Using Blockchain in a Reputation-Based Model for Grouping Agents in the Internet of Things

Giancarlo Fortino, Fabrizio Messina, Domenico Rosaci, and Giuseppe M. L. Sarné

Abstract—**The "Internet of Things" (IoT) makes attractive services available to smart objects and humans. To aim this, IoT devices need high sensing, reasoning, and real-time acting capabilities that can be also obtained by promoting adaptive forms of cooperation machine-to-machine among smart objects. The convergence of IoT and multiagent systems also relies on the association between software agents with IoT devices for exploiting their social attitude of interacting and cooperating for services. However, the choice of reliable partners for cooperation can be very difficult when IoT devices migrate across different environments, where the most part of their members will be unreferenced with respect to their trustworthiness. It is well known that agents reputation can be a viable aspect to consider to form social groups; therefore, a possible solution to this problem is to form groups of agents in each IoT environment, based on their social capabilities. In this respect, the first contribution of this paper is represented by a reputation model focused on building the reputation capital of each agent. Second, an algorithm capable to form groups of agents in IoT environments on the basis of their reputation capital was designed. Finally, since in this contest, it is important to spread reliable and certified information about the device/agent reputation in a distributed environment, the third contribution is represented by the adoption of the blockchain technology to certify the reputation capital. Some experiments we have performed show that the model is capable to detect almost all the misleading agents if their percentage is under a high enough threshold, and that good results in term of group composition are obtained. Moreover, the simulations show that, by adopting our model, malicious devices always pay for services significantly more than honest ones. We argue that the individual reputation capital of devices and, consequently, the overall reputation capital of the IoT community can take benefit from the adoption of the proposed approach.**

Index Terms—**Blockchain, group formation, Internet of Things (IoT) device, multiagent system, reputation capital.**

Manuscript received February 26, 2019; revised April 6, 2019; accepted May 16, 2019. This work was supported in part by the "COGITO - Dynamic and Cognitive Systems to allow Buildings to Learn and Adapt" Project, funded by the Italian Government under Grant ARS01_00836, in part by the "Piano della Ricerca 2016/2018 – Linea di intervento 2" (Universitá degli Studi di Catania), and in part by the **NeCS** Laboratory (DICEAM, University Mediterranea of Reggio Calabria). Review of this manuscript was arranged by Department Editor K.-K. R. Choo. *(Corresponding author: Giuseppe Maria Luigi Sarné.)*

G. Fortino is with the Department of Informatics, Modeling, Electronics, University of Calabria, 87036 Rende (CS), Italy (e-mail: giancarlo.fortino@unical.it).

F. Messina is with the Department of Mathematics and Informatics, University of Catania, 95126 Catania (CT), Italy (e-mail: messina@dmi.unict.it).

D. Rosaci is with the Department of Information, Infrastructures and Sustainable Energy Engineering, University Mediterranea, 89122 Reggio Calabria (RC), Italy (e-mail: domenico.rosaci@unirc.it).

G. M. L. Sarné is with the Department of Civil, Energy, Environment and Materials Engineering, University Mediterranea, 89122 Reggio Calabria (RC), Italy (e-mail: sarne@unirc.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEM.2019.2918162

I. Introduction

THE "Internet of Things" (IoT) [1] poses technical, social, and economical challenges for transforming our world by realizing diffuse context-aware smart environments around us [2], [3]. Since an emerging trend is represented by the complex requirements, in terms of interactive tasks, to IoT smart objects (i.e., IoT devices), promoting adaptive forms of cooperation [4], [5] among smart objects can make available services to other IoT devices [6] in order to satisfy such requirements. To this purpose, a number of IoT architectures and standards have been proposed [7]–[9] exploiting a wide range of sensory, communication, networking, and information technologies [10]–[13]. An effective solution to promote social interactions among smart devices is that of adopting a multiagent systems, where each software agent is associated with a device that works on its behalf [14], [15].

The feature of an IoT device of moving across different, federated administrative domains as well as of interacting and cooperating with other devices is very attractive, but performing the choice of the most suitable "partners" [16] may be very hard within a similar context. Note that this task heavily influences the quality of the interactions occurring among the cooperating devices and, consequently, the resulting level of "satisfaction" that each device can receive from such interactions, e.g., when device interactions involve critical activities and/or imply significantly expensive resources and/or high monetary cost for obtaining a service.

Moreover, a software agent may have to perform the choice of its partner also in the case the reliable information about other agents is not available, e.g., when a device moves itself from an environment to another one. In these cases, the usual approach adopted in human communities of asking information to other trustworthy agents is generally unfeasible because the new environment and its members are often unknown and unreferenced.

As a result, a mobile IoT device (i.e., its associated software agent) has to solve the problem of choosing own reliable partners also in the absence of a suitable experience to perform good choices. A possible solution to such problems is that of forming social structures, as agent groups, to be formed on the basis of kind social properties occurring among the group members [17] so that to provide them with a reasonable mutual expectancy to carry out positive interactions. The larger the number of positive interactions occurring among the members of a group, the larger the effectiveness of that group or, in other words, the social capital of that group [18]. Differently, a not competitive context leads to increase in the social capital of the overall community [19].

Among the proposals investigated in real and virtual communities, taking into account the dynamics underlying formation, evolution, and roles of social groups [20], [21], a promising viewpoint recently introduced is that of forming groups based on high levels of trustworthiness among the group members [6]. Indeed, a high level of mutual trustworthiness is fundamental both for promoting cooperation in real and virtual communities and to form effective groups [22].

In this perspective, a crucial problem to face is *how the trustworthiness should be represented in an IoT network,* where several disjointed environments populated by a large number of agents are federated together.

As highlighted above, we observe that in the presence of a large IoT environment/environments there is a certain probability that an agent has to interact with interlocutors whose reliability is still unknown. As a consequence, it may be very common that an agent selects a partner by relying on the *global reputation* of the agents (i.e., devices) in the whole agent community [23], because information about its past personal experiences is not available. However, in a distributed environment, the global reputation is a measure difficult to obtain and spread without to use some form of centralized repository. In particular, this measure should take into account any interaction (or at least the most recent ones) occurred between agents of the community, it should be certified with a high level of reliability and accessible in each federated environment composing our considered IoT scenario [24].

A. Our Contribution

In order to solve the problem of choosing a reliable provider in such a mobile, distributed context and, at the same time, avoiding the adoption of a central repository, we introduce the following ideas.

1) Modeling agent reputation using a sort of individual capital [called *reputation capital* (RC)] obtained by “summing” feedback received during the activities of the agents.
2) Adopting a blockchain protocol [25]–[27], as Ethereum [28], to spread, maintain, and certify, on the basis of cryptographic validation techniques, the RC of all the agents of the distributed IoT scenario.
3) Introducing a trust framework of federated IoT environments.
4) Supporting the formation of virtual groups of devices, which are affiliated to a group on the basis of their RC score.
5) Group formation is supported by a group formation algorithm, which is based on the trustworthiness information properly maintained in a blockchain.

We have implemented our framework and we have tested on a simulated scenario, containing both honest and misleading agents.

As discussed in the remaining of the paper, the analysis of the experimental results has clearly shown that the proposed framework is resilient to malicious agents, almost all detected in about five epochs, and some in their attacks if malicious agents are no more of a high enough threshold (e.g., $\sim 25\%$). Moreover, we have also tested the group formation algorithm, obtaining significant results in terms of group composition.

In other words, one of the most significant advantages of our approach is that of combining reputation information, group formation, IoT, and blockchain technologies, so that IoT devices migrating across different federated administrative domains can always rely on their RC for joining with groups active in their current environment.

B. Organization of This Paper

The remaining of the paper is organized as follows. Section II gives an overview on the related literature. Section III introduces the proposed framework scenario, whereas Section IV presents the reputation-blockchain mechanism. Section V describes the group formation algorithm. Section VI contains the results of our experimental results.

II. Related Work

A distributed system (DS) is exposed to a large number of potential threats for malicious and/or disliked behaviors than centralized environments [29]. Risks significantly increase when DSs are also open and competitive, as in our proposal. For such a reason, trust and reputation systems have been widely studied to support users’ (i.e., agents) activities. DSs can take benefit from cryptographic techniques [30] as well as trust and reputation systems [31], [32]. The former provides protection against outside attacks by safeguarding privacy and ensuring the counterparts authentication [33]. Differently, trust and reputation systems help in estimating the trustworthiness of potential partners, to limit the risk due to their possible unreliability [34], on the basis of information derived from direct experiences (reliability) and/or opinions of others (reputation), usually arranged in a single synthetic measure as, for instance, in [35]–[37].

A. Trust and Reputation Systems

The relevance of trustworthiness is witnessed by its adoption in almost every decision process and social interaction in human and virtual activities [38], [39] and, consequently, a large amount of studies adopting different viewpoints, as for instance in [40] and [41], and a wide variety of analysis, models, and architectures can be found in the literature. The accuracy of these esteems tightly depends on both number and quality of the information sources [35], the modality to aggregate and inferring trust in a local or global way [42], and the presence of a centralized or distributed context [43]. In particular, our proposal is unfavorable to a local approach and, as the most part of DSs, the relevance of direct experiences (i.e., reliability) loses of relevance given that for each member the most part the community will be not referenced.

To this regard, as stated in [44], any reputation system should satisfy some main properties. First, entities must be long lived so that past experiences give information about the expected future behaviors and, therefore, whitewashing strategies aimed to change identity for cleaning a bad reputation have to be hindered as more as possible. Second, decisions about new interactions should be driven by past experiences. Third, ratings about current interactions need to be gathered, also by planning incentives (or conversely, disincentives) to persuade participants to release their feedback that has to spread into the community. This latter task can represent a hard challenge in the presence of DSs.

To spread reputation scores in DSs [45], it is possible to realize distributed and synchronized repositories or to let this task to be realized by each participant during its interactions with third parties. In the first case, a unique model to compute reputation will be, generally, adopted, whereas in the other case, it is also possible that each member of the DS could adopt an individual metric by taking into account also the direct experiences together with the received reputation scores.

Trust and reputation scores are also useful in group formation processes involving real and virtual communities. To this aim, many proposals exploit trust or reputation to recommend to a group (member) of a community, which are the best members (groups) for the affiliation with; this problem is usually known as *group recommendation* (*affiliation* problems). Indeed, by considering trustworthiness information in forming groups, the result is more stable over time because, generally, in this case their members and groups themselves receive more benefits with respect to the use of other group formation strategies [46]. For instance, among the existing proposals, in [47], groups are formed in a peer-to-peer (P2P) system with a trust-based procedure taking into account the agent trustworthiness, and the benefits of these proposals are confirmed by the simulations. Breban and Vassileva [48] verified that a formation mechanism for long-term coalitions, of both customer and seller agents based on their trust relationships with other agents, is beneficial for both the systems and, in an exponential way, for the agents. Differently, when groups are formed only on the basis of how much the profiles of the potential group members are similar then it is not guaranteed that groups will be homogeneous over time (i.e., cohesive).

Also in the IoT world, trust and reputation criteria are becoming more and more important, and new trust and reputation models for IoT have been designed [49], [50]. At the same time, the choice of providing IoT devices communities of social structures can be assumed as an effective way to improve their performances.

For instance, in [51], interacting IoT devices mutually trust their counterparts and, with a *word of mouth* mechanism, spread their trust evaluations, in the form of recommendations, to the other devices, whereas Chen *et al.* presented in [52] a trust system able to follow the evolution of social relationships over time and adapting itself to the unavoidable trust fluctuations. Fortino *et al.* [6] studied the convergence among IoT, software agents, and cloud computing to form groups of agents (each one associated with an IoT device and living on the cloud) and designed an algorithm to form agent groups on the basis of information about reliability and reputation collected by the agents. Some simulations verified the efficiency and effectiveness of the algorithm confirming its potential advantages. Schooler *et al.* [53] described an architecture to create an eco-system for smart objects capable to reason about context and behaviors and/or to self-organize themselves into groups of like-minded smart objects, also by taking into account the trustworthiness of the potential members. To group IoT nodes into the most appropriate clusters based on their trust values in [54], a scalable trust management solution is proposed, where an algorithm provides to consider only correct trust values for an IoT service; to form trust-based clusters; and to permit trust-based intercluster migration of IoT nodes. Note that the IoT nodes can progressively gain or lose trust values as they interact with other nodes of their clusters.

B. Blockchain

Finally, to realize secure management frameworks for distributed IoT environments, recent proposals adopt the blockchain technology [55], [56] for enabling an easier sharing of resources and services between IoT devices. From its introduction, in 2009 [57], the blockchain is grown in popularity. At a broad level, it provides a mechanism to warranty data integrity and trust unknown and anonymous entities, by means of a decentralized, distributed, open, and unchangeable ledger storing data (e.g., documents, contracts, monetary transactions, etc.) across a P2P network by using cryptographic technologies to identify source and sink of the data. In such a way, a secure replacement of third parties or centralized authorities is possible, which may be disrupted, compromised, or hacked over time [58], [59].

Briefly, a blockchain is a chain of data blocks chronologically ordered and replicated on a number of distributed P2P hosts. Each data stored into the ledger are encrypted, validated, and verified by a distributed consensus[1] from some nodes. Once a block is validated and verified by consensus, it becomes permanent, immutable, and accessible [60]. Each block is formed by a header (storing information such as identifier, timestamp, number of stored transactions, size of the block, and the hash of the previous block in the chain) and the data stored into the block. Since all the ledgers local copies are "synchronized," the blockchain cannot be controlled, tampered, or deleted in an easy way or by a single actor.

In this context, the advent of Ethereum [28] represents the starting point for a huge amount of applications to the blockchain technology, as for instance smart-contracts (i.e., "a computerized transaction protocol that executes the terms of a contract" [61] or, in other words, programs loaded and executed on the blockchain) to be implemented and not only uniquely for supporting cryptocurrencies. From a technical viewpoint, smart contracts can be compared to software agents [62], which autonomously realize programmed transactions. For instance, the Ethereum [28] platform makes available a Touring-complete language programming to allow, in a relatively easy way, to develop code for smart-contracts. However, Ethereum has been the first blockchain platform supporting smart-contracts, but currently other similar platforms are active among which, for instance, Hyperledger [63], Ripple [64], Stellar [65], and Tendermint [66].

The choice of adopting a specific blockchain platform tightly depends on the required computational complexity placed on

[1] Typically, a consensus mechanism includes, first, the transaction endorsement, second, the ordering process, and third, the validation and commitment process.

the ledger by its consensus protocol. To the aim, different consensus protocol have been designed [67], which mainly differ among them in terms of robustness and computational complexity (often tightly connected). Therefore, the use of a consensus protocol in place of another also implies different costs for adding a new block on a blockchain, and this could heavily impact on the admissible IoT contexts. Indeed, trivially, expansive consensus protocols, such as the *Proof of Work* (PoW) first adopted by the Bitcoin currency [57], requires to solve a computationally expansive hashing puzzle for making valid and adding a new block and, often, its adoption in an IoT context requires the support of other technologies, such as cloud computing. The long latency, low scalability, and poor environmental friendly of PoW have led to the development of other consensus mechanisms [68] (also at detriment of security). Among the alternatives to the PoW, the *Proof of Stake* protocols are the more known; they are based on the idea that there is something at stake and include different strategies, some which are those described in [69]–[71].

A number of proposals relying on blockchain and smart-contract technologies, to validate transactions by facilitating and supporting the autonomous workflows and services sharing occurring among IoT devices, are described in [72] by highlighting their benefits in terms of payments, trading, shipping, and supply-chain management. In distributed environments, also including IoT devices, Di Pietro *et al.* [73] proposed Trustchain, which enables blockchain-based trusted transactions in an environment characterized by scalability, openness, and Sybil-resistance by adopting a consensus protocol alternative to the PoW. Moreover, an increasing number of papers is implementing a blockchain for spreading trust and reputation scores without to require a trusted and powerful third party. In particular, Falcone and Castelfranchi [74] designed a system for spreading trust across different IoT domains (here called *Islands of Trust*) by using two blockchains, the former is a private credit-based blockchain built on a (primitive) concept of reputation, while the second is used for payments. Even though some points of contact exist between our proposal and that described in [74], there are some important differences, among which the main are the following.

1) Our system introduces an effective reputation system based on the novel concept of RC and expressed by a unique synthetic score.
2) We adopt a single blockchain and any further significant complexity due to blockchain protocols is introduced into the framework.
3) We exploit smart contracts, which can include also negotiation tasks fixing the nature of the contractual relationships while, on the other side, currently this phase is not considered.

III. Proposed IoT Framework

In this section, we introduce the IoT framework—represented in Fig. 1—on which our approach is based. Our scenario is represented by a large number of heterogeneous IoT devices, which are assisted by software agents. Cooperation is another key aspect of the reference scenario: agents, on behalf of devices, can interact for any task[2] on the basis of smart contracts.[3]

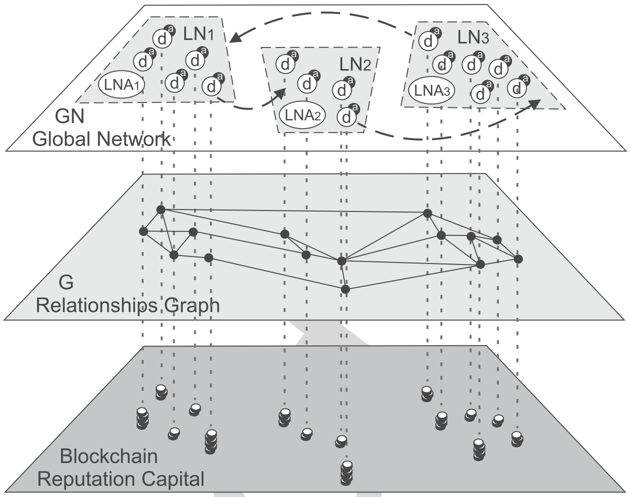

Fig. 1. Agent-based framework.

Let us denote with D, the set of devices, and with A, the set of software agents, each one associated with a unique, personal device.

Let GN be the *global network* composed by a number n (with $n > 1$) of federated *local networks* (LN)s. Let us also assume that a blockchain is associated with it. We will explain, with more detail, this aspect in Section IV.

A trusted and equipped agent called *local network administrator* (LNA) will correspond to each LN and provides all the agents temporarily living on its administrated domain LN $\in$ GN of some basic services. For instance, an LNA will assign an identifier (Id), unique into the GN, to each agent (i.e., device) entering for the first time on the GN, and it is capable of maintaining an updated registry of all the agents currently hosted in its LN. From hereafter, the terms device and agents will be used in an interchangeable way.

We represent the set of agents A registered on GN and their relationships by means of a graph $G = \langle N, L \rangle$, where N represents the set of nodes belonging to G and each node of N is associated with a unique agent $a \in A$ (i.e., device $d \in D$), while L is the set of oriented links, where each link of L represents a relationship occurring between two agents (see intermediate layer of Fig. 1). Moreover, since there are a great number of agents on NG, the graph G will be sparse.

Within the single LN, each agent can form groups on the basis of its RC witnessed by the blockchain (see Section IV). We also remark that agents can freely move from an LN to another LN and are free for joining with a group active on their current LN, on the basis of their convenience. At the same time, we assume that each group of an LN is coordinated by the respective

[2] For instance, tasks can refer to the extraction knowledge from informative sources, the exchange of knowledge, and etc.

[3] Smart contracts are self-executing contracts, within the terms of an agreement occurring between two actors, directly written into lines of code.

LNA that, to maximize the effectiveness of the group itself, can also contact other devices (i.e., agents) to join with or remove from the group of agents having an inadequate RC. Finally, let us define the jth group g formed inside the ith LN $\in$ GN as $g_j^{\mathrm{LN}_i}$.

IV. RC and Blockchain

This section presents the RC and the support provided by the blockchain, which makes the RC scores trusted over the GN. Remember that we assumed that each IoT device acts as a prosumer by following its necessities and convenience. In particular, note that the solvency (i.e., reliability) of an IoT device when it acts as a consumer (i.e., it has to pay a required service to another IoT device) is guaranteed by the blockchain, while when it acts as a provider, its capability in providing high-quality services is witnessed by its RC score described below in detail.

A. *Reputation Capital*

The RC is represented by a numerical score obtained by the historical "behaviors" of the devices (i.e., agent) carried out during its past interactions with the other devices belonging to GN when it acts as a provider of *qualified interactions* (QIs), see below. In particular, the proposed RC model is able to accomplish the following.

1) Take into account the recent agent QIs history in terms of a number of received feedback scores in a fixed horizon (h) and by their *Relevance* (R), see (3), considered in a decreasing manner based on their freshness.
2) Hinder alternate behaviors (based on the fact that small percentages of negative interactions do not damage the RC in a significant way) by assuming that a negative feedback (NF) decreases reputation more than a positive one increases it.
3) Avoid collusive behaviors of two (or more) agents, aimed to mutually and quickly increasing their RCs, by assuming that within the given horizon each agent can contribute only one time to the RC of another agent and only with its more recent feedback.
4) Limit the impact of "habitual" complainers by weighting each their NF by their credibility [75].

The RC of an agent is assumed to be a real positive number where "high" values indicate a "good" reputation. Each new agent receives an initial RC, which should not penalize too much a newcomer [76] and, at the same time, it represents a countermeasure over whitewashing strategies of malicious agents aimed to return into the system for receiving a new initial favorable RC [77].

More in detail, after that a service s with a cost p is required for an agent a_i (i.e., provider) by another agent a_j (i.e., consumer), this later gives a feedback $f_{ji} \in [0, 1] \in \mathtt{R}$ representing its appreciation for the service s. If the interaction involves a QI, then the RC$_i$ of the agent a_i (similarly for the agent a_j when it acts as a provider) will be updated. To this aim, we consider an interaction as qualified when R and f assume the following values:

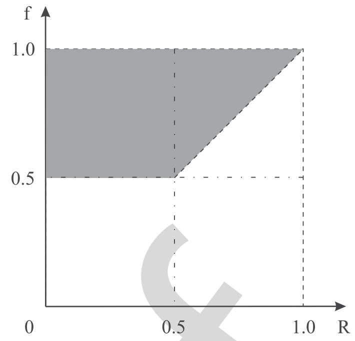

Fig. 2. Graphical representation of QIs (white area).

$$\begin{cases} f < 0.5, \ R \in [0, 1] \\ f \geq 0.5 \ \wedge \ R \geq f \end{cases}. \tag{1}$$

The ratio behind the definition of QIs [cases in (1)] consists in considering the contribution of the interactions for which a low feedback has been received for any value of R (relevance). On the other hand, when the feedback assumes a value equal or greater than 0.5, this contribution is considered as part of a qualified interaction, if and only if the value of the relevance is greater than the feedback itself.

In other words, the interactions that we have considered as unqualified can be also exploited in the context of malicious and alternate behaviors and, therefore, they have been excluded from the RC computation so that they, first, do not provide any positive or negative contribution to the RC in order to avoid the damaging correct agents and, second, do not provide advantages to dishonest agents. Alternatively, those interactions that we named as qualified are those that mainly characterize correct and incorrect behaviors and, consequently, have been considered in computing the RC.

The white area in Fig. 2 represents the combination of R and f for which there are QIs.

Therefore, for the provider agent a_i, with respect to a horizon consisting of the later h, QIs are performed by different agents (i.e., valid QIs), and then, its RC$_i$ will be updated as follows:

$$\mathrm{RC}_i = \sum_{n=1}^{h} w_n \cdot C_{j,n} \cdot R_{ji,n}^{(n^{-1})} \cdot f_{ji,n}$$

where

w (Freshness weight): This parameter weights the QIs so that the more recent is a QI, the more it contributes to the RC;

C (Credibility): It limits the effects of those agents systematically releasing NF (e.g., $f < 0.5$) for gaining unfair advantages; it is computed as the complement to 1 of the ratio between the number of NF released by an agent with respect to the overall number of its interactions (NT). More in detail, this parameter is computed as

$$C_j = \begin{cases} 1 & \frac{\mathrm{NF}}{\mathrm{NT}} \leq 0.5 \\ 1 - \dfrac{\mathrm{NF}}{\mathrm{NT}} & \frac{\mathrm{NF}}{\mathrm{NT}} > 0.5 \end{cases}; \tag{2}$$

R (Relevance): It takes into account the relevance of the interaction in terms of price, and it is computed as

$$R_{ji} = \begin{cases} \dfrac{p_s}{P} & \text{if } p_s < P \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

where p_s is the cost of the service s and P is a cost threshold for an interaction (set on the basis of the GN context) after which the relevance of the service is assumed as saturated (e.g., $R = 1$);

f (feedback): It is the appreciation of the consumer for s; it ranges in $[0, 1] \in \mathsf{R}$, where 1 (i.e., 0) represents the maximum (i.e., minimum) appreciation for s.

Finally, when noising activities are detected (e.g., a systemic activity addressed to induce the failure of a smart contract by making not more available a resource or by voluntarily interrupting the communication), the guilty agent/agents will be penalized by reducing its/their RC as follows:

$$\mathrm{RC}^{\mathrm{new}} = \alpha \cdot \left(1 - \frac{\mathrm{NA}}{\mathrm{NT} + 1}\right) \cdot \mathrm{RC}^{\mathrm{old}}$$

where NA is the number of aborted interactions, NT is the overall number of interactions of an agent, and α is a coefficient less than 1 that will be set on the basis of the adopted policies which, in turn, depends on the IoT context. In a similar way, an agent that does not release the feedback about its counterpart will be penalized on the basis of how frequently it happened.

B. RC Updating by Smart Contracts

Each time a service is provided into a LN, it is committed by means of a smart contract, running on the blockchain platform, which verifies and realizes all the contractual obligations. In this respect, also if our proposal is independent from a specific blockchain protocol, in this preliminary phase, we will refer to the well-known Ethereum platform for the advantages derived from both the availability of well-documented API and the opportunity of adopting its cryptocurrency (i.e., Ether) for payments inside the GN. Note that in the proposed framework payments occur to pay both services and the cost due to the management of the blockchain and, for the sake of simplicity, we assumed that such costs are included in the service.

In our platform a smart contract, behind the terms of the interaction occurring among the agents, has to include some steps devoted to updating both the RC of the provider and a list of data. The updated RC is an asset that will be stored on the blockchain together with the following set of information consisting of the following.

1) The identifiers of the device/agent who is referred the RC and that of the LN where it is currently affiliated to.
2) The number of all the interactions that the agent carried out with success in the past.
3) The number of all the interactions that the agent aborted.
4) The number of all the interactions for which the agent released a NF (i.e., $f < 0.5$).
5) The number of all the interactions for which the agent does not released a feedback.
6) A list of the latest q transactions that are forming the current RC score of the agent, each of the q rows of this list is a tupla consisting of the following.
    a) The identifier of the counterpart agent.
    b) The identifier of the LN, where the counterpart agent is currently affiliated to.
    c) The date and the cost of the transaction.
    d) The score of the feedback released for that interaction.

For reducing the latency time of the proposed platform, each LNA manages a private list of all the agent currently affiliated with its LN, ordered on the basis of their identifier, and storing the current (e.g., updated) value of the RC, a timestamp and the collocation on the blockchain (e.g., the block) where all the information of interest is stored. This private list will be accessible, obviously to the LNA, and at all those actors managing the blockchain. Moreover, as a receipt each agent involved in an interaction will receive a simple certificate, signed with the privet key of the ledger, storing the same information. Note that the size of this certificate is very little and does not provide computational or communication overload affecting the system performances.

V. Group Formation Algorithm

In this section, we describe the group formation algorithm executed in each local network (i.e., LN) by its administrator (i.e., LNA) having the goal of grouping devices (i.e., agents) on the basis of their RC scores. In order to highlight the contribution of both RC and groups in promoting the growth of individual and global RC within a LN (i.e., GN), in the following, the agents will be considered as homogeneous with respect to their interests and preferences (see Section II). Moreover, we will assume that if the consumer agent of a service s belongs to the same group of its provider, then the consumer has not to pay the price of s to its provider.

For each LN, its LNA sets the maximum number of groups admitted in that domain and, in an increasing way, their RC affiliation thresholds representing the RC score required for an agent for its affiliation with a group. Each agent will be affiliated with the LN group "best" fitting its RC (see Algorithm 1), and periodically its affiliation will be verified by the LNA on the basis of its RC score. As a consequence, each agent could be moved to another group that better fits its RC or, finally, the agent will be removed from every group because its RC becomes too low to join with any group active into the LN. This procedure, executed by the LNA of every federated LN, is represented by the pseudocode listed in Algorithm 1; the symbols used in its description are listed in Table I.

More in detail, let $A_m \subset A$ be the set of the agents currently affiliated with the federated LN_m; G_m be the set of the groups active on LN_m; and g_k^m be the kth group belonging to G_m (i.e., $G_m = \bigcup g^m$). Let LNA_m be the agent administrator of LN_m, which manages the two datasets DA_m and DG_m, that, respectively, store, first, identifier (ID_i), reputation capital RC_i, and the timestamp (τ_i) of the last RC check of each agent a_i currently running on LN_m; and, second, the data of each group

Algorithm 1: The Procedure Executed By Each LNA.

```
Input: $DA_m, DG_m, \phi_m, MxG_m$;
 1: for all $a_i \in LN_m$ do
 2:     if $\tau_i \geq \phi_m$ then
 3:         ask to the blockchain for the updated value of $RC$ about
 4:         $a_i$ and then update $DA_m$
 5:     end if
 6: end for
 7: for all $g_k \in G_m$ do
 8:     if$(rho_k \geq \phi_m)$ then
 9:         for all $a_i \in g_k$ do
10:             if$(RC_i < \Gamma_k) \parallel (RC_i \geq \Gamma_{k+1})$ then
11:                 Assign $(a_i, Gx_m, DA_m, DG_m)$
12:             end if
13:         end for
14:     end if
15: end for
16: for all $a_i \in LN_m$ requiring to be affiliated with a group do
17:     if $RC_i \geq \Gamma_1$ then Assign $(a_i, GxM_m, DA_m, DG_m)$
18:     else
19:         reject the request of $a_i$ and sends to the agent a message
20:     end if
21: end for
```

active on its LN$_m$, ordered on the basis of the admittance RC threshold values (e.g., Γ_k), and for each group $g_m \in G_m$ also the IDs of the agents belonging to that group, the admittance RC threshold (e.g., Γ_k) that an agent must have to be admitted into a group (e.g., g_k), the collective RC$_m$ for that group, and the timestamp (ρ_m) of the last RC check. Moreover, a LNA will set the maximum number of groups (MxG_m) active on LN and a time threshold (ϕ_m) representing the time interval occurring between two consecutive agent RC checks.

For the mth local network LN$_m \subset$ GN, the Algorithm 1 is periodically executed by its administrator LNA$_m$ after a time ϕ_m from its last execution or when an agent requires to LNA its affiliation with a group active on LN$_m$. By lines 1–5, the LNA$_M$ verifies on the blockchain the RC of all the LN$_m$ agents after a time ϕ_m is elapsed and then updates their RC scores on the dataset DA$_m$. In lines 7–15, for each group active on LN$_m$, periodically after that a time ϕ_m is elapsed, LNA$_m$ checks if the group members still satisfy the affiliation requisites of their belonging groups (i.e., the RC threshold Γ), otherwise LNA$_m$ calls the function *Assign( )* (see Algorithm 2). Finally, the last step of the procedure is executed when LNA$_m$ receives the agent request to be affiliated with a group (lines 16–21). Then the administrator LNA$_m$ verifies if the RC of the requester is greater or equal to the lower threshold (i.e., Γ_1); if this result is positive then the function *Assign( )* is called, otherwise the agent request is rejected and an appropriate message is sent to the requester.

The function *Assign( )* is represented in Algorithm 2 and receives an agent, the maximum number of groups that can be active in LN$_m$, and the two datasets DA$_m$ and DG$_m$. More in detail, for each agent belonging to LN$_m$, first, (see line 2) the function *Assign ( )* checks their RC admittance threshold (i.e., Γ). If RC$_i$ is lower than the threshold Γ_1 of the last group (that is the lower Γ in G_m), the function *Remove* ( ) is called to remove a_i from every group in G_m as long as its RC$_i$ will not be adequate to require a new affiliation with a group. Otherwise, lines 4–7 realize the assignation of a_i to the group g_k best fitting with its RC$_i$.

Algorithm 2: The Function *Assign* (a_i, MxG_m, DA_m, DG_m).

```
1: for all $a_i \in LN_m$ do
2:     if $RC_i < \Gamma_1$ then Remove $(a_i, DA_m, DG_m)$
3:     else
4:         for all $g_k \in G_m$ do
5:             if $RC_i \geq \Gamma_k$ then assign $a_i$ to the group $g_k$
6:             end if
7:         end for
8:     end if
9: end for
```

VI. Experiments

This section presents the results of a number of simulations performed to test the proposed framework and carried out by varying both the horizon (h) and the percentage of malicious actors. In particular, the simulations were aimed to verify the following.

1) The ability of identifying the malicious actors in the presence of different and concomitant typologies of attacks carried out also by adopting different strategies (see Section IV-A).
2) The distribution of devices (i.e., agents) among the different groups (notice that devices act as prosumers).
3) The growth of the RC with the number of interactions performed (remember that the RC of a device is referred only to its provider activity because consumers are trusted by the blockchain).
4) The costs sustained by devices to purchase services from providers.

The parameter settings and the obtained results will be presented and discussed below.

A. *Parameter Setting*

Simulations were performed with respect to only one federated LN $\in$ GN, where a sequence of interactions were carried out by IoT devices, each one associated with a software agent, among which a percentage of cheater devices (i.e., agents) performing collusive, noising, complainer, and different modalities of alternate behaviors were present. The setting of the main parameters adopted in these simulations was as follows.

1) A LN population of 10^3 IoT devices/agents.
2) Each interaction involved two devices (one acting as a consumer and the other as a provider of a service) chosen in a random way. Interactions were arranged in epochs

TABLE I
TABLE OF THE MAIN SYMBOLS

Symbol	Description
GN	the Global Network joining more Local Networks LNs
LN	a Local Network, with $LN \subset GN$
LNA	the Network Administrator agent of a local network
DA	dataset of agents affiliated with a local network
DG	dataset of groups active on LN_m
ϕ	time threshold set by a local network administrator
MxG	max. number of groups active on a given local network
G	set of active groups on a given local network
g	an group which is active on a given local network, with $g \subseteq G_m$
RC	reputation capital for a given group
$\mathit{\Gamma}$	RC required to be affiliated with a given group
A	set of agents living on the Global Network
A_m	set of agents affiliated with LM_m, with $A_m \subset A$
a_i	i-th agent affiliated with LM_m, with $a_i \in A_m \subset A$
ID_i	identifier of the agent a_i
RC_i	reputation capital of the agent a_i
τ	timestamp of the last computation of RC_i
ρ_k	timestamp of last computation of the requisite for belonging to a group g

and, in turn, each epoch was formed from 10^3 interactions, so that each one of the 10^3 agents acted as provider one time for epoch in average.

3) Simulations were carried out for 10^3 epochs, although results came "stable" in a maximum of about 10 epochs.
4) The initial RC score assigned to each device was set to 1.0 (a value taking into account different issues).
5) The cost p_s of a service s was randomly assigned in the range $1/cent \div 1.5\$$, while the cost threshold P of a service was set to 1$.
6) The *horizon* (h) adopted for the simulations varied from $h = 4$ to $h = 10$ with step 2.
7) Based on the different strategies adopted by honest and malicious devices, QIs occurred with rates of $1 : h$, $1 : h/2$, $1 : 1$ and in a random way.
8) We assumed a percentage of cheaters in the LN varying from 5% to 25% of the overall device population with a step of the 5%.
9) As previously described, we considered the following four types of *malicious behaviors*.
   a) *Alternate*, where low-value interactions are correctly closed in order to gain RC for cheating on high-value interactions; this activity was carried out with different (cheating:honest) ratios, namely $1 : h$, $1 : h/2$, $1 : 1$ and in a random way.
   b) *Collusive*, where two or more devices repetitively interact for mutually increasing their RC.
   c) *Complainer*, where NF are released to the counterparts in a systemic way in order to gain undue advantages; this activity was simulated with a rate varying from $1 : h$ to $1 : 1$.
   d) *Noising*, simulating aborting interactions with a frequency of 1:100. Note that aborting transaction also included the effects due to the presence of blockchain in witnessing the unreliability of a consumer device, in this case the interactions did not take place and, therefore, any RC penalization occurred.
10) The number of groups was set to 3 with affiliation RC thresholds, respectively, set to 2.5, 4.5, and 6.0.

B. Results

In the following are described some significant results obtained from the experimental campaign that we carried out by simulating the proposed framework. These results confirmed our expectations about the advantages derived from the adoption of the RC in a competitive context, in synergy with a group formation algorithm and the adoption of a blockchain protocol.

1) Malicious Identification: In the first experiment, we analyzed the accuracy of our framework by measuring the percentage of malicious devices/agents correctly recognized (on the basis of their RC) as the number of cheaters and the horizon increased. We assumed an initial RC $= 1.0$ as an initial condition

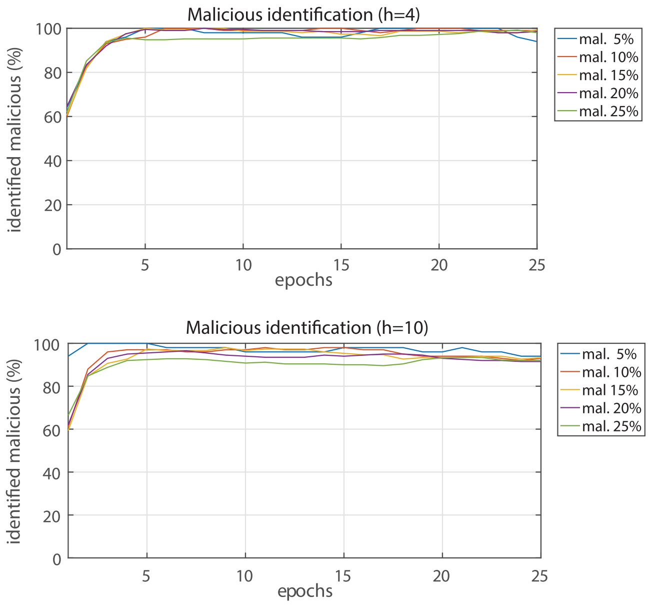

Fig. 3. Malicious identification for $h = 4$ and $h = 10$.

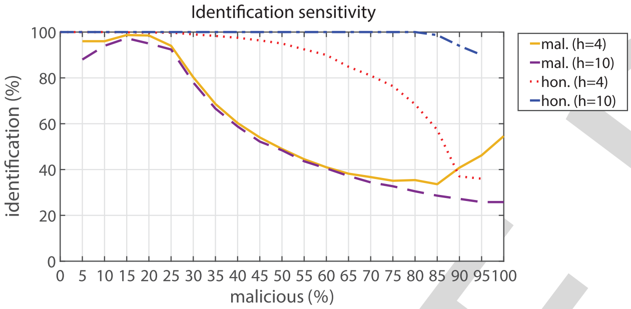

Fig. 4. Sensitivity for $h = 4$ and $h = 10$ in identifying honest and malicious devices when the population composition vary.

and represented here the results obtained for $h = 4$ and $h = 10$, both for cheaters percentages varying from 5% to 25% with step 5, see Fig. 3 as epochs increase. Note that malicious devices are recognized only on the basis of their RC with respect to the RC initially assigned for default to each device. Consider that even though the initial RC is a low value, after only 5 epochs, the percentage of recognized malicious varied from the 100% ($h = 4$, malicious = 5%) to 96% ($h = 10$, malicious = 25%). Remember also that all the malicious attacks acted during all the simulations.

Summarizing, this experiment has shown that the accuracy of the RC model is quite high for the considered scenario (take into account that the performance of the model ia also better in recognizing honest actors).

Furthermore, in order to know the sensitivity of our model (at the 25th epoch), the percentages of honest and malicious agents recognized for $h = 4$ and $h = 10$ when the percentage of malicious varies from 0% to 100% of the overall population are presented in Fig. 4. Results highlight that when the percentage of malicious is greater than 25% then the ability of the RC, in the adopted configuration, to recognize the nature (e.g., honest or malicious) of the devices, decreases. In particular, we can observe that, while the percentage of malicious agents correctly recognized ($h = 4$ and $h = 10$) starts to decrease just when the real percentage of malicious is equal to 25%, the percentage of honest agents ($h = 4$) correctly identified starts to decrease when the percentage of malicious reach the value of 50%, while the percentage of honest agents ($h = 10$) correctly identified slightly decreases only when the percentage of malicious reach a value around the 95%. From this last analysis, we can state that a limitation of the RC model is not being able to recognize malicious agents when the percentage of them with respect to the entire population is very high ($\geq$25%). From the other hand, the RC is still able to recognize honest agents even when the percentage of malicious agents with respect to the entire population is very high ($\geq$25%).

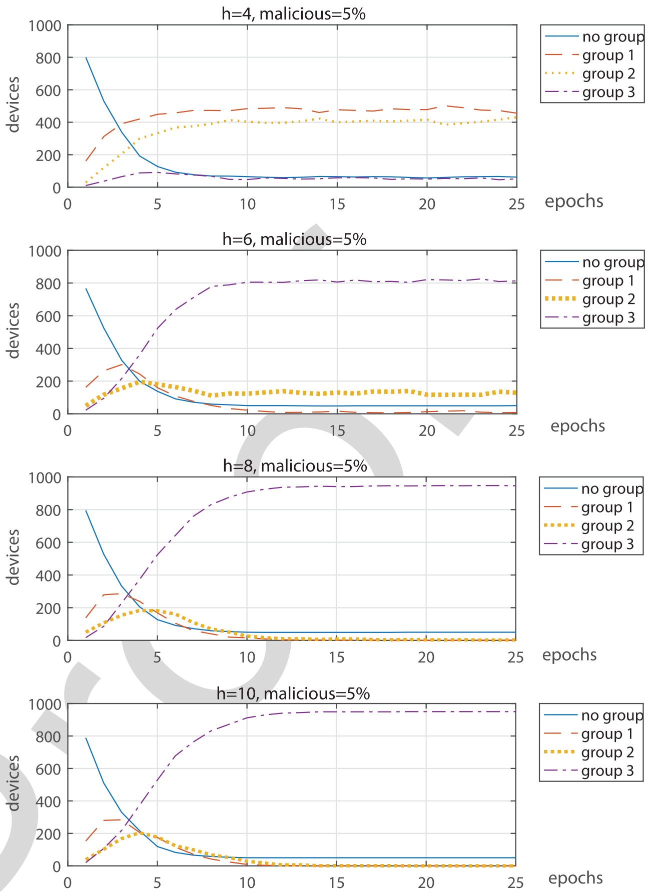

Fig. 5. Group affiliation for different *h* in the presence of the 5% of malicious devices.

2) Group Affiliation: The second experiment investigated on the affiliation of the devices to the groups active on LN. In particular, in Figs. 5 and 6 are represented the distributions of the devices among the groups when malicious devices are the 5% and the 25% of the population and for the horizon thresholds $h = 4, 6, 8, 10$. Note that the distribution trend also depends on the adopted affiliation RC thresholds (specified in Section VI-A and valid for all the scenarios).

Results clearly highlighted three aspects. The first one is that almost all the malicious devices do not belong to any group

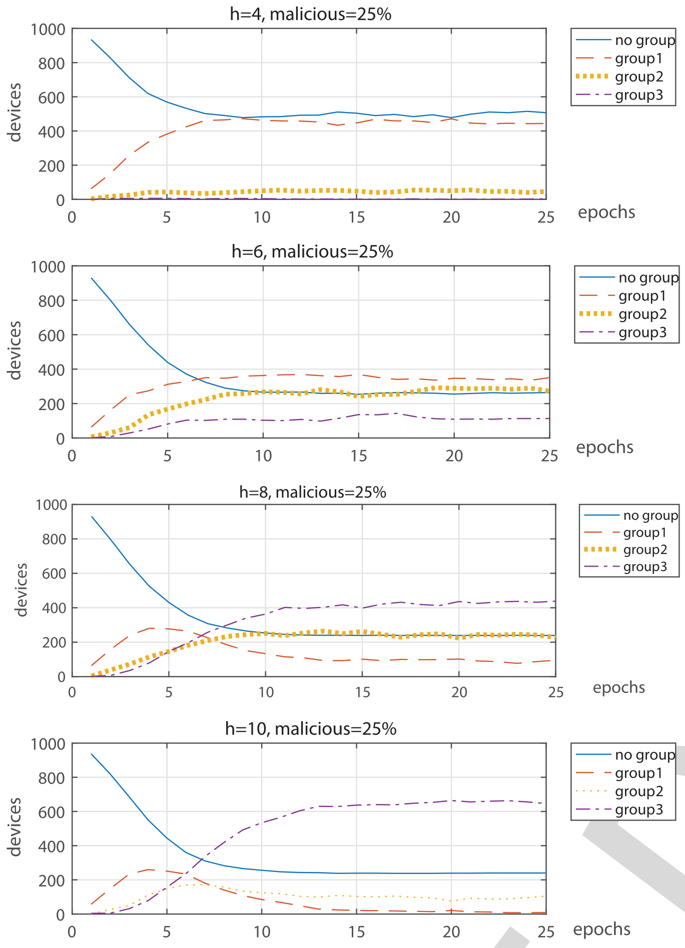

Fig. 6. Group affiliation for different *h* in the presence of the 25% of malicious devices.

because their RC score is unsuitable to perform a group affiliation and, in average, this result becomes stable in about $5 \div 10$ epochs. The other one aspect is referred to the honest devices that, first, with a low horizon rarely belong to the group 3 because this type of horizon does not allow them to acquire an RC score equal or greater than 6.0, that is, the affiliation threshold adopted for the group 3, conversely, second, with a large horizon, the most part of the honest devices is able to belong to the group 3. Also in this case, results are stable within the tenth epoch. Finally, the last observation is referred to the fact that, trivially, when malicious agents increase in percentage then it is needed to adopt a wider horizon.

3) RC: The third experiment analyzed the behavior of the *RC* when horizon and percentage of malicious agents vary. Also in this case, we present only the results obtained for $h = 4$ and $h = 10$ and for percentages of malicious devices varying in the range $5 \div 25\%$ with step 5, see Fig. 7. The dependence of the RC from horizon and numerosity of malicious devices is evident, as it also was deducible from the previous experiments.

4) Costs: Finally, the last simulations carried out (see Fig. 8) confirmed that malicious actors always pay for services more and more than honest devices. More in detail, the amount of the money paid from a malicious with respect to that paid from an honest device varied with a ratio of about $1 : 1.5$ to $1 : 4.2$ (both measured at the 25th epoch).

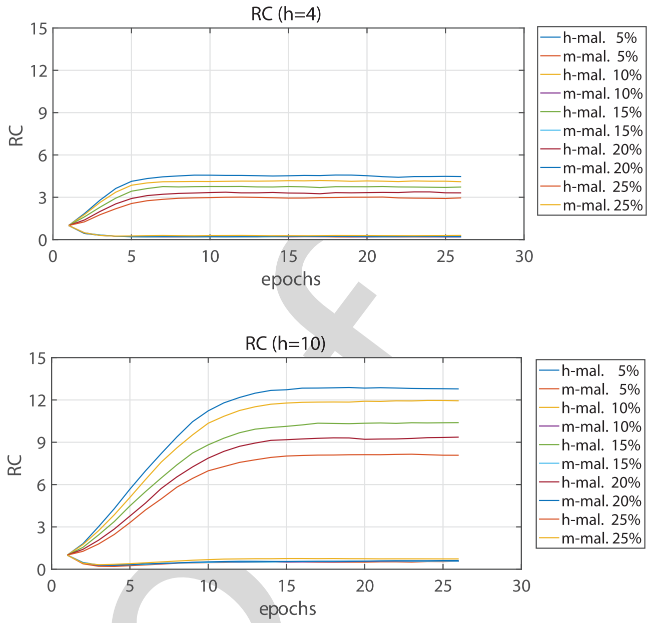

Fig. 7. RC for honest (h) and malicious (m) devices for $h = 4$ and $h = 10$ and different malicious percentages (mal).

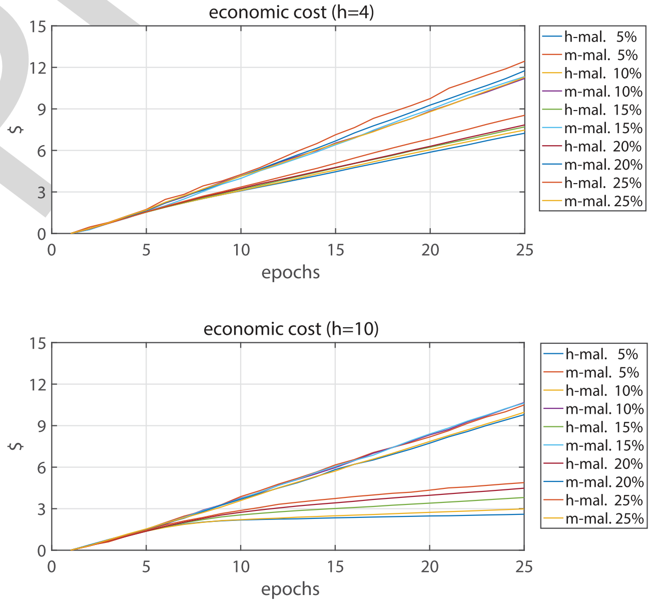

Fig. 8. Cost for services paid from honest (h) and malicious (m) devices for $h = 4$ and $h = 10$ and different malicious percentages (mal).

C. Discussion

The analysis of the experimental results has clearly shown that the proposed framework is resilient to malicious agents, almost all detected in about five epochs, and some their attacks if malicious agents are not more of a high enough threshold (e.g., $\sim$25%). Moreover, the experimental results regarding the RC scores, the dynamic underlying groups affiliation, and the costs paid for services by honest and malicious agents are positive and they reach a good stability in few epochs (e.g., $\sim$5–10). In particular, related results mainly depend on both the horizon threshold and the number of malicious devices.

In other words, experimental results highlighted that the synergy derived from the RC model, the group formation algorithm, and the blockchain allow the framework to correctly work.

VII. Conclusion

In this paper, we took into account a scenario comprising a wide IoT network federating several environments (local networks) of heterogeneous, smart IoT devices. In the considered scenario, devices can move across local networks, and they cooperate to reach their own goals with their peers.

Since cooperation implies the selection of reliable partners to cooperate with, the level of "satisfaction" that each device can receive from such interactions may vary in a significant way. This is a particularly sensible aspect, especially when device interactions involve critical and/or expensive (also in terms of resources) activities. In the scenario described previously, information as reputation can help in that choice, assuming that information about reputation is spread in a proper way.

To this purpose, we designed a framework where every IoT device was associated with a software agent capable to exploit its social attitudes to cooperate as well as to form complex agent social structures, as groups. To support cooperation, we introduced the RC, a numerical value, which is updated on the basis of the devices' feedback. To enable the dissemination of the reputation within the considered scenario, without the use of any centralized component, we exploit the support of the blockchain technology. Moreover, based on the RC values, each device can decide of asking the affiliation to a group of reliable agents with the expectancy of having satisfactory interactions and economic advantages.

In particular, we designed, first, a suitable RC model that implemented some countermeasures against collusive and malicious behaviors aimed to gain unfair RC and, second, a distributed group formation algorithm, driven by information about the RC of the agents (stored in a blockchain) that provided to divide the agents in groups on the basis of their RC score.

The experimental campaign of simulations carried out to verify efficiency and effectiveness of the proposed IoT framework highlighted that the synergy derived from the RC model, the group formation algorithm, and the blockchain allows the framework to work correctly.

In order to better validate the advantages introduced by our proposal, an experimental campaign in a real IoT scenario should be performed, and this will be the subject of our ongoing research.

References

[1] K. Ashton, "That' Internet of Things' thing," *RFID J.*, Jun. 2009.

[2] J. C. Augusto, V. Callaghan, D. Cook, A. Kameas, and I. Satoh, "Intelligent environments: A manifesto," *Human-Centric Comput. Inf. Sci.*, vol. 3, no. 1, p. 12, 2013.

[3] G. Fortino, A. Guerrieri, W. Russo, and C. Savaglio, "Middlewares for smart objects and smart environments: Overview and comparison," in *Internet of Things Based on Smart Objects*. New York, NY, USA: Springer, 2014, pp. 1–27.

[4] G. Kortuem, F. Kawsar, V. Sundramoorthy, and D. Fitton, "Smart objects as building blocks for the internet of things," *IEEE Internet Comput.*, vol. 14, no. 1, pp. 44–51, Jan./Feb. 2010.

[5] G. Fortino, W. Russo, C. Savaglio, W. Shen, and M. Zhou, "Agent-oriented cooperative smart objects: From IoT system design to implementation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 11, pp. 1939–1956, Nov. 2018, doi: 10.1109TSMC.2017.2780618, 2018.

[6] G. Fortino, F. Messina, D. Rosaci, and G. M. L. Sarné, "Using trust and local reputation for group formation in the cloud of things," *Future Gener. Comput. Syst.*, vol. 89, pp. 804–815, 2018.

[7] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. Mccann, and K. Leung, "A survey on the IETF protocol suite for the internet of things: Standards, challenges, and opportunities," *IEEE Wireless Commun.*, vol. 20, no. 6, pp. 91–98, Dec. 2013.

[8] J. Wan, B. Yin, D. Li, A. Celesti, F. Tao, and Q. Hua, "An ontology-based resource reconfiguration method for manufacturing cyber-physical systems," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 6, pp. 2537–2546, Dec. 2018.

[9] J. Wan, J. Li, Q. Hua, A. Celesti, and Z. Wang, "Intelligent equipment design assisted by cognitive internet of things and industrial big data," *Neural Comput. Appl.*, 2018.

[10] A. Celesti, M. Fazio, A. Galletta, L. Carnevale, J. Wan, and M. Villari, "An approach for the secure management of hybrid cloud–edge environments," *Future Gener. Comput. Syst.*, vol. 90, pp. 1–19, 2019.

[11] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[12] B. Chen, J. Wan, A. Celesti, D. Li, H. Abbas, and Q. Zhang, "Edge computing in IoT-based manufacturing," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 103–109, Sep. 2018.

[13] X. Zhang, K.-K. R. Choo, and N. L. Beebe, "How do i share my IoT forensic experience with the broader community? An automated knowledge sharing IoT forensic platform," *IEEE Internet Things J.*, to be published.

[14] D. Uckelmann, M. Harrison, and F. Michahelles, "An architectural approach towards the future Internet of Things," in *Architecting the Internet of Things*. New York, NY, USA: Springer, 2011, pp. 1–24.

[15] G. Fortino, R. Gravina, W. Russo, and C. Savaglio, "Modeling and simulating internet-of-things systems: A hybrid agent-oriented approach," *Comput. Sci. Eng.*, vol. 19, no. 5, pp. 68–76, 2017.

[16] H. Zhu and M. Zhou, "Role-based collaboration and its kernel mechanisms," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 36, no. 4, pp. 578–589, Jul. 2006.

[17] C.-M. Chiu, M.-H. Hsu, and E. T. Wang, "Understanding knowledge sharing in virtual communities: An integration of social capital and social cognitive theories," *Decis. Support Syst.*, vol. 42, no. 3, pp. 1872–1888, 2006.

[18] A. Blanchard and T. Horan, "Virtual communities and social capital," in *Knowledge and Social Capital*. Amsterdam, The Netherlands: Elsevier, 2000, pp. 159–178.

[19] M. Paldam, "Social capital: One or many? Definition and measurement," *J. Econ. Surv.*, vol. 14, no. 5, pp. 629–653, 2000.

[20] J. M. Leimeister, P. Sidiras, and H. Krcmar, "Success factors of virtual communities from the perspective of members and operators: An empirical study," in *Proc. 37th Annu. Hawaii Int. Conf. Syst. Sci.*, 2004, pp. 10.

[21] H. Zhu, "Avoiding conflicts by group role assignment," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 4, pp. 535–547, Apr. 2016.

[22] L. Fotia, F. Messina, D. Rosaci, and G. M. L. Sarné, "Using local trust for forming cohesive social structures in virtual communities," *Comput. J.*, vol. 60, no. 11, pp. 1717–1727, 2017.

[23] S. Ruohomaa, L. Kutvonen, and E. Koutrouli, "Reputation management survey," in *Proc. 2nd Int. Conf. Availability, Rel. Secur.*, 2007, pp. 103–111.

[24] P. De Meo, F. Messina, D. Rosaci, and G. M. L. Sarné, "Recommending users in social networks by integrating local and global reputation," in *Proc. Int. Conf. Internet Distrib. Comput. Syst.*, 2014, pp. 437–446.

[25] Z. Zheng, S. Xie, H.-N. Dai, and H. Wang, "Blockchain challenges and opportunities: A survey," *Work Pap.–2016*, 2016.

[26] M. Banerjee, J. Lee, and K.-K. R. Choo, "A blockchain future for internet of things security: A position paper," *Digit. Commun. Netw.*, vol. 4, no. 3, pp. 149–160, 2018.

[27] A. Buzachis, A. Celesti, A. Galletta, M. Fazio, and M. Villari, “A secure and dependable multi-agent autonomous intersection management (maaim) system leveraging blockchain facilities,” in *Proc. IEEE/ACM Int. Conf. Utility Cloud Comput. Companion*, 2018, pp. 226–231.

[28] 2018. [Online]. Available: https://www.ethereum.org

[29] R. Roman, J. Zhou, and J. Lopez, “On the features and challenges of security and privacy in distributed internet of things,” *Comput. Netw.*, vol. 57, no. 10, pp. 2266–2279, 2013.

[30] M. G. V. Kumar and U. Ragupathy, “A survey on current key issues and status in cryptography,” in *Proc. Int. Conf. Wireless Commun., Signal Process. Netw.*, 2016, pp. 205–210.

[31] J. Sabater-Mir and M. Paolucci, “On representation and aggregation of social evaluations in computational trust and reputation models,” *Int. J. Approx. Reasoning*, vol. 46, no. 3, pp. 458–483, 2007.

[32] Y. Han, “A survey of trust and reputation management systems in wireless communications,” *Proc. IEEE*, vol. 98, no. 10, pp. 291–298, Oct. 2010.

[33] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, “A survey on security and privacy issues in internet-of-things,” *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1250–1258, Oct. 2017.

[34] J. Fogel and E. Nehmad, “Internet social network communities: Risk taking, trust, and privacy concerns,” *Comput. Human Behav.*, vol. 25, no. 1, pp. 153–160, 2009.

[35] T. Huynh, N. Jennings, and N. Shadbolt, “An integrated trust and reputation model for open multi-agent system,” *Auton. Agent Multi Agent Syst.*, vol. 13, no. 2, pp. 119–154, 2006. Q9

[36] D. Rosaci, G. M. L. Sarnè, and S. Garruzzo, “Integrating trust measures in multiagent systems,” *Int. J. Intell. Syst.*, vol. 27, no. 1, pp. 1–15, 2012.

[37] L. Xiong and L. Liu, “Supporting reputation-based trust for peer-to-peer electronic communities,” *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 7, pp. 843–857, Jul. 2004.

[38] J. Heidemann, M. Klier, and F. Probst, “Online social networks: A survey of a global phenomenon,” *Comput. Netw.*, vol. 56, no. 18, pp. 3866–3878, 2012.

[39] J. Zhan and X. Fang, “Social computing: The state of the art,” *Int. J. Social Comput. Cyber-Phys. Syst.*, vol. 1, no. 1, pp. 1–12, 2011.

[40] T. French, N. Bessis, F. Xhafa, and C. Maple, “Towards a corporate governance trust agent scoring model for collaborative virtual organisations,” *Int. J. Grid Utility Comput.*, vol. 2, no. 2, pp. 98–108, 2011.

[41] S. Gächter, B. Herrmann, and C. Thöni, “Trust, voluntary cooperation, and socio-economic background: Survey and experimental evidence,” *J. Econ. Behav. Org.*, vol. 55, no. 4, pp. 505–531, 2004.

[42] Y. Kim and H. Song, “Strategies for predicting local trust based on trust propagation in social networks,” *Knowl.-Based Syst.*, vol. 24, no. 8, pp. 1360–1371, 2011.

[43] A. Jøsang, R. Ismail, and C. Boyd, “A survey of trust and reputation systems for online service provision,” *Decis. Support Syst.*, vol. 43, no. 2, pp. 618–644, 2007.

[44] P. Resnick, R. Zeckhauser, F. Friedman, and K. Kuwabara, “Reputation systems,” *Commun. ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[45] M. Momani and S. Challa, “Survey of trust models in different network domains,” arXiv:1010.0168, 2010.

[46] P. De Meo, E. Ferrara, D. Rosaci, and G. M. L. Sarnè, “Trust and compactness in social network groups,” *ACM Trans. Cybern.*, vol. 45, no. 2, pp. 205–2016, 2015.

[47] A. Aikebaier, T. Enokido, and M. Takizawa, “Trustworthy group making algorithm in distributed systems,” *Human-Centric Comput. Inf. Sci.*, vol. 1, no. 1, p. 6, 2011.

[48] S. Breban and J. Vassileva, “Using inter-agent trust relationships for efficient coalition formation,” in *Proc. Conf. Can. Soc. Comput. Stud. Intell.*, 2002, pp. 221–236.

[49] Z. Yan, P. Zhang, and A. V. Vasilakos, “A survey on trust management for internet of things,” *J. Netw. and Comput. Appl.*, vol. 42, pp. 120–134, 2014.

[50] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, “Security, privacy and trust in internet of things: The road ahead,” *Comput. Netw.*, vol. 76, pp. 146–164, 2015.

[51] F. Bao and I.-R. Chen, “Dynamic trust management for internet of things applications,” in *Proc. Int. Work Self-Aware Internet Things*, 2012, pp. 1–6.

[52] R. Chen, F. Bao, and J. Guo, “Trust-based service management for social internet of things systems,” *IEEE Trans. Depend. Sec. Comput.*, vol. 13, no. 6, pp. 684–696, Nov./ Dec. 2016.

[53] E. M. Schooler, D. Zage, J. Sedayao, H. Moustafa, A. Brown, and M. Ambrosin, “An architectural vision for a data-centric IoT: Rethinking things, trust and clouds,” in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 1717–1728.

[54] M. D. Alshehri, F. K. Hussain, and O. K. Hussain, “Clustering-driven intelligent trust management methodology for the internet of things (CITM-IoT),” *Mobile Netw. Appl.*, vol. 23, pp. 419–431, 2018.

[55] N. M. Kumara and P. K. Mallickb, “Blockchain technology for security issues and challenges in IoT,” *Procedia Comput. Sci.*, vol. 132, pp. 1815–1823, 2018.

[56] N. B. Truong, T.-W. Um, B. Zhou, and G. M. Lee, “Strengthening the blockchain-based internet of value with trust,” in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–7.

[57] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.

[58] M. A. Khan and K. Salah, “IoT security: Review, blockchain solutions, and open challenges,” *Future Gener. Comput. Syst.*, vol. 82, pp. 395–411, 2018.

[59] P. J. Taylor, T. Dargahi, A. Dehghantanha, R. M. Parizi, and K.-K. R. Choo, “A systematic literature review of blockchain cyber security,” *Digit. Commun. Netw.*, 2019.

[60] M. Pilkington, “11 blockchain technology: Principles and applications,” *Research Handbook on Digital Transformations*. Northampton, MA, USA: Edward Elgar, 2016, p. 225.

[61] N. Szabo, “Smart contracts,” to be published, 1994.

[62] V. Buterin *et al.*, “A next-generation smart contract and decentralized application platform,” *white paper*, 2014.

[63] 2018. [Online]. Available: https://www.hyperledger.org

[64] 2018. [Online]. Available: https://ripple.com/

[65] 2018. [Online]. Available: https://www.stellar.org

[66] 2018. [Online]. Available: https://tendermint.com

[67] N. Chalaemwongwan and W. Kurutach, “State of the art and challenges facing consensus protocols on blockchain,” in *Proc. Int. Conf. Inf. Netw.*, 2018, pp. 957–962.

[68] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An overview of blockchain technology: Architecture, consensus, and future trends,” in *Proc. IEEE Int. Congr. Big Data*, 2017, pp. 557–564.

[69] P. Vasin, Blackcoin’s proof-of-stake protocol v2. 2014. [Online]. Available: : https://blackcoin. co/blackcoin-pos-protocolv2-whitepaper. pdf

[70] D. Larimer, “Delegated proof-of-stake,” *Bitshare whitepaper*, 2014.

[71] V. Buterin and V. Griffith, “Casper the friendly finality gadget,” arXiv:1710.09437, 2017.

[72] K. Christidis and M. Devetsikiotis, “Blockchains and smart contracts for the internet of things,” *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[73] P. Otte, M. de Vos, and J. Pouwelse, “Trustchain: A sybil-resistant scalable blockchain,” *Future Gener. Comput. Syst.*, 2017.

[74] R. Di Pietro, X. Salleras, M. Signorini, and E. Waisbard, “A blockchain-based trust system for the internet of things,” in *Proc. 23nd ACM Symp. Access Control Models Technol.*, 2018, pp. 77–83.

[75] R. Falcone and C. Castelfranchi, “The socio-cognitive dynamics of trust: Does trust create trust?,” in *Proc. Workshop Deception, Fraud, Trust Agent Societies Held During Auton. Agents Conf.*, 2001, pp. 55–72.

[76] S. Ramchurn, D. Huynh, and N. Jennings, “Trust in multi-agent systems,” *Knowl. Eng. Rev.*, vol. 19, no. 1, pp. 1–25, 2004.

[77] G. Zacharia and P. Maes, “Trust management through reputation mechanisms,” *Appl. Artif. Intell.*, vol. 14, no. 9, pp. 881–907, 2000.

Giancarlo Fortino received the Ph.D. degree in computer engineering from the University of Calabria (Unical), Rende, Italy, in 1995 and 2000, respectively. Q10

He is a Full Professor of Computer Engineering with the Department of Informatics, Modeling, Electronics, and Systems, Unical. He is also an Adjunct Professor with the Wuhan University of Technology, Wuhan, China, and a Senior Research Fellow with the Italian National Research Council ICAR Institute, Rome, Italy. He is author of more than 350 papers in international journals, conferences, and books. He is a Co-Founder and CEO of SenSysCal S.r.l., a Unical spinoff focused on innovative IoT systems. His research interests include agent-based computing, wireless (body) sensor networks, and Internet of Things.

Dr. Fortino is currently member of the IEEE SMCS BoG and Chair of the IEEE SMCS Italian Chapter.

Fabrizio Messina received the Ph.D. degree in computer science from the Department of Mathematics and Informatics, University of Catania, Catania, Italy.

He is currently a Postdoctoral Researcher with the Department of Mathematics and Informatics, University of Catania. His research interest includes distributed systems, complex networks, simulation systems, and trust.

Domenico Rosaci received the Ph.D. degree in electronic engineering, in 1999.

He is an Associated Professor of Computer Science with the Department of Information, Infrastructures and Sustainable Energy Engineering, University Mediterranea of Reggio Calabria, Reggio Calabria, Italy. His research interests include distributed artificial intelligence, multiagent systems, and trust and reputation in social communities.

Dr. Rosaci is a member of a number of conference PCs and is an Associate Editor of the *Journal of Universal Computer Science* (Springer).

Giuseppe M. L. Sarné is an Assistant Professor of Computer Science with the Department of Civil, Energy, Environment and Materials Engineering, University Mediterranea of Reggio Calabria, Reggio Calabria, Italy. His main research interests include distributed artificial intelligence, multiagent systems, and trust and reputation systems.

Dr. Sarné is a member of a number of conference PCs and is an Associate Editor of *E-Commerce Research and Applications* (Elsevier) and of Big Data and Cognitive Computing (MDPI).