graphically in Figure 2. The choice of using a permissioned Blockchain allows the definition of different

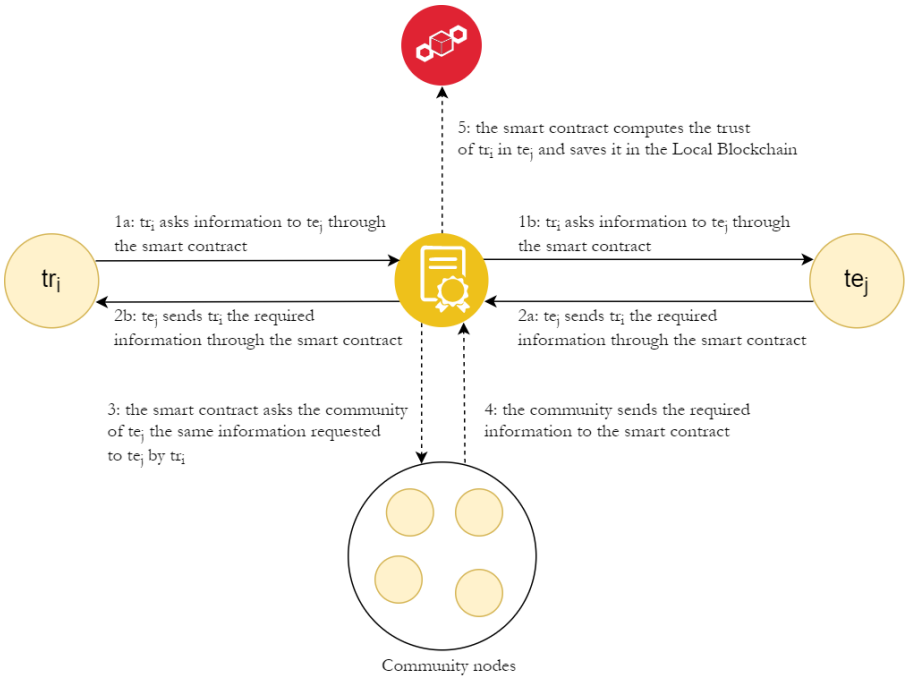


Figure 2: Computation of the trust of a trustor tr_i in a trustee te_j

policies for smart objects. In particular, these policies could be closely related to the criticality level of the communities. For example, in communities with a high level of criticality, joining can be made during the installation and maintenance tasks performed by system administrators. Instead, some communities, such as those related to smart home scenarios, might define less restrictive joining policies. In this case, smart objects could autonomously join a community.

Algorithm 1 Smart contract for the computation of the trust of a trustor tr_i in a trustee te_j

Require: The probability p_{act} that tr_i activates a probing transaction with te_j

generate a random value v_{act}

if $(v_{act} < p_{act})$ **then**

tr_i activates a probing transaction asking features/services to te_j

te_j provides the required output out_j

the smart objects of $\widehat{P_{i_j}}$ are required to provide the same output as te_i

the average value $\overline{out_{i_j}}$ is computed

the value of the trust T_{i_j} of tr_i in te_j is determined by applying Equation 1

the transaction and the corresponding trust is stored in the Local Blockchain

end if

4.2.2 Using a lightweight Blockchain for the computation of reputation values

Once many probing transactions are available in a community, it is possible to compute an aggregate measure, called reputation, for each smart object. It summarizes the opinion of the whole community towards the object [44]. In our case, the computation of the reputation also allows the implementation of a technique to keep the Blockchain size low.

Following an approach similar to the one presented in [78], we define a time window ω and consider all probing transactions made in this time window. At the end of ω, our framework aggregates the information about trusts and computes the reputation of the smart objects in their community as specified below.

Let TrS_j be the set of nodes that required at least one probing transaction to te_j. The reputation R_j^ω of te_j at the end of the time window ω is computed as:

$$R_j^\omega = \begin{cases} \alpha \cdot R_j^{\omega-1} + (1-\alpha)\overline{T}_j^\omega & \textit{if } TrS_j \neq \emptyset \\ \\ R_j^{\omega-1} & \textit{otherwise} \end{cases} \tag{2}$$

Here:

- $R_j^{\omega-1}$ is the reputation of te_j at the end of the previous time window;
- α is a parameter used to weigh the importance of past data with respect to the present ones. It plays an important role in the ability of our approach to react to anomalous situations. In fact, a high value of α would give a great importance to the historical behavior of a node, smoothing the effect of recent temporary variations in its interactions with other nodes. On the contrary, a low value of α would make our approach extremely reactive to any variation in the behavior of a node. A perfect balance between the history of a node and its recent interactions (which is achieved by setting $\alpha = 0.5$) might be a good choice for most application contexts. However, low values of α could be adopted in critical scenarios, where a high security level must be guaranteed. In this case, having a fast reaction of the reputation system is essential to exclude nodes that start to show a suspicious behavior.
- $\overline{T_j^\omega}$ is the average trust obtained by aggregating the values of trusts in te_j computed during ω. It can be obtained as:

$$\overline{T}_j^\omega = \frac{\sum_{tr_i \in TrS_j} T_{i_j}}{|TrS_j|} \tag{3}$$

The reputation values thus computed have a great influence on the evaluation of communities. In fact, the smart objects that do not meet the minimum reputation requirements are removed from the community. After these computations, the Local Blockchain is reset, following the approach described in [78], and all the transactions occurred during ω are no longer considered.

From a technical point of view, the computation of the reputation is carried out by a smart contract activated at the end of each time window. Given a community, the smart contract computes the reputation of each of its smart objects using Equation 2. The smart contract time scheduling can be done following existing technical

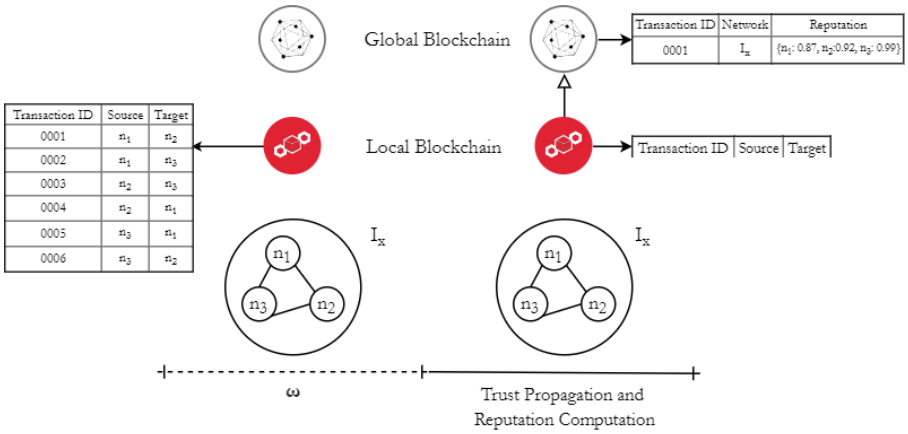


Figure 3: Transaction aggregation and computation of the reputation of the smart objects of a community

approaches, for example the Ethereum Alarm Clock[4]. Note that the length of the time window can be related to the number of transactions generated in the Blockchain, instead of a clock. In this case, when transactions exceed a certain threshold, the smart contract is activated.

Once all the reputation values have been computed, the smart contract updates the list of smart objects that can be still part of the community (i.e., the smart objects whose reputation score is higher than a minimum threshold). This list, together with the reputation score of the smart objects (also of the ones that will be removed), is registered in the Global Blockchain through a new transaction. This behavior is represented in a more coded way in Algorithm 2. Instead, a graphical representation is provided in Figure 3.

Algorithm 2 Transaction aggregation and computation of the reputation of the smart objects of a community

Require: the time window ω, the reputation threshold th_R and the weight α of the past reputations

```
wait for the end of the time window ω
for te_j in the community C do
    compute T̄_j^ω applying Equation 3
    compute R_j^ω applying Equation 2
    if R_j^ω < th_R then
        remove te_j from C
    end if
end for
register all the reputation values in the Global Blockchain
```

4.3 The Global Blockchain tier: towards reliable community level interactions

In this section, we discuss how to evaluate and activate reliable transactions between smart objects belonging to different communities. In particular, in Subsection 4.3.1, we illustrate how probing transactions between smart

[4] `https://www.ethereum-alarm-clock.com/`

objects belonging to different communities are managed. In Subsection 4.3.2, we describe the computation of the trust of a community in another one. Finally, in Subsection 4.3.3, we show how a smart object can evaluate the reliability of another smart object of a different community before starting a communication with it.

4.3.1 Enabling community level reliable interactions

So far we have seen the interactions between smart objects in the same community. However, as we said in Section 4.1, our framework also allows smart objects from different communities to interact with each other. This is done implementing a hierarchical approach, that is possible thanks to the presence of a two-tier Blockchain framework. Our approach is inspired by some of the ideas proposed in [27].

In our framework, the smart objects of each community are free to interact with any smart object of the framework, even those belonging to different communities. In the latter case, a smart object can rely on the information concerning the community of the smart object it wants to communicate with and the reputation of this last object in its community. This information is registered in the Global Blockchain.

However, different attack scenarios may lead to the fact that the only information about the reputation of a smart object, resulting from its interactions within its community, is not sufficient to guarantee its reliability. In fact, an attacked smart object could assume a polymorphic behavior, interacting positively with all the smart objects of its community but acting negatively with the ones belonging to other communities.

For this reason, our approach provides a mechanism to compute the trust also between objects belonging to different communities. To this end, we extend the probing mechanism described in Section 4.2.1. In particular, a smart object tr_{i_k}, belonging to a community C_k, can start the test of a smart object te_{j_q}, belonging to a community C_q, $k \neq q$, with a certain probability. For this purpose, tr_{i_k} makes a request req_{i_j} for a feature/service to te_{j_q}.

After that, tr_{i_k} randomly selects a node $\overline{t_q}$ and sends it the same request req_{i_j} previously sent to te_{j_q}. Clearly, req_{i_j} is built taking into account the features/services offered by te_{j_q} to make sure that this last object can provide an answer. The node te_{j_q} does not know that tr_{i_k} is making a probing transaction, while $\overline{t_q}$ is informed about it. The task of $\overline{t_q}$ is to select a partition TrS_q of the nodes of the community C_q that can answer req_{i_j}. Similar to what has been done in Section 4.2.1, among the nodes of TrS_q, our approach selects those ones being most likely to provide a correct answer. This leads to a "pruning" of TrS_q; we call $\widehat{TrS_q}$ the resulting set.

It is worth noting that, also in this case, the partition TrS_q of support nodes is selected from the trustee community itself. This choice strictly depends on the use cases considered in our approach and described in the Introduction. Indeed, although a larger set of nodes from different communities might be involved in the selection of the support partition, our approach makes assumptions only on the availability of redundant services within communities and not across them. This choice also reduces the complexity of service discovery strategies [84, 73], which can be applied to just a controlled-size community, instead of a whole world-scale IoT.

The smart object $\overline{t_q}$ sends req_{i_j} to the objects of $\widehat{TrS_q}$ and, when it receives the corresponding answers, computes the average output $\overline{out_{i_j}}$. At this point, $\overline{t_q}$ must send this value to tr_{i_k}. However, to avoid attacks from this last object, $\overline{t_q}$ does not send $\overline{out_{i_j}}$ directly to tr_{i_k}. Instead, it randomly selects an object $\overline{t_k}$ of C_k and sends $\overline{out_{i_j}}$ to it by specifying the final receiver. Indeed, if $\overline{t_q}$ would send the answer to tr_{i_k} directly, this last

could alter this answer and, therefore, force the assignment of a disadvantageous trust value to te_{j_q}. Instead, if $\overline{t_q}$ sends the answer to a random node $\overline{t_k}$ of the same community as tr_{i_k}, because all trust interactions are stored in the Local Blockchain, tr_{i_k} cannot change the answer provided by $\overline{t_q}$ through $\overline{t_k}$.

Finally, te_{j_q} returns its output out_j to tr_{i_k}. Again, in order to be protected from the attacks of tr_{i_k}, te_{j_q} can decide, with a certain probability, to forward out_j to tr_{i_k} through a randomly selected node $\overline{t'_k}$ of C_k, instead of sending out_j directly to tr_{i_k}. Again, in case of an indirect answer, te_{j_q} specifies the final receiver in the message sent to $\overline{t'_k}$. When tr_{i_k} receives out_j and $\overline{out_{i_j}}$, it can compute the value of the trust T_{i_j} by applying Equation 1. All communications made within the communities C_k and C_q continue to be saved in the corresponding Local Blockchains so that the test results can be verified by the other smart objects of the communities. Also these results are saved in the Local Blockchain of C_k and, therefore, can be partially reproduced starting from the transactions involving $\overline{t_k}$ and $\overline{t'_k}$ (if this last one has been involved by te_{j_q}).

4.3.2 Computing community trust

As seen in Section 4.1, when a time window ω has passed, probing transactions are used to compute the trust of the community C_k in the community C_q. For this purpose, we proceed as specified below. Let TrS_{k_q} be the set of the smart objects of C_q with which any smart object of C_k had a probing transaction during the time windows just passed. The trust $T^{\omega}_{k_q}$ of C_k in C_q at the end of ω can be computed as:

$$T^{\omega}_{k_q} = \begin{cases} \beta \cdot \Lambda^{\omega-1}_{k_q} + (1-\beta)\overline{T}^{\omega}_{k_q} & \textit{if } TrS_{k_q} \neq \emptyset \\ \\ \Lambda^{\omega-1}_{k_q} & \textit{otherwise} \end{cases} \tag{4}$$

Here:

- The parameter $\Lambda^{\omega-1}_{k_q}$ is defined as:

$$\Lambda^{\omega-1}_{k_q} = \mathcal{J}^{\omega}_{q} \cdot T^{\omega-1}_{k_q} + (1 - \mathcal{J}^{\omega}_{q}) \cdot \delta \tag{5}$$

  where:

  - δ is a damping factor that denotes the initial trust in a community C_q when no probing transaction has been requested to any of its smart objects.
  - $\mathcal{J}^{\omega}_{q}$ is an index that expresses how much C_q has changed (in the composition of its smart objects) since the previous time window $(\omega - 1)$. It can be obtained by computing the Jaccard Coefficient between the sets of the smart objects present in C_q in the time windows $\omega - 1$ and ω.
  - $T^{\omega-1}_{k_q}$ is the trust of C_k in C_q in the time window $\omega - 1$.

  The rationale behind this equation is related to the fact that the trust of C_k in C_q depends on the interactions that the corresponding nodes had during the time window ω and on the ones they had during the other past windows. If C_q has changed heavily (because its smart objects present during ω are very different from the ones present in the past) then the historical trust must be reset to the damping value δ.

- β is a parameter weighting the importance of historical data compared to those obtained in the last time window. The role of β is identical to the one assumed by α in Section 4.2.2. Specifically, it can be used to adjust the adaptation level of our security mechanism to new probing results. Again, the lower β, the higher the importance of recent trust interactions.
- $\overline{T_{k_q}^{\omega}}$ is the average of the trust values that the smart objects of C_k had in the smart objects of C_q with which they interacted during ω. It can be computed by means of the following formula:

$$\overline{T}_{k_q}^{\omega} = \frac{\sum_{te_{j_q} \in TrS_{k_q}} \overline{T_j}}{|TrS_{k_q}|} \tag{6}$$

  where $\overline{T_j}$ is the average trust assigned by the smart objects of C_k to the smart object te_{j_q}.

The values of $T_{k_q}^{\omega}$ obtained through Equation 4 are published in the Global Blockchain. Also in this case, at a technical level, the activities carried out to obtain the trust values described above are managed through a dedicated smart contract.

4.3.3 Assessing smart object reliability for community-level interactions

In the previous sections we have seen that it is possible to compute the trust of a smart object in another one of the same community. We have also seen that each smart object has a reputation within its community. Finally, we have seen that it is possible to compute the trust of a community in another one. These last two pieces of information are registered in the Global Blockchain and, as we will see, allow us to compute the reliability that a smart object of a community assigns to a smart object of a different community.

In particular, the reliability assigned by a smart object tr_{i_k} of a community C_k to a smart object te_{j_q} of a community C_q after the time window ω is computed as:

$$\mathcal{R}_{i_j}^{\omega} = \begin{cases} T_{k_q}^{\omega} \cdot R_j^{\omega} & \textit{if } T_{k_q}^{\omega} \textit{ is not null} \\ \\ \delta \cdot R_j^{\omega} & \textit{otherwise} \end{cases} \tag{7}$$

The rationale behind this equation is as follows: In case C_q has interacted with C_k in the past (which implies that $T_{k_q}^{\omega}$ is not null), the reliability $\mathcal{R}_{i_j}^{\omega}$ assigned by tr_{i_k} to te_{j_q} is obtained by multiplying the reputation that te_{j_q} has within C_q with the trust of C_k in C_q. Instead, if there has been no interaction between C_k and C_q in the past, then, in order to obtain $\mathcal{R}_{i_j}^{\omega}$, the reputation of te_{j_q} in C_q is "corrected" with a damping factor equal to the one used in Equation 5 and indicating the minimum trust that a community has in another one of the framework. Clearly, tr_{i_k} only interacts with te_{j_q} if $\mathcal{R}_{i_j}^{\omega}$ is high enough, i.e., greater than a minimum acceptable value.

5 Security Model

In this section, we illustrate the security model conceived for our framework. In particular, we present both the attack model and a security analysis showing that our framework addresses its objectives also in presence

of attacks. In the security analysis, we refer to classical attacks to reputation systems adapted to our approach [46, 47] .

5.1 Attack Model

As a preliminary assumption, we consider a realistic scenario in which a sufficient number of nodes is available so that our approach can be implemented successfully. Therefore, we do not consider anomalous situations or the startup time, in which the number of the nodes available in the framework is less than the minimum necessary.

In the analysis of security properties, we will consider that our threat model includes the following assumptions:

A.1 At most t smart objects can collude to break the security properties of the protocol.

A.2 The size of all the pruned support partitions, $|\widehat{P}|$ and $|\widehat{TrS}|$, is greater than t (see Sections 4.2.1 and 4.3.1).

A.3 An attacker cannot control a whole group of smart objects; moreover, she/he cannot own all the smart objects providing a certain service.

A.4 An attacker has no additional knowledge derived from any direct physical access to smart objects.

A.5 The Blockchain technologies exploited to implement both the Local and the Global tier are compliant with the standard security requirements already adopted for common Blockchain applications.

As for the first assumption, we recall that probing transactions are produced collaboratively by several smart objects in our protocol. Some of them might be corrupted but we assume the honesty of the majority of them, as done in [40, 25].

The list of the security properties (hereafter, SP) that our framework must assure is the following:

SP.1 Resistance to the Local and Global Blockchain tier Attacks.

SP.2 Resistance to Self-promoting Attacks.

SP.3 Resistance to Whitewashing or Self-serving Attacks.

SP.4 Resistance to Slandering or Bad-mouthing Attacks.

SP.5 Resistance to Opportunistic Service Attacks.

SP.6 Resistance to Ballot Stuffing Attacks.

SP.7 Resistance to Denial of Service (DoS) Attacks.

SP.8 Resistance to Orchestrated Attacks.

SP.9 Resistance to malicious probing exploitation.

5.2 Security Analysis

In this section, we focus on each of the security properties introduced above and analyze if and how our approach can guarantee it.

5.2.1 SP.1 - Resistance to the Local and Global Blockchain tier Attacks

This category of attacks aims at finding vulnerabilities in the Blockchain layers adopted in our framework. Of course, if one of the ledgers is compromised, our approach cannot work properly because probing transactions could be tampered or removed to modify the recorded behavior of each smart object involved. Even if, in the recent years, Blockchain has received a lot of attention from both the scientific community and the industry, the security of Blockchain is still subject of debate. A lot of approaches to face security flaws of the Blockchain in application scenarios related to ours have been proposed [57, 53]. Our approach is orthogonal with respect to these approaches. Indeed, we do not focus on improving Blockchain security, but we use it to implement a public ledger to store probing transactions among smart objects, as well as reputation values. Therefore, as stated by Assumption **A.5**, we consider the Blockchain as a secure layer in our approach.

5.2.2 SP.2 - Resistance to Self-promoting Attacks

This attack occurs when a smart object manipulates its own reputation to increase it falsely and promote itself. It can be carried out by an attacker operating alone or organized in groups of collaborating identities.

In our approach, a smart object tests another one through probing transactions. The trust score, obtained thanks to them, is stored in the Local Blockchain of the corresponding community. After this, a reputation score is computed starting from these trust scores. Hence, a smart object cannot alter its own score by itself. This ensures data authenticity and integrity and, therefore, our framework's capability of resisting to such an attack.

As for inter-community transactions, a smart object cannot assign a false score to itself because, also in this case, our framework allows the computation of smart object reliability only after a set of probing transactions, devoted to evaluate the reputation of the smart object in its community and the trust of the other communities in this last one. All the probing transactions are stored in Local Blockchains and, after a time window, they are aggregated in the Global Blockchain.

However, even if source data is authentic, a self-promotion attack would be still possible if a single attacker (or more attackers) manipulates nodes through a Sybil attack [34]. In this case, the sybil nodes would collude to promote each other. However, this cannot be possible due to Assumptions **A.1**, **A.2** and **A.3**, which imply that only $t < |\widehat{P_{i_j}}|$ smart objects (where $|\widehat{P_{i_j}}|$ is the number of smart objects in the pruned partition involved in a probing transaction - see Section 4.2.1) can collude.

Furthermore, at a local level, since smart objects are not aware if they are answering a probing or a standard query, a malicious behavior of them would cause a reduction of their reputation score. Instead, at the global level, support smart objects for testing are chosen randomly (see Section 4.3.1). This inhibits an attacker to rely on the possibility of controlling both the tested smart object and the support one.

In the remote possibility that, by chances, the support smart object is controlled by the attacker, this last could force a low trust score for a target smart object. However, this malicious attempt will not strongly affect the overall trustworthiness of the target smart object. Indeed, our metrics considers the whole history of interactions and, therefore, the impact of outlier values is strongly reduced.

5.2.3 SP.3 - Resistance to Whitewashing or Self-serving Attacks

This attack occurs when a malicious smart object with a compromised reputation, also called traitor [62], behaves in such a way as to quickly degrade its reputation with the goal of being removed from the framework. After this, it asks to rejoin the framework with a fresh reputation score in order to continue behaving maliciously. This kind of attack cannot be carried out in our framework because reputation scores are stored permanently in both the Local Blockchains and in the Global one. Due to this fact, our approach keeps memory of a malicious behavior even after the corresponding smart object has been removed from our framework.

Actually, it is possible to define a time interval, say ϕ_{ban}, during which the object can no longer be part of the framework. After this interval, the object can be restored and can join its community again with the initial minimum reputation value. Of course, the tuning of ϕ_{ban} is strictly related to the safety level of the considered scenario. The higher the safety level, the higher the ban interval. In the extreme case, for a very critical scenario (e.g., smart grids, nuclear firms, and so forth), ϕ_{ban} can even tend to infinity (which is equivalent to a permanent removal of the banned node from the framework). It should be noted that, in case of object outage, the ban interval can also be estimated based on the time required by a system administrator of the local community to intervene and restore it.

The previous solution implies that our approach must be able to maintain a clear association between objects and their corresponding reputations. This assumes that each object has an appropriate identifier. However, in a real-world scenario, in which objects join the network autonomously, an attacker could forge a new identifier for an object each time it is banned from the system. In this way, she/he could try to whitewash the reputation of the object, which would be identified as a new actor. Consequently, she/he could make multiple attacks avoiding the banning interval.

The past literature on this topic reports several studies aimed to define mechanisms allowing the management of strong identifiers for smart objects even in untrusted scenarios. For example, the authors of [45] propose a fully decentralized, self-maintaining and lightweight approach to handle consistent ID-to-dynamic IP mappings and use them in the routing process. Other approaches are based on object fingerprinting and focus on the problem of identifying general characteristics that may be present in any IoT device, whose values allow the extraction of patterns to unambiguously identify a single specific object. For example, to compute object fingerprints, the authors of [26] extract 19 features from 802.11 probe fields, while the authors of [67] focus on a set of features related to TCP timestamp and clock characteristics. Still in this context, the authors of [61] consider the relationships between objects and human actors in the IoT to model a new identifier format called GARI. Each of these approaches could be adopted by our model to ensure a robust mapping between trust and reputation values and the corresponding smart objects.

Actually, as for this issue, there is another research strand that proposes mitigation strategies for whitewashing attacks by leveraging a pessimistic attitude to the initial reputation values associated with newly added

actors [88, 41, 55]. In this case, a new object, or an object with a new forged identifier, is admitted to the network with a low default reputation value, less than the chosen threshold. Therefore, it is automatically put in a suspended state for a time equal to ϕ_{ban}. By adopting this strategy, an attacker is discouraged from performing whitewashing attacks by changing the object identifier. In fact, joining the system with a new identifier would coincide with the case where a node is temporarily banned. This solution seems the most appropriate for our scenario because it allows our approach to be resistant to this type of attacks without the need to integrate strategies for managing object identifiers.

5.2.4 SP.4 - Resistance to Slandering or Bad-mouthing Attacks

In this case, an attacker tries to manipulate the reputation of other smart objects by reporting false data. The attack can be carried out by a single smart object or a coalition of smart objects. Our model is resistant to this kind of attack because of its strict feedback mechanisms and the fact that the input validation is based on the Blockchain technology.

In particular, as for the intra-community case, smart objects are not aware if they are answering a test or a query. Hence, being malicious for an object could mean lowering its own reputation score and, after a while, being removed from the framework. Observe that, in this case, controlling a coalition of smart objects would not guarantee any benefit to the attacker.

As far as inter-community communications are concerned, several interesting situations should be analyzed. Specifically, assume that a smart object, say tr_{i_k}, belonging to a community C_k, decides to test another smart object te_{j_q}, belonging to a community C_q. As explained in Section 4.3, tr_{i_k} randomly chooses a support smart object, say $\overline{t_q}$, belonging to C_q. In turn, $\overline{t_q}$ has to select a pruned partition $\widehat{TrS_q}$ of smart objects of C_q that can answer the probing query. At this point, the following Slandering Attack attempts could be carried out:

1. The attacker tries to control $\widehat{TrS_q}$ in such a way that, after $\overline{t_q}$ sends req_{i_j} to the objects of $\widehat{TrS_q}$, it receives only false answers (or, at least, a great majority of false answers) from them. Of course, in this case, the computation of the trust score of te_{j_q} would be compromised. However, this scenario cannot happen due to Assumptions **A.1**, **A.2** and **A.3**. Indeed, according to them, the attacker cannot control the overall community and only $t < |\widehat{TrS_q}|$ smart objects can collude. As for the case in which a partial attack occurs, smart objects for testing are randomly chosen. This lowers the probability of selecting two or more colluding smart objects among the ones controlled by the attacker, which are at maximum t. Finally, in the very remote case in which all the t smart objects controlled by the attacker have been included in the partition, this malicious attempt would impact the single trust value computed for te_{j_q}. However, it does not affect the overall trust score yet. In fact, as already said, our metric is designed in such a way as to average all trust values. Therefore, no overall advantage is achieved by the attacker in this case.

2. $\overline{t_q}$ could send req_{i_j} to the smart objects of $\widehat{TrS_q}$ and, after having received the corresponding answers, it returns a corrupted average output. In this case, since the choice of $\overline{t_q}$ is random, the attacker cannot control this situation and design a global attack strategy. As a consequence, even in this case, our trust and reputation model is not compromised.

3. tr_{i_k} lies on the answers of te_{j_q} and $\overline{t_q}$. As explained in Section 4.3.1, in order to contrast this case, te_{j_q} and $\overline{t_q}$ can send their responses to randomly selected objects of C_k. These last ones will use the Local Blockchain to securely store such values.

5.2.5 SP.5 - Resistance to Opportunistic Service Attacks

In this case, a malicious smart object can provide good or bad services opportunistically. In our scenario, this attack can be designed as a partial Slandering Attack, in which a smart object acts well inside its community, whereas it acts maliciously when interacting with smart objects of other communities in order to lower the trustworthiness of its community. This could happen during the inter-group probing transactions, when a smart object chosen for the test, say $\overline{t_q}$, returns a corrupted average output. However, thanks to Assumptions **A.1**, **A.2** and **A.3**, since the choice of $\overline{t_q}$ is random, an attacker cannot design a global attack strategy and, hence, it cannot compromise the overall trustworthiness of the community.

5.2.6 SP.6 - Resistance to Ballot Stuffing Attacks

In this case, an attacker could boost the reputation of bad objects providing good recommendations for them to increase the chance that they are trusted by the community. The countermeasures for this kind of attack fall in the ones described for Slandering Attacks (see Section 5.2.4). Recall that, thanks to the use of the Blockchain technology, no smart object can corrupt or change responses by itself, either positively (in such a way as to increase its trust or reputation scores) or negatively (in such a way as to decrease the trust and reputation scores of other objects).

5.2.7 SP.7 - Resistance to Denial of Service (DoS) Attacks

In this case, attackers may cause Denial of Service preventing a reputation system from operating properly due to the flooding of an excessive number of transactions. A particular group of DoS attacks, very common in an IoT scenario, is represented by the Sleep Deprivation Attacks. In this case, the goal of an intruder is to maximize the power consumption of a victim in order to minimize its lifetime.

In general, our approach does not deal with DoS attacks. Hence, the strategies for preventing them are orthogonal to it, and any of these strategies, such as the ones presented in [15, 23, 87], could be adopted. For example, a naive strategy might operate as follows. Whenever a target smart object receives a suspect sequence of consecutive queries from a source one (it can use the communication history to classify anomalous probing activities), it starts to add a random delay in its answers to them. In case the anomalous probing continues over time, the target object stops answering any next query coming from the attacker for a certain time interval.

5.2.8 SP.8 - Resistance to Orchestrated Attack

In this case, malicious smart objects orchestrate their actions and leverage several of the previous strategies to perform a coordinated and multi-faced attack, which can change over time. All these types of attacks cannot happen thanks to Assumptions **A.1**, **A.2**, and **A.3**. Hence, an attacker cannot compromise an overall community or even a number of smart objects sufficient to conduct these attacks.

5.2.9 SP.9 - Resistance to malicious probing exploitation

In this case, a probing request is made against a node providing invasive services, like critical automation. First of all, it is worth observing that this kind of device can introduce important critical issues in the considered scenario. In fact, if an adversary gains access to these objects, the consequences of the actions that she/he could make may strongly impact on the safety of the environment. Think, for example, of objects such as smart kitchen appliances, like smart gas valves or electric cookers. For these devices, probing transactions can lead to dangerous actions if performed with respect to these invasive services. However, our solution does not introduce vulnerabilities that could provide advantages to an attacker who gained access or control of a smart object in the system. In fact, it leverages normal object-to-object communications to implement probing transactions. In this sense, a probing request does not differ from a real one. Consequently, an attacker who chooses to make a probing request through a smart object is not empowered with more functionalities than the ones she/he could obtain in a standard solution without using our approach. However, in this context, the probing strategy could represent a safety risk in itself.

Generally speaking, smart objects can be classified into sensors and actuators. Sensors provide sensing capabilities, measure well-defined physical indicators or collect information on their network and/or possible applications [37]. Actuators perform specific actions based on the inputs received. In our scenario, we are explicitly referring to modern smart objects for IoT. To achieve autonomy, these objects are equipped with both sets of monitoring sensors and a management module that controls object automation services. The probing tests we consider in our approach generally consist of measurements that can be reproduced and compared by means of the other related devices. Therefore, in scenarios characterized by modern smart objects, our solution can be configured in such a way that probing transactions leverage only the sensing capabilities of objects (and not on their capability of performing automation services). Consider, for example, a modern electric cooker. It generally has a management module to control cooking automation (e.g., to turn it on to starting cooking). However, it also has sensing modules, e.g., a module to measure the temperature in order to keep the food at an acceptable temperature with respect to the surrounding environment. In general, using only the sensing capabilities of objects for probing transactions can reduce the risks introduced by critical automation services.

However, in legacy IoT contexts, where several objects can be dummy actuators, our approach could be forced to rely on automation services. Once again we observe that, since probing transactions are based on normal object transactions, they do not introduce additional vulnerabilities. Nevertheless, the vulnerability related to the need to use dummy actuators remains. Consider, for example, the case in which the object to be tested is a legacy smart gas valve. Of course, opening a valve is a critical action and if an attacker were to gain access to this object, a big safety problem could arise. For this reason, it is worth carrying out the probing transactions only in conjunction with normal ones in such a way as not to increase the number of occurrences in which the dummy actuator carries out its actions.

Given this premise, our approach works using the normal interactions between other objects and the dummy actuator to assess the reliability of the latter. In fact, with a configurable probability degree, the querying object will perform the probing task along with the normal transaction. For dummy actuators, a transaction is in any case a request to perform the actions associated with them. In this case, the partition of support nodes, engaged

to check the trustworthiness of the queried node, must verify that the action was performed correctly. Therefore, this partition should contain objects that provide sensing services compatible with the action performed by the tested node. For example, consider the case where the action performed by an actuator is switching on a light bulb. In this case, the support partition for probing could consist of smart cameras, smart light detectors, etc.

To cope with this setting, the only necessary change in our strategy concerns the fact that, in Equation 1, the value out_j is not the measure returned by the probed node te_j, but the expected variation of a suitable measurable quantity corresponding to the impact of the environment caused by this action. In the previous example, switching on a smart bulb would increase the brightness of the environment related to the total amount of visible light that the bulb is able to emit in the unit of time.

Finally, we observe that, since our probing mechanism is triggered only when a normal transaction is made between a generic object and the dummy actuator, there might be an impact in terms of the time required to collect enough probing results to measure a degradation in the reputation of the dummy actuator. Furthermore, the interaction with the suitable partition of smart objects involved to assess the quality of the action performed by the dummy actuator could involve a larger number of transactions than in the case where a simple measurement sensed by an object is tested. We performed some tests to evaluate these aspects. They are shown in Section 6.2.

6 Experiments

In this section, we report the experiments we have carried out to test the effectiveness and the performance of our proposal. Specifically, in Subsection 6.1, we describe the dataset adopted. In Subsection 6.2, we analyze the performance of our approach. Finally, in Subsection 6.3, we compare it with other related ones previously proposed in literature.

6.1 Dataset Description

In order to test the effectiveness of our approach we needed both a prototype (that we realized) and a dataset. As real datasets with information about IoT transactions on a two-tier Blockchain do not exist yet, we built a simulator. To make "concrete" and "realistic" the simulated scenario, we leveraged real-life datasets.

In order to perform our task, we needed two main pieces of information, namely: *(i)* data exchanged among the smart objects of the IoT during a given time interval; *(ii)* data about real Blockchain transactions. We employed a complete online report about IoT data exchanges across several domains, available online at the Zscaler company website[5]. We joined this information with data available from a complete dataset of US Ethereum transactions, obtained at the address `https://console.cloud.google.com/marketplace/details/ethereum/crypto-ethereum-Blockchain?pli=1`.

By proceeding in this way, our final dataset contained information about both the number of transactions performed by IoT smart objects during a month and the actual time required for these transactions to be also stored in a real-life Blockchain. Table 6.1 shows an example of our dataset. Here, *Source Object* and *Destination*

[5] `https://www.zscaler.com/threatlabz/iot-dashboard`

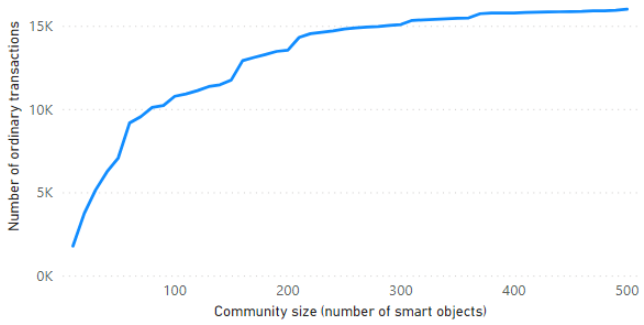


Figure 4: Number of ordinary transactions performed in a month against community size

Object are the identifiers of a transaction end-points; *Timestamp* is the time instant in epoch when a transaction took place; finally, *Duration* represents the transaction execution time in seconds.

Source Object	*Destination Object*	*Timestamp*	*Duration*
1	3	1575158400	0.025
2	4	1575163800	0.028
4	6	1575167220	0.022
...	...	...	...

Table 2: An example of our dataset

Using the above dataset, we were able to simulate different configurations of our multi-IoT framework. Specifically, we simulated different combinations of smart object communities and object interactions. To measure the impact of probing transactions, as well as smart contract execution times, we built our prototype on top of a real-life public Blockchain. In this way, we had the possibility to experiment probing traffic impact according to our two-tier Blockchain model. In our experiments, we adopted Hyperledger as referring platform.

Figure 4 reports the number of ordinary transactions (i.e., those performed to obtain a feature/service and not for probing goals) performed in a month against the community size. The average time necessary to execute all the ordinary transactions of a month against community size is reported in Figure 5.

6.2 Performance analysis of our approach

The first experiment that we carried out was devoted to test the efficiency of our approach. To do so, we focused on the most costly operation, which is the computation of trust values between pairs of smart objects inside communities. We recall that, to create a safe and controlled domain inside each community, smart objects are forced to perform tests on other members of their community randomly selected according to a given probability. We measured the overhead in terms of both the number of generated transactions and the time spent to perform tests. Let p be the probing probability, i.e. the probability for a smart object to generate a test towards another one. We considered a variable size of communities, ranging from 10 to 500 smart objects, and five different values of the probing probability, i.e. $p = 0.1$, $p = 0.2$, $p = 0.3$, $p = 0.4$, and $p = 0.5$. The results obtained are

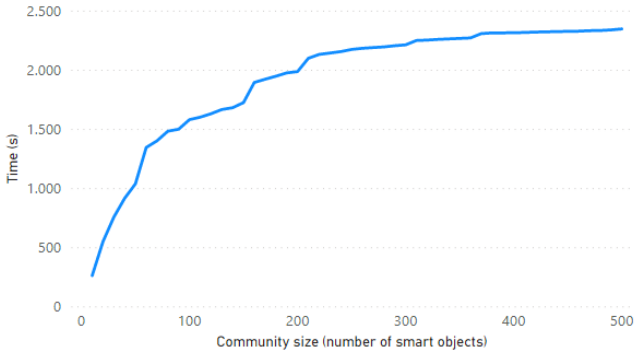


Figure 5: Average time necessary to execute all the ordinary transactions of a month against community size

reported in Figures 6-7.

In these figures, blue lines represent the cost of the ordinary transactions, whereas red lines denote the costs of the probing ones. In more detail, each box corresponds to one of the possible values of p and reports two graphics. The top one compares the execution time of ordinary transactions (in blue) and probing transactions (in red). The bottom one, instead, compares the number of ordinary and probing transactions. This figure suggests that as p increases the overhead introduced by our approach grows linearly reaching the same value as the one of the ordinary transactions when $p = 0.5$; in this last case, the effort to maintain object interaction is doubled.

At this point, to properly tune our framework, we performed a further experiment with the aim of computing the time required by communities to identify (and, hence, remove) an attacked smart object. We carried out this task considering the same probing probabilities analyzed in the previous experiment. Furthermore, we fixed the size of communities to 100 smart objects and we forced our framework to recompute all reputation values after every probing transaction inside a community. Figure 8 reports the trend of the reputation decay of an attacked smart object over time. In this figure, each plot corresponds to a value of the probing probability. This figure shows that, as p increases, the reputation decay curve is increasingly steep, as more tests will be executed in a very small time interval. If we assume a value of 0.6 as the minimum reputation for a node to be a member of a group, we can see that the reputation of the attacked node goes under the minimum threshold after less than 4 seconds in all the five plots of Figure 8. The lowest time of about 2 seconds is reached for $p = 0.5$.

Now, in Sections 4.2.1 and 4.2.2, we have seen that, in a real world scenario, the propagation of local trust values and, hence, the computation of node reputations cannot be performed continuously. Indeed, this activity implies the activation of a dedicated smart contract requiring computational efforts to Blockchain peers. To avoid this situation, in our approach, we defined a time window tuning the activation frequency of the above smart contract. The objective of this way of proceeding is limiting the activation frequency of the above smart contract, on the one hand, and controlling the dimension of the Local Blockchain (before aggregating all probing transactions and resetting it), on the other hand.

As a consequence, in a real life scenario, the value of the size of the time window should not be too low.

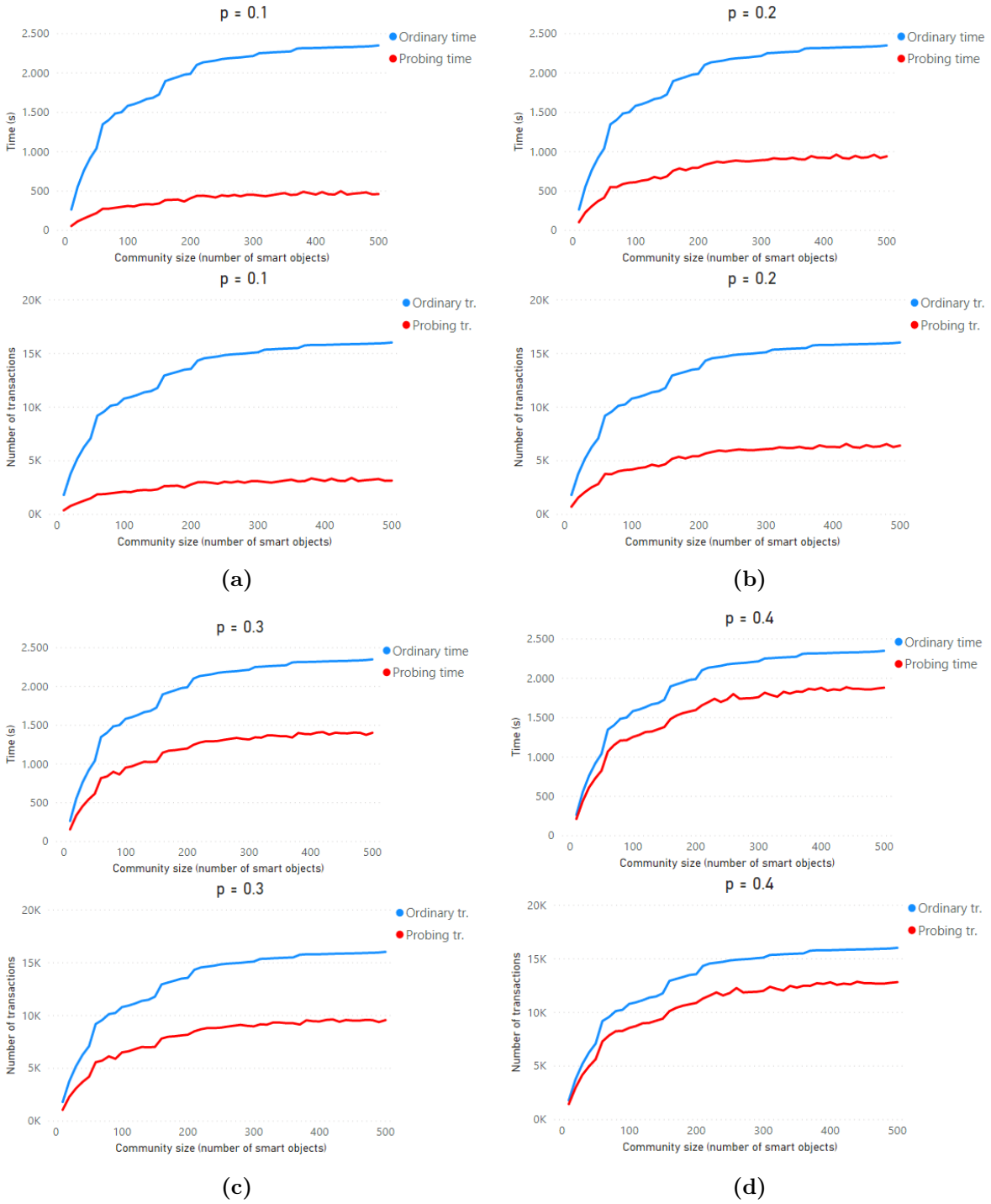
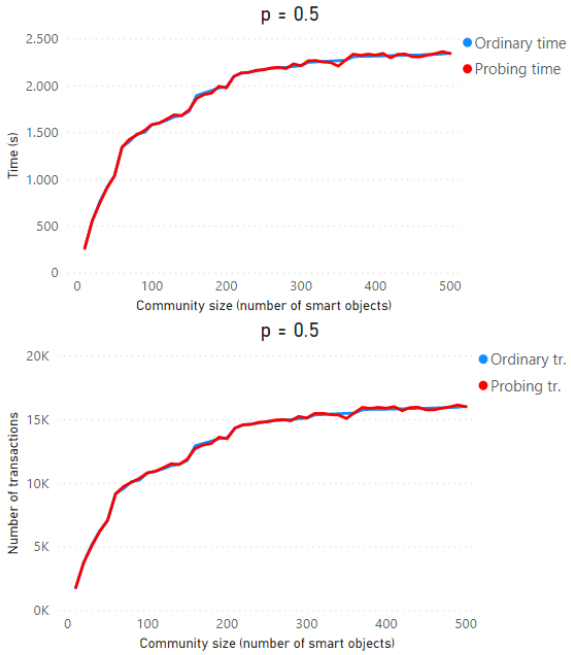


Figure 6: Number of transactions in a month and time necessary to execute them against community size and probing probability - Part I

In a related study, in which Blockchain transactions are aggregated to control the size of the chain, the time interval for the aggregation is set to 3600 seconds [78]. Of course, we could set the same size, even if, in presence



(e)

Figure 7: Number of transactions in a month and time necessary to execute them against community size and probing probability - Part II

of specific security requirements (i.e., an attacked node must be isolated in less time than an hour), we could reduce it accordingly. Therefore, starting from the value reported in [78], we could use a heuristic based on the Elbow method [52] to reduce this value and the size of the time window in such a way as to satisfy both the requirement on the size of Local Blockchains and the security constraints. Anyway, thanks to the experiment described above, we proved that, in a common IoT scenario, with p set to its lowest value (i.e., $p = 0.1$), only 4 seconds are necessary to collect the probing transactions needed to reduce the reputation of an attacked node and detect it as malicious. Therefore, for any value of the time window size greater than 4 seconds, we can set $p = 0.1$ without losing detection precision. This choice preserves the framework usability, because a negligible overhead will be generated, and still guarantees a satisfactory performance from the security point of view.

As a final experiment, we considered the scenario, described in Section 5.2.9, in which the network also includes legacy devices that provide only automation services (we previously called this type of devices "dummy actuators"). As seen in Section 5.2.9, the presence of dummy actuators could cause an increase in the number of transactions required with objects in the support partition to properly assess the quality of the action performed. Furthermore, this presence could lead to an increase in the overall time required to collect a sufficient number of probing results.

To carry out this experiment, we considered a community consisting of 100 nodes and ran the simulation for the same number of ordinary transactions seen above. In our experiment, we chose different percentages of involved dummy actuators (ranging from 5% to 20%). The results obtained are shown in Figures 9 and 10.

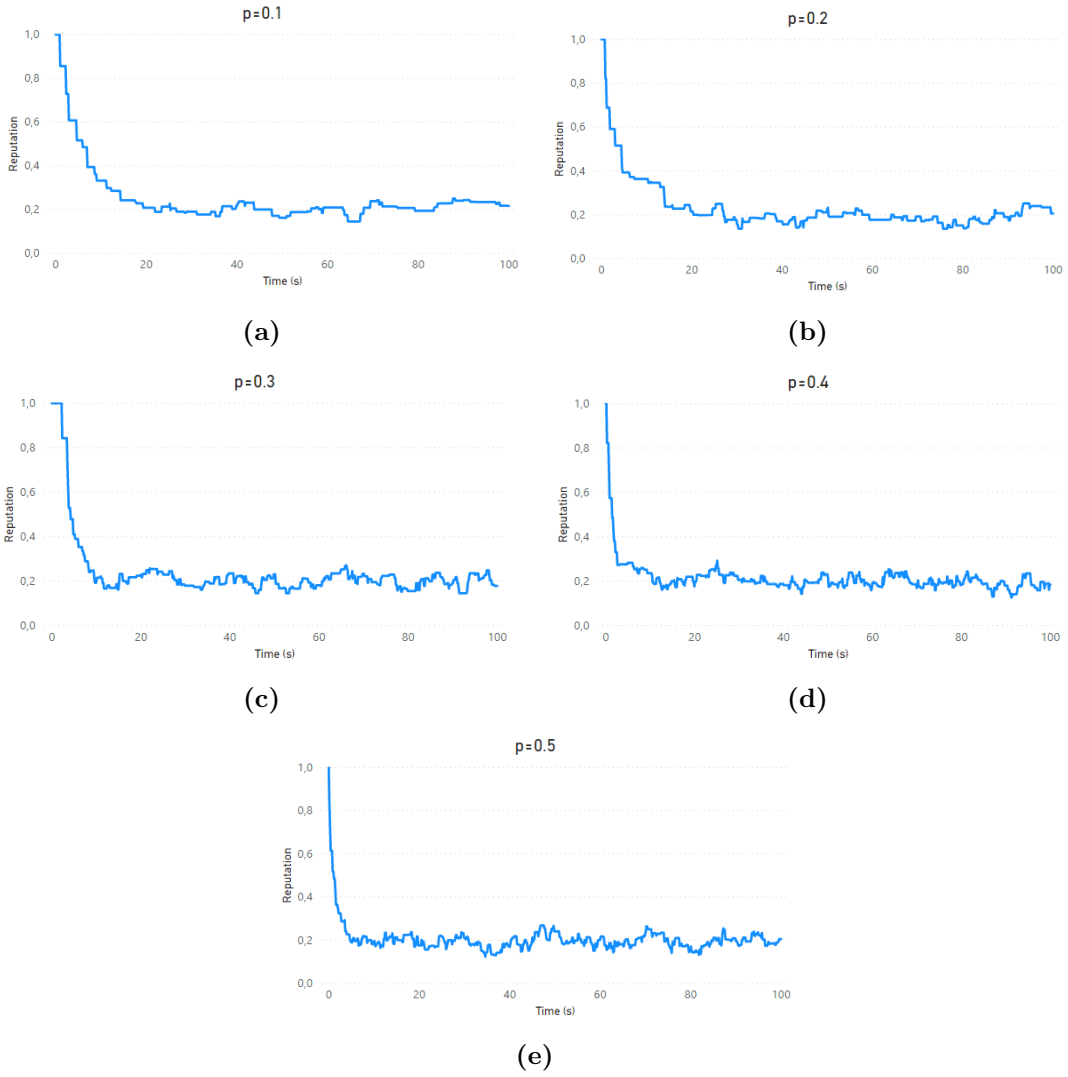


Figure 8: Reputation decay for a malicious smart object inside a community of 100 components

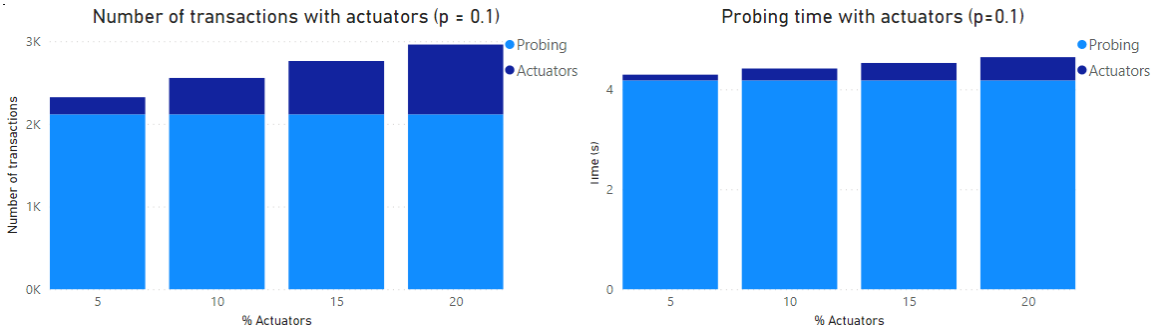


Figure 9: Number of probing transactions and probing time with dummy actuators ($p = 0.1$)

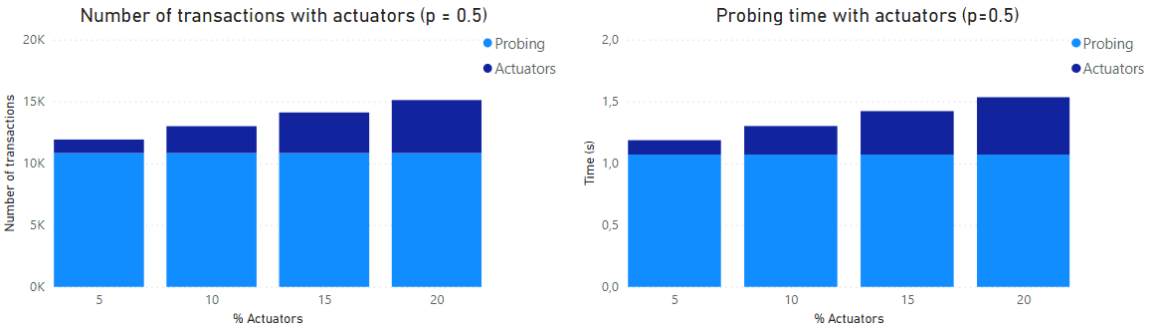


Figure 10: Number of probing transactions and probing time with dummy actuators ($p = 0.5$)

From the analysis of this figure, we can observe that, as we expected, the presence of the dummy actuators leads to an increase in the number of transactions required to complete the probing activities. This increase ranges from 9.78% to 39.88% for $p = 0.1$ and from 10.07% to 41.05% for $p = 0.5$, depending on the percentage of the dummy actuators (the smaller the percentage, the smaller the increase). This increment, although not always negligible, is anyway acceptable, also because the highest values are obtained in correspondence of very high percentages of dummy actuators. We can also observe an increase of the average time necessary to collect the number of probing results needed to reduce the reputation of an attacked node and identify it as malicious. Specifically, the average additional time ranges from 2.63% to 11.24% for $p = 0.1$ and from 11.21% to 42.98% for $p = 0.5$ (again, the smaller the percentage, the smaller the increment). This increase is negligible for $p = 0.1$, which is the configuration suggested by us. It is not negligible for $p = 0.5$, especially in presence of a high percentage of dummy actuators. However, we observe that the configuration $p = 0.5$ is extreme; it certainly has a theoretical interest but is very far from the one we suggest for real cases (i.e., $p = 0.1$).

6.3 Comparison with other approaches

The aim of this experiment is comparing our approach with other related ones proposed in past literature. In particular, we selected two related approaches having different goals but sharing several similarities with ours in both the reference scenario and the adopted methodology.

The first selected approach [5] regards an intrusion detection system useful to protect smart devices in vehicular networks. The main idea proposed by the authors is grouping nodes into "clusters" to identify protected zones where security is achieved with nodes collaboration. Even though the aim of this approach is quite different from ours, they are similar in two aspects, namely: *(i)* the definition of a security model operating on smart devices and IoT, and *(ii)* the usage of groups and clusters of things (corresponding to communities of smart objects in our model).

The second selected approach [64] deals with an orthogonal issue, that is the modeling of a privacy preserving object grouping scheme. This guarantees the protection of user's privacy in all those IoT scenarios where the knowledge of the object features may help an attacker to collect information about user habit and behavior.

In order to compare our approach with the ones of [5] and [64], we measured the communication delay introduced by the evaluated approach against the community size. The communication delay refers to the

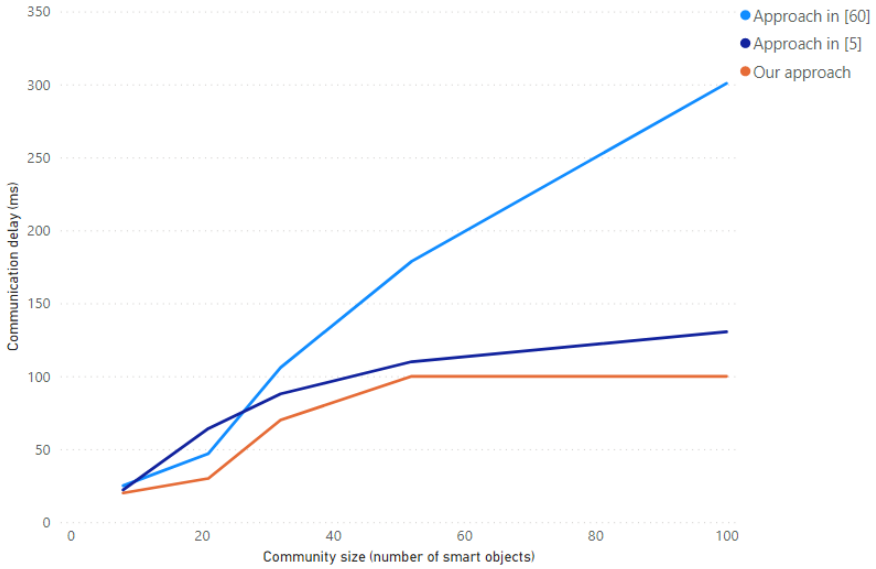


Figure 11: Comparison of the average delay against the community size between our approach and the ones of [5] ad [64]

latency rate introduced by the activation of the evaluated approach in the considered application scenario. Basically, it consists of the increase of the delay in processing and delivering a specific service. In our approach, we defined this parameter as the average difference, in terms of delivery time, between a scenario in which our approach is used and another one where it is not adopted. The results obtained are shown in Figure 11.

This figure shows that the average delay introduced by our approach ranges from 20 ms to 100 ms, whereas the one of the approach of [5] ranges between 24 ms and 150 ms and the one of the approach of [64] ranges between 22 ms and 300 ms. This result highlights that the performance of our approach is comparable with, and even better than, the ones of the approaches described in [5] and [64]. Hence, we can state that our approach achieves good results still maintaining the overall IoT overhead to considerable low values.

7 Conclusion

In this paper, we have proposed a two-tier Blockchain framework conceived to increase protection and autonomy of smart objects in the IoT. First of all, we have seen the motivations underlying our decision to address this issue. Then, we have examined related literature and we have pointed out the main differences and novelties of our approach with respect to the past ones. Afterwards, we have proposed a reference model which both our framework and the algorithms operating in it are based on. Next, we have illustrated our approach to compute the trust of a smart object in another one, the reputation of a smart object in its community and the trust of a community in another one. After this, we have presented the security model that can be activated by means of our framework. Finally, we have illustrated several experiments devoted to evaluate the performance of our approach and to compare it with two other ones already presented in the past literature.

The research issues addressed in this paper must not be considered as an ending point. Instead, they represent the starting point of further efforts that we plan to perform in the future along several directions. For instance, we plan to combine our approach with other community-based ones conceived to ensure the privacy of smart objects and their owners, with the ultimate goal to define a single solution handling both privacy and security in IoT. Furthermore, we would like to extend our approach adding the possibility to protect the authenticity of the services offered by smart objects. In fact, we have not currently considered how nodes advertise and, then, deliver their services; therefore, we have not taken into account that they might lie about this. Last, but not the least, we plan to improve the computation of object and community reliability using machine learning techniques that can also predict the type of content the requester expects to receive, based on its past history. In this way, the reliability computation would depend not only on technological aspects but also on semantic evaluations.

Acknowledgments

This work was partially funded by the Department of Information Engineering at the Polytechnic University of Marche under the project "An integrated approach for innovative and eco-sustainable freight transport solutions in emergency and last mile logistics" (RSAB 2020), and by the Marche Region under the project "Human Digital Flexible Factory of the Future Laboratory (HDSFIab) - POR MARCHE FESR 2014-2020 - CUP B16H18000050007".

References

[1] W. Abdelghani, C.A. Zayani, I. Amous, and F. Sèdes. Trust management in social internet of things: a survey. In *Proc. of the International Conference on e-Business, e-Services and e-Society (IFIP'16)*, pages 430–441, Swansea, United Kingdom, 2016. Springer.

[2] M. Abomhara and G.M. Køien. Security and privacy in the Internet of Things: Current status and open issues. In *Proc. of the International Conference on Privacy and Security in Mobile Systems (PRISMS'14)*, pages 1–8, Aalborg, Denmark, 2014. IEEE.

[3] F. Al-Turjman and S. Alturjman. Context-sensitive access in industrial internet of things (iiot) healthcare applications. *IEEE Transactions on Industrial Informatics*, 14(6):2736–2744, 2018. IEEE.

[4] F. Al-Turjman, H. Zahmatkesh, and R. Shahroze. An overview of security and privacy in smart cities' iot communications. *Transactions on Emerging Telecommunications Technologies*, page e3677, 2019. Wiley Online Library.

[5] M. Aloqaily, S. Otoum, I. Al Ridhawi, and Y. Jararweh. An intrusion detection system for connected vehicles in smart cities. *Ad Hoc Networks*, 90:101842, 2019. Elsevier.

[6] Q.M. Ashraf and M. H. Habaebi. Introducing autonomy in internet of things. In *Proc. of the International Conference on Applied Computer and Applied Computational Science (ACACOS'15)*, Kuala Lumpur, Malaysia, 2015.

[7] L. Atzori, A. Iera, and G. Morabito. From "smart objects" to "social objects": The next evolutionary step of the Internet of Things. *IEEE Communications Magazine*, 52(1):97–105, 2014. IEEE.

[8] L. Atzori, A. Iera, and G. Morabito. Understanding the Internet of Things: definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*, 56:122–140, 2017. Elsevier.

[9] L. Atzori, A. Iera, G. Morabito, and M. Nitti. 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, 2012. Elsevier.

[10] S. Bandyopadhyay, M. Sengupta, S. Maiti, and S. Dutta. A survey of middleware for Internet of Things. In *Recent trends in wireless and mobile networks*, pages 288–296. Springer, 2011.

[11] F. Bao and R. Chen. Trust management for the Internet of Things and its application to service composition. In *Proc. of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'12)*, pages 1–6, San Francisco, CA, USA, 2012. IEEE.

[12] F. Bao, R. Chen, and J. Guo. Scalable, adaptive and survivable trust management for community of interest based Internet of Things systems. In *Proc. of the IEEE International Symposium on Autonomous Decentralized Systems (ISADS'13)*, pages 1–7, Mexico City, Mexico, 2013. IEEE.

[13] F. Bao and I. Cheny. Dynamic trust management for internet of things applications. In *Proc. of the International Workshop on Self-aware Internet of Things (ICAC'12)*, pages 1–6, San Jose, CA, USA, 2012. ACM.

[14] K. Biswas and V. Muthukkumarasamy. Securing smart cities using blockchain technology. In *Proc. of the IEEE International Conference on High pPerformance Computing and Communications; IEEE International Conference on Smart City; IEEE International Conference on Data Science and Systems (HPCC/SmartCity/DSS 2016)*, pages 1392–1393, Sydney, Australia, 2016. IEEE.

[15] M. Bouabdellah, N. Kaabouch, F. El Bouanani, and H. Ben-Azza. Network layer attacks and countermeasures in cognitive radio networks: A survey. *Journal of Information Security and Applications*, 38:40–49, 2018. Elsevier.

[16] F. Buccafurri, L. Coppolino, S. D'Antonio, A. Garofalo, G. Lax, A. Nocera, and L. Romano. Trust-Based Intrusion Tolerant Routing in Wireless Sensor Networks. In *Proc. of the International Conference on Computer Safety, Reliability and Security (SAFECOMP 2014)*, pages 214–229, Firenze, Italy, 2014. Springer.

[17] F. Buccafurri, G. Lax, S. Nicolazzo, and A. Nocera. Accountability-Preserving Anonymous Delivery of Cloud Services. In *Proc. of the International Conference on Trust, Privacy and Security in Digital Business (TRUSTBUS 2015)*, pages 124–135. Springer, 2015.

[18] A. Chauhan, O. P. Malviya, M. Verma, and T. S. Mor. Blockchain and scalability. In *Proc. of the IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C 2018)*, pages 122–128, Lisbon, Portugal, 2018. IEEE.

[19] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang. TRM-IoT: A trust management model based on fuzzy reputation for Internet of Things. *Computer Science and Information Systems*, 8(4):1207–1228, 2011. ComSIS Consortium.

[20] H. Chen, P. Han, B. Yu, and C. Gao. A new kind of session keys based on message scheme for sensor networks. In *Proc. of the International Asia-Pacific Microwave Conference (APMC'05)*, volume 1, pages 4–pp, Suzhou, China, 2005. IEEE.

[21] H. Chen, H. Wu, X. Zhou, and C. Gao. Agent-based trust model in wireless sensor networks. In *Proc. of the ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, volume 3, pages 119–124, Qingdao, China, 2007. IEEE.

[22] J. Chen, Z. Tian, X. Cui, L. Yin, and X. Wang. Trust architecture and reputation evaluation for internet of things. *Journal of Ambient Intelligence and Humanized Computing*, 10(8):3099–3107, 2019. Springer.

[23] N. Chouhan, H.K. Saini, and S.C. Jain. Internet of Things: Illuminating and Study of Protection and Justifying Potential Countermeasures. In *Soft Computing and Signal Processing*, pages 21–27. Springer, 2019.

[24] K. Christidis and M. Devetsikiotis. Blockchains and smart contracts for the Internet of Things. *IEEE Access*, 4:2292–2303, 2016. IEEE.

[25] R. Cramery, R. Gennaroz, and B. Schoenmakersx. A secure and optimally efficient multi-authority election scheme. *European transactions on Telecommunications*, 8(5):481–490, 1997.

[26] A.K. Dalai and S.K. Jena. Wdtf: A technique for wireless device type fingerprinting. *Wireless Personal Communications*, 97(2):1911–1928, 2017.

[27] T. Dasu, Y.Kanza, and D. Srivastava. Unchain your blockchain. In *Proc. of the International Symposium on Foundations and Applications of Blockchain (FAB'18)*, volume 1, pages 16–23, Los Angeles, CA, USA, 2018.

[28] J. Deng, R. Han, and S. Mishra. A performance evaluation of intrusion-tolerant routing in wireless sensor networks. In *Information Processing in Sensor Networks*, pages 349–364, 2003. Springer.

[29] C. Diamantini, A. Nocera, D. Potena, E. Storti, and D. Ursino. Find the Right Peers: Building and Querying Multi-IoT Networks Based on Contexts. In *Proc. of the International Conference on Flexible Query Answering Systems (FQAS'19)*, pages 302–313, Amantea, Italy, 2019. Springer.

[30] A. Dorri, S. Kanhere, R. Jurdak, and P. Gauravaram. LSB: A Lightweight Scalable Blockchain for IoT security and anonymity. *Journal of Parallel and Distributed Computing*, 134:180–197, 2019.

[31] A. Dorri, S.S. Kanhere, and R. Jurdakand P. Gauravaram. Blockchain for IoT security and privacy: The case study of a smart home. In *Proc. of the IEEE international Conference on Pervasive Computing and Communications Workshops (PerCom'17 workshops)*, pages 618–623, Kona, HI, USA, 2017. IEEE.

[32] A. Dorri, S.S. Kanhere, and R. Jurdak. Towards an optimized blockchain for IoT. In *Proc. of the International Conference on Internet-of-Things Design and Implementation (IoTDI'17)*, pages 173–178, Pittsburgh, PA, USA, 2017. IEEE.

[33] A. Dorri, S.S. Kanhere, R. Jurdak, and P. Gauravaram. Lsb: A lightweight scalable blockchain for iot security and privacy. *Journal of Parallel and Distributed Computing*, 134:180–197, 2017. Elsevier.

[34] J. R. Douceur. The Sybil attack. In *Proc. of the International Workshop on Peer-To-Peer Systems (IPTPS'02)*, pages 251–260, Cambridge, MA , USA, 2002. Springer.

[35] R. Duan, X. Chen, and T. Xing. A QoS architecture for IOT. In *Proc. of the International Conference on Internet of Things and International Conference on Cyber, Physical and Social Computing (CPSCom'11)*, pages 717–720, Dalian, China, 2011. IEEE.

[36] A.D. Dwivedi, G. Srivastava, S. Dhar, and R. Singh. A decentralized privacy-preserving healthcare blockchain for IoT. *Sensors*, 19(2):326, 2019. Multidisciplinary Digital Publishing Institute.

[37] EU ENISA. Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures, 2017.

[38] I. Eyal, A.E. Gencer, E.G. Sirer, and R. Van Renesse. Bitcoin-ng: A scalable blockchain protocol. In *Proc. of the International Symposium on Networked Systems Design and Implementation (NSDI'16)*, pages 45–59, Boston, MA, USA, 2016. ACM.

[39] A. Floris and L. Atzori. Quality of Experience in the Multimedia Internet of Things: Definition and practical use-cases. In *Proc. of the IEEE International Conference on Communication Workshop (ICCW'15)*, pages 1747–1752, London, United Kingdom, 2015. IEEE.

[40] P. Fouque, G. Poupard, and J. Stern. Sharing decryption in the context of voting or lotteries. In *Proc. of the International Conference on Financial Cryptography (FC'00)*, pages 90–104, Anguilla, Anguilla, 2000. Springer.

[41] S. Ganeriwal, L.K. Balzano, and M.B. Srivastava. Reputation-based framework for high integrity sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 4(3):1–37, 2008.

[42] S. Ganeriwal, R. Kumar, C.C. Han, S. Lee, and M.B. Srivastava. Location & Identity based Secure Event Report Generation for Sensor Networks. *NESL Technical Report*, 2004.

[43] I.D. Guedalia, J. Guedalia, R.P. Chandhok, and S. Glickfield. Methods to discover, configure, and leverage relationships in Internet of Things (IoT) networks, February 20 2018. US Patent 9,900,171.

[44] J. Guo, R. Chen, and J. JP. Tsai. A survey of trust computation models for service management in internet of things systems. *Computer Communications*, 97:1–14, 2017. Elsevier.

[45] M. Hauswirth, A. Datta, and K. Aberer. Handling identity in peer-to-peer systems. In *14th International Workshop on Database and Expert Systems Applications, 2003. Proceedings.*, pages 942–946. IEEE, 2003.

[46] F. Hendrikx, K. Bubendorfer R., and Chard. Reputation systems: A survey and taxonomy. *Journal of Parallel and Distributed Computing*, 75:184–197, 2015. Elsevier.

[47] K. Hoffman, D. Zage, and C. Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys*, 42(1):1–31, 2009. ACM New York, NY, USA.

[48] W. Huang, J. Hu, S. Lin, T. Liu, P. Tsai, C. Yang, F.I. Yeh, J.H. Chen, and J.J. Mambretti. Design and implementation of an automatic network topology discovery system for the future internet across different domains. In *Proc. of the International Conference on Advanced Information Networking and Applications Workshops (AINA'12)*, pages 903–908, Fukuoka, Japan, 2012. IEEE.

[49] S. Huh, S. Cho, and S. Kim. Managing IoT devices using blockchain platform. In *Proc. of the International Conference on Advanced Communication Technology (ICACT'17)*, pages 464–467, PyeongChang, Korea, 2017. IEEE.

[50] H. Jin, X. Dai, and J. Xiao. Towards a novel architecture for enabling interoperability amongst multiple blockchains. In *Proc. of the IEEE International Conference on Distributed Computing Systems (ICDCS'18)*, pages 1203–1211, Vienna, Austria, 2018. IEEE.

[51] C. Karlof, N. Sastry, and D. Wagner. TinySec: a link layer security architecture for wireless sensor networks. In *Proc. of the International Conference on Embedded Networked Sensor Systems (SenSys'04)*, pages 162–175, Baltimore, MD, USA, 2004. ACM.

[52] D.J. Ketchen and C.L. Shook. The application of cluster analysis in strategic management research: an analysis and critique. *Strategic Management Journal*, 17(6):441–458, 1996. Wiley Online Library.

[53] M.A. Khan and K. Salah. IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, 82:395–411, 2018. Elsevier.

[54] A. Konev, R. Khaydarova, M. Lapaev, L. Feng, L. Hu, M. Chen, and I. Bondarenko. CHPC: A complex semantic-based secured approach to heritage preservation and secure IoT-based museum processes. *Computer Communications*, 148:240–249, 2019.

[55] N. Labraoui, M. Gueroui, and L. Sekhri. A risk-aware reputation-based trust management in wireless sensor networks. *Wireless Personal Communications*, 87(3):1037–1055, 2016.

[56] L. Lao, Z. Li, S. Hou, B. Xiao, S. Guo, and Y. Yang. A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling. *ACM Computing Surveys (CSUR)*, 53(1):1–32, 2020. ACM New York, NY, USA.

[57] J. Lin, Z. Shen, and C. Miao. Using blockchain technology to build trust in sharing LoRaWAN IoT. In *Proc. of the International Conference on Crowd Science and Engineering (ICCSE'17)*, pages 38–43, Beijing, China, 2017. ACM.

[58] Y. Liu, K. Wang, Y. Lin, and W. Xu. LightChain: A lightweight blockchain system for industrial Internet of Things. *IEEE Transactions on Industrial Informatics*, 15(6):3571–3581, 2019. IEEE.

[59] X. Ma and H. Xue. Intelligent smart city parking facility layout optimization based on intelligent IoT analysis. *Computer Communications*, 153:145–151, 2020.

[60] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan. Internet of things (IoT) security: Current status, challenges and prospective measures. In *Proc. of the International Conference for Internet Technology and Secured Transactions (ICITST'15)*, pages 336–341, London, United Kingdom, 2015. IEEE.

[61] A. Majeed and A. Al-Yasiri. Formulating a global identifier based on actor relationship for the internet of things. In *Interoperability, Safety and Security in IoT*, pages 79–91. Springer, 2016.

[62] S. Marti and H. Garcia-Molina. Taxonomy of trust: Categorizing P2P reputation systems. *Computer Networks*, 50(4):472–484, 2006. Elsevier.

[63] Thomas Moellers. IOTA-based Business Model Configurations. *https://www.alexandria.unisg.ch/257117/*, 2018.

[64] S. Nicolazzo, A. Nocera, D. Ursino, and L. Virgili. A privacy-preserving approach to prevent feature disclosure in an iot scenario. In *Future Generation Computer Systems*, volume 105, pages 1–8, 2019. IEEE.

[65] B. Nour, K. Sharif, F. Li, H. Moungla, and Y. Liu. A unified hybrid information-centric naming scheme for IoT applications. *Computer Communications*, 150:103–114, 2020.

[66] O. Novo. Blockchain meets IoT: An architecture for scalable access management in IoT. *IEEE Internet of Things Journal*, 5(2):1184–1195, 2018. IEEE.

[67] P. Oser, F. Kargl, and S. Lüders. Identifying devices of the internet of things using machine learning on clock characteristics. In *International conference on security, privacy and anonymity in computation, communication and storage*, pages 417–427. Springer, 2018.

[68] P. Otte, M. de Vos, and J. Pouwelse. TrustChain: A Sybil-resistant scalable blockchain. *Future Generation Computer Systems*, 107(48):770–780, 2017. Elsevier.

[69] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, and D.E. Culler. SPINS: Security protocols for sensor networks. *Wireless networks*, 8(5):521–534, 2002. Springer.

[70] R. Di Pietro, X. Salleras, M. Signorini, and E. Waisbard. A blockchain-based Trust System for the Internet of Things. In *Proc. of the ACM International Symposium on Access Control Models and Technologies (SACMAT'18)*, pages 77–83, Indianapolis, IN, USA, 2018. ACM.

[71] S. Popov. The tangle. *White paper*, 1:3, 2018.

[72] G. Pujolle. An autonomic-oriented architecture for the internet of things. In *Proc. of the IEEE John Vincent Atanasoff 2006 International Symposium on Modern Computing (JVA'06)*, pages 163–168, Sofia, Bulgaria, 2006. IEEE.

[73] J. Quevedo, M. Antunes, D. Corujo, D. Gomes, and R.L. Aguiar. On the application of contextual iot service discovery in information centric networks. *Computer Communications*, 89:117–127, 2016. Elsevier.

[74] M. Rehman, N. Javaid, M. Awais, M. Imran, and N. Naseer. Cloud based secure service providing for IoTs using blockchain. In *Proc. of the IEEE Global Communications Conference (GLOBCOM 2019)*, pages 1–7, Puako, Hawaii, USA, 2019.

[75] M. Samaniego and R. Deters. Blockchain as a Service for IoT. In *Proc. of the International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 433–436, Chengdu, China, 2016. IEEE.

[76] M. Samaniego and R. Deters. Using blockchain to push software-defined IoT components onto edge hosts. In *Proc. of the International Conference on Big Data and Advanced Wireless Technologies (BDAW'16)*, page 58, Blagoevgrad, Bulgaria, 2016. ACM.

[77] C. G. Schmidt and S. M. Wagner. Blockchain and supply chain relations: A transaction cost theory perspective. *Journal of Purchasing and Supply Management*, 25(4):100552, 2019.

[78] A.R. Shahid, N. Pissinou, C. Staier, and R. Kwan. Sensor-Chain: A Lightweight Scalable Blockchain Framework for Internet of Things. In *Proc. of the International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1154–1161, Atlanta, GE, USA, 2019. IEEE.

[79] M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani. Privacy-Preserving Support Vector Machine Training over Blockchain-Based Encrypted IoT Data in Smart Cities. *IEEE Internet of Things Journal*, 2019. IEEE.

[80] S. Sicari, A. Rizzardi, L.A. Grieco, and A. Coen-Porisini. Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks*, 76:146–164, 2015. Elsevier.

[81] W.F. Silvano and R. Marcelino. Iota Tangle: A cryptocurrency to communicate Internet-of-Things data. *Future Generation Computer Systems*, 112:307–319, 2020. Elsevier.

[82] A. Srinivasan, J. Teitelbaum, and J. Wu. DRBTS: distributed reputation-based beacon trust system. In *Proc. of the IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC'06)*, pages 277–283, Indianapolis, IN, USA, 2006. IEEE.

[83] C. Thirumalai, S. Mohan, and G. Srivastava. An efficient public key secure scheme for cloud and IoT security. *Computer Communications*, 150:634–643, 2020.

[84] Q. Wei and Z. Jin. Service discovery for internet of things: a context-awareness perspective. In *Proc. of the Fourth Asia-Pacific Symposium on Internetware (Internetware)*, pages 1–6, Qingdao, China, 2012.

[85] M. Wu, K. Wang, X. Cai, S. Guo, M. Guo, and C. Rong. A comprehensive survey of blockchain: From theory to IoT applications and beyond. *IEEE Internet of Things Journal*, 6(5):8114–8154, 2019. IEEE.

[86] Z. Yan, P. Zhang, and A.V. Vasilakos. A survey on trust management for Internet of Things. *Journal of Network and Computer Applications*, 42:120–134, 2014. Elsevier.

[87] W. Yang, Y. Wang, Z. Lai, Y. Wan, and Z. Cheng. Security Vulnerabilities and Countermeasures in the RPL-based Internet of Things. In *Proc. of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC'18)*, pages 49–495, Henan, China, 2018. IEEE.

[88] Y. Yu, K. Li, W. Zhou, and P. Li. Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures. *Journal of Network and computer Applications*, 35(3):867–880, 2012.

[89] J. Zhang, W. Li, N. Han, and J. Kan. Forest fire detection system based on a ZigBee wireless sensor network. *Frontiers of Forestry in China*, 3(3):369–374, 2008.

[90] Z. Zhang, M.C.Y. Cho, C. Wang, C. Hsu, C. Chen, and S. Shieh. IoT security: ongoing challenges and research opportunities. In *Proc. of the IEEE International Conference on Service-Oriented Computing and Applications (SOCA'14)*, pages 230–234, Matsue, Japan, 2014. IEEE.

[91] K. Zhao and L. Ge. a survey on the internet of things security. In *Proc. of the International Conference on Computational Intelligence and sSecurity (CISIS'13)*, pages 663–667, Leshan, China, 2013. IEEE.