IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON RADIO FREQUENCY IDENTIFICATION AND SECURITY TECHNIQUES

Received July 4, 2018, accepted August 8, 2018, date of publication August 20, 2018, date of current version September 21, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2866185

Securing Internet-of-Things Systems Through Implicit and Explicit Reputation Models

BORJA BORDEL[1], RAMÓN ALCARRIA[2], DIEGO MARTÍN DE ANDRÉS,[1] AND ILSUN YOU[3], (Senior Member, IEEE)

[1]Department of Telematics Systems Engineering, Universidad Politécnica de Madrid, 28040 Madrid, Spain
[2]Department of Surveying and Cartography Engineering, Universidad Politécnica de Madrid, 28040 Madrid, Spain
[3]Department of Information Security Engineering, Soonchunhyang University, Seoul 31538, South Korea

Corresponding author: Ilsun You (ilsunu@gmail.com)

This work was supported in part by the Autonomous Region of Madrid through the MOSI-AGIL-CM Project, co-funded by EU Structural Funds FSE and FEDER, under Grant P2013/ICE-3019, and in part by the Soonchunhyang University Research Fund.

ABSTRACT Internet-of-Things (IoT) systems are usually composed of thousands of different components among hardware devices and different software modules. In order to address the design of these complex systems, different abstraction layers are usually defined. However, as these layers are isolated, high-level components always have uncertainty about the nature of the low-level components they relate with. In particular, as low-level component identities are not known by user applications, and current IoT systems are vulnerable to the injection of new components and to the modification of the behavior of existing ones (adequate security solutions at the network level for these problems have not been found yet), the reliability of the received data is generally compromised. In this context, new mechanisms are required to avoid the interactions or directly remove the malicious components relying on high-level information. This paper describes a statistical framework to discover IoT components with malicious behaviors, using a hybrid reputation model. On the one hand, an implicit reputation definition is employed, based on the observations made by a certain IoT component and other modules it relies on. On the other hand, an explicit reputation model considers a scheme of recommendations and negative grades. The proposed solution is evaluated in a simulation scenario by using the NS3 simulator, in order to perform an experimental validation.

INDEX TERMS Information systems, Internet-of-Things, security, reputation, uncertainty, pervasive sensing, knowledge discovery.

I. INTRODUCTION

Nowadays, Internet-of-Things (IoT) is one of the most interesting and promising technological fields, including from the very popular Cyber-Physical Systems [1] to the well-known smart cities [2]. In most of these areas, moreover, first pilots based on IoT technologies have been designed and deployed [3]. In general, real IoT deployments (such as, for example, Smart Santander or the smart buildings of the Korea Electronic Technology Institute -KETI-) are very complex and heterogeneous architectures where a large number of very different components (including processing devices, services, execution engines, communication middleware, etc.) are connected in an ad hoc way. In order to face and make feasible the implementation and management of these complex systems, different abstraction layers are defined: hardware-level, middleware, final users, etc. [4]. However, the definition of these abstraction layers also causes that components in various levels are isolated from the elements located in other tiers. For example, high-level applications are not usually enabled to provide Quality-of-Service in IoT systems, as they do not have enough information about the infrastructure [5].

Furthermore, as IoT solutions are not often standard nowadays, relevant information for the system operation cannot be adequately transferred among the different layers, because of incompatibilities among the selected technologies. The most critical example of this situation is the extreme difficulty of high-level applications to know the identity of the low-level component they interact with. In fact, this is particularly problematic as, nowadays, IoT systems are mostly insecure, and they are vulnerable to the injection of new components and to the modification of the behavior of existing ones (the development of adequate security solutions at the network level for IoT systems is a pending challenge), so, as a result, the reliability of the received data is generally compromised.

The complexity of real IoT deployments, and overall the lack of information about the elements which are also present

in the system, facilitates the appearance (deliberate or not) of malicious components; providing uncertain data, services, information, etc. Malicious components may be part of a cyber-attack (focused on modifying the system operation) and then they are deliberately introduced into the system. But, on the contrary, malicious components may also be the result of a malfunction or a programming bug and, then, they are not part of a deliberate strategy to attack the system.

In general, IoT architectures merge thousands of different hardware devices, software components, final applications. . ., which are often geographically sparse and conceptually very distant [6]. In this context, low-level information produced by hardware devices (usually using a binary data format) must be collected, aggregated, and transformed multiple times before escalating to the upper-level final applications (which, in a typical case, must receive data in a JSON object or XML document). However, as we said, no meta-information describing the underlying hardware (e.g. sensor accuracy or device identities) or the lower-level components is supplied to the upper layers.

The problem of detecting malicious components in IoT systems is analyzed by the authors in a previous work [7], considering that new tools are necessary to generate this lacking information at the top level from the available information about the lower level components. In particular, as final applications lack of knowledge about the system and possess limited control over the data infrastructure [8]; extremely important trust parameters in unsecured systems (as some IoT deployments) cannot be obtained using traditional solutions (for example, in standard web services, connections not based on HTTPS protocol are directly tagged as unsecured). Thus, parameters describing the QoS associated with provided services, or the uncertainty level associated with received data must be estimated using new solutions based only on information available at high-level.

Based on the contribution of our previous work [7], analyzing the estimation of the uncertainty level associated with the received data, in this paper we propose a statistical framework for detecting components with a malicious behavior, to prevent user applications to interact with them. Interactions with these components will be avoided if possible, or (if necessary) an alarm informing about a relevant malfunction in the system will be triggered. The proposal complements previous estimations with an enhanced statistical framework and a more sophisticated reputation model based on both implicit (based on deductions about the performed observation) and explicit (based on grades and direct recommendations) conceptions.

The objective of the proposed contribution is solving the most relevant current limitations of the state of the art. In particular, the development of this technology is motivated by the need of faster cyberattack detection techniques. Trust-based solutions have been proved to be adequate to be implemented in IoT deployments (as they are tolerant to complex relations among components), but existing calculation algorithms require a large convergence time, so fast cyberattacks can cause serious damage to IoT systems. The described contribution in this paper aims to address this challenge.

The rest of the paper is organized as follows: Section II describes the state of the art on reputation models and trust provision in IoT systems; Section III includes the detailed description of the hybrid reputation model and some notions of how the proposed framework is mathematically formalized; Section IV presents an experimental validation in a relevant simulated scenario (representing a real IoT deployment) in order to test the performance of the proposed solution; Section V describes the experimental results and Section VI concludes the paper.

II. STATE OF THE ART

Various Internet-of-Things reputation models have been reported in the last years [9]. Most of them are related to the emerging concept of ''social Internet of Things'', which refers to the idea that common objects may interact with other daily living things and create a social network as people do [10]. Furthermore, very commonly, ''reputation calculation'' is only an intermediate step in the obtaining process of trust models [11] (which are employed to infer the presence of malicious -untrustworthy- components). In comparison to the presented proposal in this paper, most works on social IoT do not offer a practical algorithm to obtain reputation, they only introduce the concept. On the other hand, although trust models are valid solutions to detect malicious components, they are usually more complex and heavier than reputation models, as the one proposed in this paper. Any case, from traditional security solutions, to current reputation-based solutions, all security proposals try to achieve the same information security objectives: confidentiality, integrity, availability, etc.

A. TRUST MANAGEMENT AND CALCULATION

Works about security and intrusion detection using reputation and trust models are divided into two groups. On the one hand, some works are focused on trust evaluation. In this case, researchers try to determine the network and physical parameters influencing trust (and/or reputation) in order to calculate all these parameters using direct observations or indirect measures (made by components supporting IoT modules) and some data analysis expressions [12]. Other works related to this topic define uncertainty taxonomies [13], uncertainty models [14] and processing algorithms [15]. In general, to detect a malicious component, a certain trust threshold is defined, and every component presenting a trust level below this limit is considered untrustworthy [11]. Some works employ network parameters such as the packet loss rate in order to perform the calculations [16], but others select definitions based on the social network theory [17]. Besides providing direct and indirect observations, a first idea of ''explicit recommendations'' is also included.

The proposed solution in this paper is based on previous definitions and understanding presented in the works cited above. However, it solves some existing problems, such as the

fast detection of burst, ephemeral or very fast attacks (which are hard to detect using the previous solutions, based on very heavy -and slow- algorithms) or the detection of malicious components which do not affect the network performance (specially compared to previous works based on network theory).

Some proposals considering reputation as fuzzy information have been also reported. In particular, in these works, both types of trust (and reputation) are defined: local and global [11]. This idea is also present in other previous works [7]. A special type of solution belonging to this first group consists of low-level reputation algorithms, implemented into hardware nodes and sensors [18].

These solutions are usually very powerful, as very different malicious behaviors may be detected, and well-founded decisions are taken. However, they require a long time to resolve if a certain component is malicious or not (as many observations are needed), and aggressive attacks or the protection of critical infrastructures are scenarios where these kinds of proposals fail. This work provides a hybrid model for reputation calculation (based on explicit and implicit reputation) to address these problems.

Among all the previously cited works, papers of Bao *et al.* [12] and Bao and and Chen [16], [19] are, nowadays, a reference for researchers in digital trust (especially in trust for IoT systems). In these works, trust is obtained through a reputation model where basic network parameters (such as the packet loss rate) are employed. Algebraic mathematical expressions (based on exponential functions) are defined to enable a reputation evolution according to social rules, whose inputs are the selected network parameters. Because of their relevance, papers of Bao et al. are the most adequate precedent to compare and evaluate the advances and contributions to the state of the art.

B. RECOMMENDATION SYSTEMS

As we said, in the context of reputation and IoT systems (focused on malicious behaviors detection) a second type of works may be found. These works propose a recommendation framework (similar to which are included in social platforms as eBay or AirBnB [20]), where IoT components may publish their opinion about the behavior of other modules [21]. Important facts as the reduction in trust over the distance of the information source in the social graph are studied and evaluated in this type of solutions [22].

In service-oriented architectures, works on service reputation are also found [23]. In particular, models based on user's trust evaluation in a service and service classifications [24]; as well as models considering authentication history and penalty [25] have been reported.

These solutions are much faster, so malicious behaviors are immediately detected and removed. Nevertheless, the rate of false positive detections goes up (as the amount of required information to determine a malicious behavior is lower) and, also, the critical number of malicious components necessary to attack the system and break the reputation algorithm is lower [26]. In this work, a hybrid model is specifically proposed to address these problems, by mixing slow but deep evaluation of reputation (based on implicit information) with a lighter calculation method based on recommendations.

In order to equilibrate both visions, in this paper the authors propose a hybrid solution, where an implicit definition of reputation based on direct and indirect observations is mixed with an explicit interpretation calculated from direct recommendations (or bad grades, depending on the case, see Section III).

Both definitions are integrated in a common expression, which tries to equilibrate the negative effects of each view with the benefits of the other.

III. A HYBRID REPUTATION MODEL

In IoT systems, trust must be more than some mechanisms that reduce IoT component uncertainty as they interact with other parts of the system, although such mechanisms are important in helping components to choose an adequate remote module to interact with [26]. Besides, in IoT, in order to face the most recent challenges related to cyber security, such mechanisms must be able to define trust and reputation in a dynamic and collaborative way, so malicious components could be detected in the most efficient and fast manner.

Therefore, in the first part of this work [7] both concepts were defined: local and global reputation. In local reputation, only observations made by the IoT component under study are considered. In global reputation, an average of all local reputations calculated in the system is obtained.

Figure 1 shows the basic scenario. In this scenario, every IoT component has a ''trust circle'' made of the IoT modules it trusts unconditionally (the reputation of these nodes is not evaluated as the information provided by them is always considered trustworthy -for example, because they belong to the same owner or because they implement additional security policies-). The trust circle dynamic calculation is not described in this work, so a configuration design is considered. It must be noted that these components could also be the objective of a cyber-attack and, eventually, become malicious (a critical situation modifying the entire system operation). Detecting this situation is a relevant challenge,

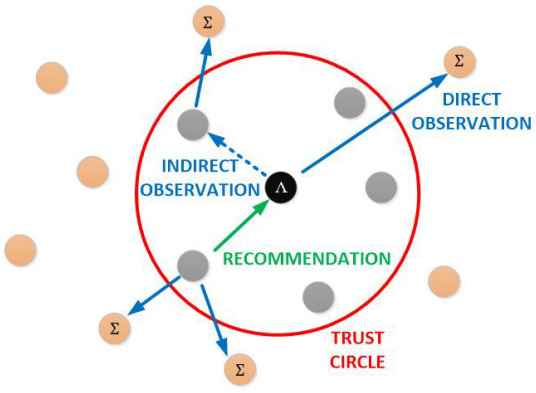

FIGURE 1. Basic scenario for reputation calculation.

so future works should investigate an adaptation of the proposed solution based only in direct observations (i.e. only the implicit reputation is considered) to suppress the trust circle, evaluate its members and detect and remove possible malicious nodes.

As can be seen, the component under study (in black) may directly observe some external components it does not rely on (in orange). However, information about implicit reputation (lines in blue) may be also obtained through an indirect procedure. In this case, nodes in the trust circle provide information to the IoT module under study based on their own direct observations. Finally, components in the trust circle may share recommendations about the external modules, which are employed to obtain the explicit reputation of the external IoT components.

In this context we understand direct or explicit recommendations as messages where IoT components clearly describe their knowledge about the behavior of other IoT modules (no proofs are provided and no knowledge discovering algorithm is needed; it is an explicit description). For example, in these messages, modules may indicate whether components are trustworthy or not; or they can indicate a high score of node trustworthiness. Messages describing bad behaviors are negative recommendations, and messages describing good behaviors are positive recommendations. Contrary to this notion, implicit reputation is based on the information inferred from regular data or control messages, which do not contain in their payload any description about reputation or system behavior. In that way, implicit reputation is obtained from implicit information about the IoT modules' behavior included in every message they generate, process, transmit, etc.

It is important to note that components in the ''trust circle'' are not the objective of this proposal. These components are necessary to generate trustworthy recommendations and to know about modules not directly connected to the node under study (black node in Figure 1). However, this ''trust circle'', in general, does not include all components required to allow a regular system operation (it includes only some selected modules, very special, secured, trustworthy, etc.), so a mechanism to know about other external nodes (white nodes in Figure 1) is required.

In this context, we define the reputation $\mathcal{R}_\Sigma$ of an external IoT component Σ as the geometric mean of the implicit reputation $\mathcal{R}^i_\Sigma$ and the explicit reputation $\mathcal{R}^e_\Sigma$ (1).

$$\mathcal{R}_\Sigma = \sqrt{\mathcal{R}^i_\Sigma \cdot \mathcal{R}^e_\Sigma} \tag{1}$$

Implicit reputation is obtained by means of a statistical knowledge extraction process, from regular interactions among components. On the other hand, explicit reputation is obtained from explicit recommendations published by nodes in the trust circle.

The use of the geometric mean (instead of, for example, the arithmetic mean) allows the global reputation to follow the evolution of the reputation value (implicit or explicit) that detects a change in the system situation in a faster, stronger and more stable way. In that way, the global reputation presents the same good behavior in situations where explicit reputation better detects malicious components, as in situations where implicit reputation offers better results.

Although the proposed calculation methods for both types of reputation are slightly different from the ones described in other works (to allow a faster and more scalable solution), the presented understanding of reputation is totally compatible with other previous and reference works. Previous papers consider reputation in IoT systems as ''a measure derived from direct or indirect knowledge or experiences on earlier interactions among entities [9]''. Various relevant works calculate node reputation using information from interactions among components (implicit reputation, direct knowledge) [10], recommendations from a set of nodes the component trusts (explicit reputation, indirect knowledge) [9] or combinations of these information sources (the global reputation) [19].

In order to determine if an external component is malicious, it must be evaluated if its reputation is lower than the trust threshold μ_Λ (different thresholds may be defined for various components, considering, for example, the component class).

Next sections describe the calculation procedure of both considered types of reputation.

Any case, the proposed detection procedure (focused on eventually triggering an isolation process) is a new tool to reach some protection goals. In fact, although the isolation process or the actions to be taken when a malicious component is detected are not discussed in this paper, the detection procedure should be always planned considering these objectives.

A. EXPLICIT REPUTATION CALCULATION

In this work we suppose each IoT component is provided with a recommendation algorithm, being able to publish grades and notes regarding the behavior of the external nodes. The employed procedure to calculate and decide about the recommendation type which is published at each moment is not analyzed in this work. However, some useful proposals may be found in the literature [27], [28]. In respect to the sharing process, a Publication/Subscription network [29], for example, could be deployed to publicly send information to the trust circle.

Using the described infrastructure, an IoT module (represented in Figure 1 as a black node) receives direct recommendations from every module in its trust circle (grey nodes in Figure 1). Recommendations related to the same external IoT module Σ (orange nodes in Figure 1) are employed (by the central component, represented in Figure 1 as a black node) to infer the explicit reputation $\mathcal{R}^e_\Sigma$ of the external module Σ under study. The evaluation algorithm is graphically represented on Figure 2.

The proposed algorithm has three steps. First, recommendations are grouped depending on the components they

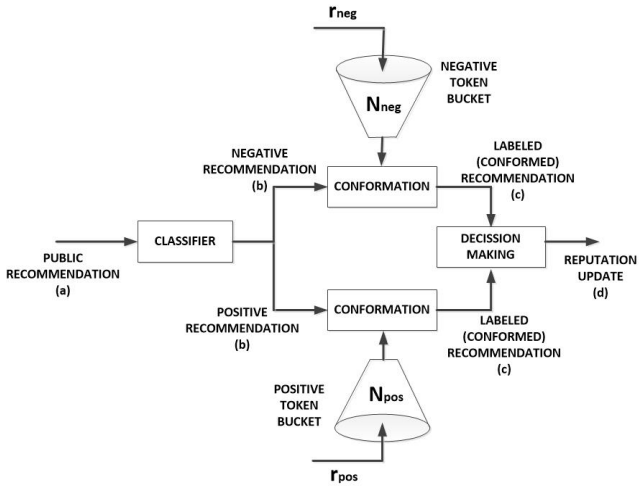

FIGURE 2. Graphic representation of the basic explicit reputation calculation algorithm.

describe. Besides, recommendations related to the same IoT module are divided into positive and negative recommendations. Before considering received recommendations as relevant information to modify the explicit reputation of a node, these messages must be filtered. As IoT systems can fluctuate, a certain amount of positive and negative recommendations per time unit are tolerated (i.e. they are received, but the information is not relevant). Only recommendations (positive and/or negative) that surpass these rates are considered to obtain a new value for explicit reputation. This new value is obtained by a decision-making module which follows a collection of predefined rules to update reputation depending on the received flows of positive and negative recommendations.

As can be seen, in this algorithm, we are adapting the concept of ''token bucket'' [30] defined in the traffic engineering field. Basically, this model (contrary to other models such as the leaky bucket) does not force the packet losses appearance and accepts traffic bursts. However, it establishes certain limits to the number of received recommendations, considering them not relevant in a specific time slot. These properties help us to ensure that decision is based on permanent (or, at least, perdurable) behaviors, while fast changes may be also addressed if they are strong enough (accidental events, as we see, are not taken into account).

Thus, the algorithm considers two different token buckets: the bucket of negative tokens and the one of positive tokens. Negative tokens are generated at a rate r_{neg} and the bucket has a maximum capacity of N_{neg} negative tokens. In a similar way, positive tokens are generated at a rate r_{pos} and the bucket has a maximum capacity of N_{pos} positive tokens. Each time a recommendation related to the same IoT component is received; it is classified into two different groups: positive or negative recommendations. Then, the positive and negative recommendation flows are conformed. To be conformed, each recommendation in a flow consumes a token from the corresponding bucket and gets validated (filtered). If no tokens are available, the recommendation is not validated. In the token bucket paradigm, each time a message is received it tries to take a token from the bucket. A recommendation that obtains a token from the bucket is said to be validated (filtered). As tokens are unconditionally generated at the corresponding rate (until the bucket is totally filled), short bursts of positive or negative recommendations are tolerated (contrary to leaky bucket filters). Once the bucket is empty (the maximum acceptable burst duration is overpassed), only recommendations are validated at the token generation rate. Any recommendation above this limit is not validated.

Validated and non-validated recommendations are then sent to a decision-making module. In this module four different actions may be performed:

- If only validated recommendations are received by the decision-making module the explicit reputation is not modified.
- If only non-validated negative recommendations are received by the decision-making module, explicit reputation is degraded.
- If only non-validated positive recommendations are received by the decision-making module, explicit reputation is upgraded.
- Finally, if both, non-validated positive and non-validated negative recommendations are received by the decision-making module, an ambiguous situation is detected. Usually, this situation is due to different IoT components publishing opposite recommendations about a same IoT module. In this case, the decision-making module may request additional IoT components (belonging to its trust circle) for new recommendations and follow the majority criterion (explicit reputation is upgraded if most recommendations are positive and vice versa). If the ambiguous situation remains, explicit reputation is not modified.

The entire algorithm is showed as flowchart in Figure 3.

Then, the explicit reputation may take three different values (2). In a standard case, where no recommendations have been received (or all of them have been validated) explicit reputation is considered equal to implicit reputation. If positive non-validated recommendations are received reputation is set to 1. If negative non-validated recommendations are received reputation is set to 0.

$$\mathcal{R}^e_\Sigma \in \left\{0, \mathcal{R}^i_\Sigma, 1\right\} \tag{2}$$

With this design, fluctuations due to punctual problems are removed, and final reputation is not affected. Furthermore, a certain rate of negative (or positive) recommendations is accepted as typically IoT solutions are very unstable systems, many times based on opportunistic communications (so a certain fault-tolerance must be considered in those systems).

The use of token buckets also allows considering the previous reputation for calculation of current reputation. In fact, explicit reputation model is based in the actualization of the

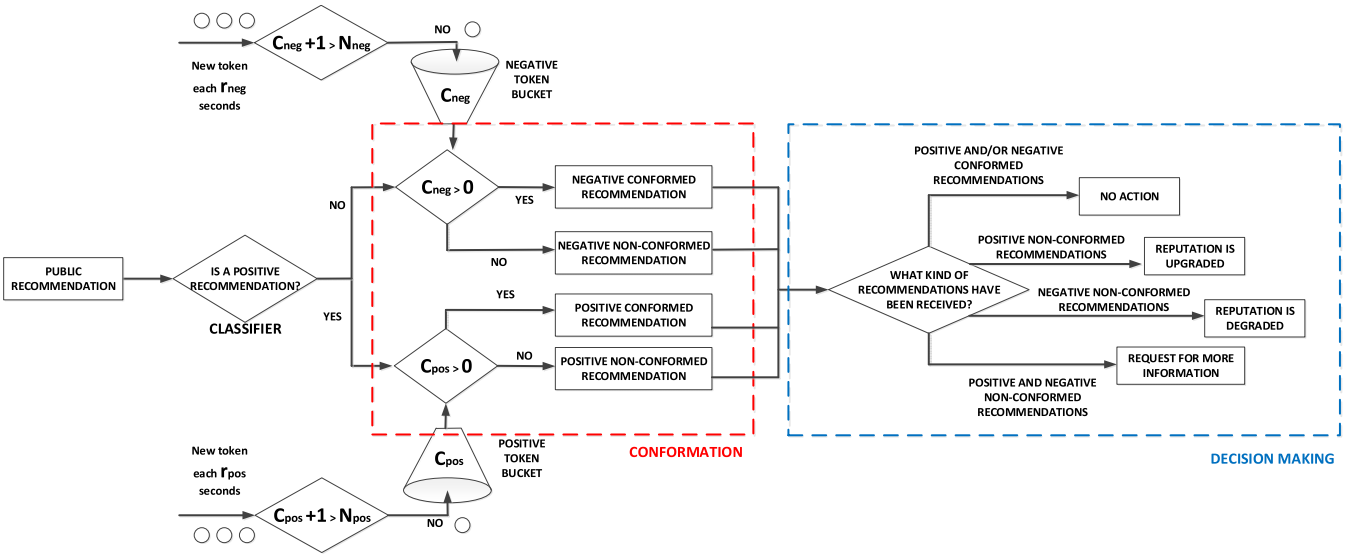

FIGURE 3. **Detailed flowchart of the basic explicit reputation calculation algorithm.**

current reputation to obtain its future value. This approach could enable the use of predictive systems to select, employing past information, the most trustworthy communication channels among sensors.

Only one scenario must be described with more detail: if both, a non-validated positive and a non-validated negative recommendation are received. In practical scenarios, it is very strange to receive at the same instant two notifications; however, all messages received during a certain time slot are describing the same situation. Therefore, each time a recommendation is received, the decision-making module (before updating the value of the explicit reputation) triggers a timer T_{slot}. Until the timer expires, all received recommendations are considered together. Once the timer expires, the decision-making process starts. If a conflict appears, then, a consultation procedure is triggered. The IoT node under study considers the received recommendation, and asks to the rest of components in its trust circle to send an explicit recommendation about the external module being analyzed. If most components (including those which generated the conflict) send positive recommendations, reputation is upgraded. On the contrary, if most components send negative recommendations, reputation is degraded. If the conflict gets unresolved (because a similar number of positive and negative recommendations are received, or because components send neutral recommendations) no action is performed.

B. IMPLICIT REPUTATION CALCULATION

As implicit reputation computation is a complex procedure, we are explaining the proposed method using a particular scenario. In certain scenarios there are two IoT components named as Λ and Σ. Λ (black node in Figure 1) is calculating the reputation of Σ using implicit information in the transactions between these two modules.

The implicit reputation computation was briefly described in our previous work [7], where concepts such as global and local reputation were presented. In fact, the global implicit reputation $\mathcal{R}^{i}_{\Sigma}$ of a given IoT component Σ, is defined as the global perception of the component's behavior. Particularly, it is measured if transactions including this component generally present positive outcomes. This vision, however, is very difficult to implement in this way, as several actors may be involved. Therefore, to facilitate this task, the $\Lambda-$local implicit reputation of a IoT component was defined. In this context, the Λ-local implicit reputation of an IoT component Σ, $\mathcal{R}^{i}_{\Sigma}|_{\Lambda}$, is the local perception of the behavior of the component Σ in a given system's component Λ. Particularly, it is measured if transactions including both components generally present positive outcomes. With this definition, it is immediate to define the relation between $\mathcal{R}^{i}_{\Sigma}$ and $\mathcal{R}^{i}_{\Sigma}|_{\Lambda}$ as (3), where $\mathfrak{C}$ is the set of all components in the system and λ_{Λ} the relative weight of $\mathcal{R}^{i}_{\Sigma}|_{\Lambda}$.

$$\mathcal{R}^{i}_{\Sigma} = \sum_{\Lambda \in \mathfrak{C}} \lambda_{\Lambda} \cdot \mathcal{R}^{i}_{\Sigma}|_{\Lambda} \tag{3}$$

As said in Section II, the social Internet of Things is based on the ability of technology-powered objects to establish social networks as people do. In fact, in relation to human social networks, reputation is a very popular concept. Several works on the factors that allow a person [31] or a company [32] to estimate their reputation have been reported. Taking into account these previous works, three main parameters may be considered as the main factors that determine the reputation of an entity:

- Nobleness ($\mathcal{N}$): It refers to the perception of the component as an honest entity, which provides true information.
- Solidarity ($\mathcal{S}$): Sometimes it is also named as "social responsibility". It refers to the perception of the

component as an entity committed to the objective of the system (being willing to answer the request of other components, provide resources if necessary, etc.).

- Relevance ($\mathcal{R}e$): Important components in the system have better reputation than entities implementing very common functionalities. It refers to the specific weight or importance of a certain component in the system. In general, in this work, a component is more important as its functionalities are less redundant in the system. A component is essential if no other component has the same capabilities. This definition is coherent with other previous definitions [16].

Some of the previously cited parameters have been defined as key elements in reputation calculation in some existing works. For example, the provided definition for nobleness is consistent with the idea of ''honesty'', present in many works about security and IoT systems. Furthermore, the concept of ''relevance'' is included to preserve system availability anytime (as important or essential components can be rarely considered malicious and isolated from other components).

Then, considering the results extracted from that previous analysis, the $\Lambda-$local implicit reputation may be calculated as indicated below (4). In this expression, α, β and γ are dimensionless scalar parameters (or weights) employed to control the relative importance of the three components of implicit reputation.

$$\mathcal{R}^{i}_{\Sigma}|_{\Lambda} = \begin{bmatrix} \alpha & \beta & \gamma \end{bmatrix} \cdot \begin{bmatrix} \mathcal{N} \\ \mathcal{S} \\ \mathcal{R}e \end{bmatrix} \tag{4}$$

In the simplest IoT systems and scenarios each low-level entity (such as sensors) communicates in a unidirectional way (e.g. sensors do not receive messages or information, they generate it) with a final application (see Figure 4(a)). In this case, relevance and solidarity may be considered null [7], as components only interact in pairs (no supportive behaviors are allowed, and relevance has no sense -all the components are essential-). However, in this work, we are considering a more complex scenario, where several low-level IoT components communicate (e.g. sensors) with the same final application in a bidirectional way (see Figure 4(b)). In this scenario, the calculation of nobleness becomes more complicated (as there are several data flows) and solidarity

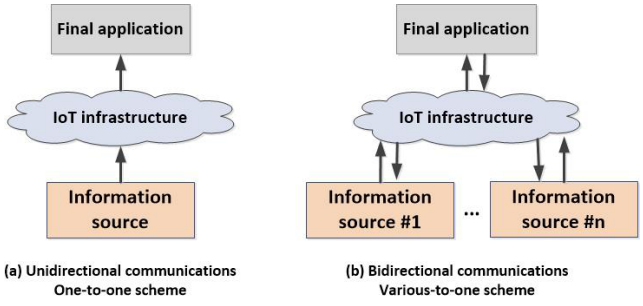

FIGURE 4. Scenarios for the implicit reputation calculation.

and relevance are not null. Below, a description about the calculation process of all these parameters is provided.

The calculation procedure for ''nobleness'' is the most complex one. It has been described in a detailed way in our previous work [7]. In this context, the nobleness $\mathcal{N}$ of an IoT component Σ, according to a second component Λ, is defined as the expectancy of Λ to obtain correct information from Σ. This information may be produced data or some other meta-information (e.g. the offered QoS for a certain service). Previous experiences affect nobleness calculation. These experiences are weighted to limit the effect of time-distant events. Moreover, as said, it is difficult to establish the nobleness of a component based on a few interactions. Thus, a threshold N_{th} must be defined in order to accumulate a significant number of measurements to start estimating the nobleness value. Equation (5) represents the mathematical model for nobleness calculation; where n is the number of accumulated nobleness measurements and h is the weighted ratio of the number of times the component behaved nobly (i.e. it sent correct information).

For this algorithm, operation time is divided into time slots. For each time slot, information about the behavior of component Σ is collected. In order to calculate reputation at the current time slot, not only information collected during the current slot has to be considered, but also all previous observations. However, as time passes, previous observations are less representative of the current reputation. Therefore, the total percentage of successful transactions cannot be obtained by adding the percentage obtained for each time slot, but old information must be firstly weighted to reduce its influence. h represents this ratio of successful (i.e. trustworthy) transactions, where old information has been weighted to reduce its influence.

$$\mathcal{N} = \begin{cases} 1; & n < N_{th} \\ \sqrt{2}\frac{h}{\sqrt{1+h^2}}; & n > N_{th} \end{cases} \tag{5}$$

$$h = \sum_{j=0}^{n} u[-j] \cdot r^{j+1} \tag{6}$$

As can be seen, nobleness follows an algebraic function belonging to the sigmoid class. Thus, $\mathcal{N} \in [0, 1] \forall h \in [0, 1]$. Moreover, the model includes the presumption of nobleness, as every component is sincere ($\mathcal{N} = 1$) until enough measurements are collected. To calculate the weighted ratio h, we consider a geometric sum (6); where the common ratio r can be freely fixed (in order to limit the influence of the past behaviors, as $|r| \rightarrow 0$). As explained before, h represents a ratio where old information has been weighted to reduce its influence. As information gets older, the weight coefficient must be smaller (until reaching zero). The array of employed coefficients may be defined one by one by the system administrator (considering the previous indications), but in general it is possible to obtain a much better performance of the proposed algorithm if coefficients are defined by following a pattern. In this case an exponential pattern is proposed.

A common radio r (smaller than one) must be selected. Information obtained during each time slot is weighted using a different power of this common ratio. In fact, as the power index goes up, the coefficient goes down, weighting old information by smaller coefficients.

The proposed weighting pattern is selected to be coherent with previous studies about information propagation and the value and relevance of information as time passes. Information trustworthiness is, in general, proved to decrease exponentially [38] (as represented in geometric series); although other weighted observation methods (quadratic, for example) have been reported for specific scenarios, which are not applicable in the described technology.

The sequence $\boldsymbol{u}[\cdot]$ represents the natural ratio of the number of times the component behaved honestly in every time slot. $\boldsymbol{u[j]}$ is defined (7) as the quotient between the times the component provided correct information in the j-th time slot, represented by $\boldsymbol{p_j}$, and the total number of transactions in that time slot, named as $\boldsymbol{t_j}$.

$$u\,[j] = \frac{p_j}{t_j} \tag{7}$$

In order to calculate whether an IoT component has provided correct information in a certain transaction (interaction or data provision to the component calculating the reputation value -black node in Figure 1-), we are evaluating the uncertainty level θ associated with the provided information. If this level remains below a certain threshold μ_h the component is considered to be honest (noble). In some cases, the uncertainty perceived by the component performing the calculation is directly caused by the analyzed low-level component (e.g. uncertainties can be caused by network problems). However, from the final applications' point of view, the provided information is uncertain, and, in consequence, the low-level component is considered as not noble.

Then, to evaluate the uncertainty level θ we propose the following statistical model. Figure 5 shows the most basic representation of the scenario under study (although the same statistical framework may be employed in more complicated scenarios). A final application received from an information source (IoT component) a certain information $\overline{\overline{x}}$. In IoT systems, the uncertainty level associated with $\overline{\overline{x}}$ is the addition of the following two variables: $\mathfrak{T}_{IT}$ and $\mathfrak{T}_{PHY}$.

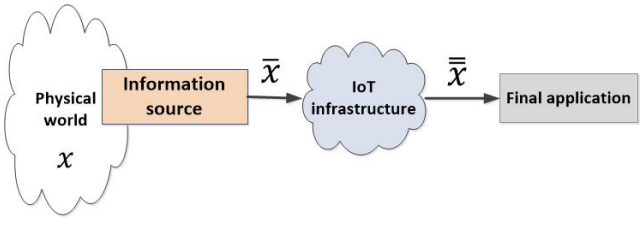

FIGURE 5. Basic calculation scenario for the uncertainty level.

First, the uncertainties $\mathfrak{T}_{IT}$ about the equivalence between the received information $\overline{\overline{x}}$ and the information generated by the information source $\bar{x}$. In this first case, the relation between both data may be described as a surjective stochastic application $T[\cdot]$, as every information $\overline{\overline{x}}$ must be the image of a certain information $\bar{x}$. These uncertainties are caused by the IoT infrastructure, so they are IT (information technologies) uncertainties. And, second, the uncertainties $\mathfrak{T}_{PHY}$ about the equivalence between the information generated by the information source $\bar{x}$ and the real information existing in the physical world x. These uncertainties are caused by physical limitations in the information capture. Therefore, they are physical uncertainties.

Thus, associated with a received information $\overline{\overline{x}}$ there exists an enumerable set of uncertainty sources $\mathfrak{T} = \{\mathfrak{T}_{PHY}, \mathfrak{T}_{IT}\} = \{i_k, k = 1, \ldots, K$ whose cardinality K may reach the cardinality of the natural numbers $\aleph_0$. This uncertainty sources transform the process of acquiring a certain information $\overline{\overline{x}}$ in a random experiment ε, which takes values from the discrete sample space Ω.

Each uncertainty source is described as a bi-varied stochastic process (8), being Ψ the sample space of all possible values (real, considered in protocols, applications, etc.) for the uncertain event and $\boldsymbol{\psi}$ an element of this sample space. In these statistical expressions, m represents the time slot (or time instant), $\boldsymbol{\Omega}$ represents the sample space of all possible received values (not only real values considered in applications and protocols, but also erroneous values, incoherent messages, etc.) and $\boldsymbol{\omega}$ is an element of this sample space.

$$i_k \asymp \mathrm{X}_k[m; \omega, \psi)/\omega \in \Omega. \quad \psi \in \Psi \tag{8}$$

The stochastic processes represented before (8) are discrete in time, as final applications are cyber components (and, therefore, digital elements), but the sample space Ψ may be continuous or discrete, depending on the nature of the uncertainty source. For example, the measurement error has a continuous nature; however, the possibility of suffering a cyber-attack is described by a discrete variable. Furthermore, in general, uncertainties' values change slowly, so these stochastic processes may be considered stationary during a time slot. As they are unknown effects, stochastic processes are expressed in a parametric way, depending, each one, on a certain parameter ϑ_k. Three basic probability density functions or probability distributions may be used to describe uncertainty sources: uniform distributions, triangular distributions and Gaussian distributions (see Figure 6).

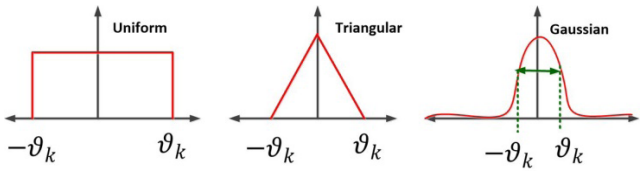

FIGURE 6. Basic probability functions.

Uniform distributions are employed to describe unknown effects limited to the range $[-\vartheta_k, \vartheta_k]$. In data acquisition processes this is the most general distribution, as the sample space is bounded. Triangular distributions are employed

when, besides the variation range, the error probability goes down as its value goes up. Finally, if more information is available (for example, if noise is considered) a Gaussian distribution with typical deviation ϑ_k can be employed.

Considering $\mathcal{F}$ the set of parts of Ω, and a function $P(A)\, A\epsilon\mathcal{F}$ which fulfills the three Kolmogorov's probability axioms, the experiment ε is completely characterized with the event algebra $E = \langle\Omega, \mathcal{F}, P(\cdot)\rangle$, which (additionally) is a $\sigma-$algebra. In this context, it is possible to create a partition $\Pi = \{\pi_1, \ldots, \pi_p$ of Ω, and select a main value $\delta_i \in \pi_i$ representing every cluster. Partitions contain the same elements as Ω, but grouped in several sets (homogenous or not). These partitions, in this proposal, are employed to reduce the number of possible received values, as each partition or set is now considered as a unique element, so it is important to consider them. Then, the process to estimate the uncertainty level $\theta[j]$ in the j-th time slot is as follows.

When certain information $\bar{\bar{x}}$ is received, it is included in the observations vector v_j of the current time slot (9). Considering Figure 4, x represents the real information (the real value of temperature, for example) and $\bar{x}$ represents the information acquired or generated by the hardware platform (the temperature value obtained by sensors, for example). However, in this case, we are obtaining reputation from the information finally obtained by high-level applications, called observations hereinafter, (representing by $\bar{\bar{x}}$), which is obtained through processing, composing, etc., the information $\bar{x}$ generated by the hardware platform.

$$v_j = \left(\bar{\bar{x}}_1, \ldots, \bar{\bar{x}}_m\right) \tag{9}$$

As these observations are independent, it is possible to calculate the value of the ϑ_k parameter for each uncertainty source using the maximum likelihood estimation (MLE) [33] algorithm and the vector v_j. This method is the most adequate as prior probability distributions are unknown. Then, the partition π_i to which belongs the received information (i.e. the real message which most probably it represents) is located. First, for each uncertainty source i and time slot j, the probability ρ_i^j of the information belonging to the partition π_i in that time slot is calculated (10). In this expression, X_k, $\bar{\bar{x}}$ and ψ maintain the same meaning as before. This expression is a direct application of the probability calculation from random variables.

$$\begin{aligned} \rho_i^j &= \int_{\pi_i} \mathrm{X}_k[j; \bar{\bar{x}}, \psi] d\psi \quad \text{or} \\ \rho_i^j &= \sum_{\psi\in\pi_i} \mathrm{X}_k[j; \bar{\bar{x}}, \psi] \end{aligned} \tag{10}$$

Later, in that way, as uncertainty sources are also independent, the global probability (considering all possible uncertainty sources) ρ^j of $\bar{\bar{x}}$ to belong to π_i during j time slot is calculated as a probability multiplication (11).

$$\rho^j = \prod_{k=1}^{K} \rho_k^j \tag{11}$$

Finally, the information δ_i (the representative value of the partition or cluster π_i, as defined before) is received with an uncertainty level $\theta[j]$ calculated as indicated in (12).

$$\theta[j] = 1 - \rho^j \tag{12}$$

In this transaction, the information source is considered to be honest if it meets the condition explained above ($\theta[j] > \mu_h$).

In the simplest scenario, the obtained value for nobleness is directly associated with an external IoT component. However, if various external components are providing information at the same time (the component under study receives the aggregation of all these data), this value must be "divided" into all the involved components. To do that, we are applying a ponderation factor based on the previous estimation of the nobleness for each node (13).

$$\begin{aligned} \mathcal{N}_\Sigma[n] = \mathcal{N}_{measure}[n] \\ \cdot\left(\frac{\mathcal{N}_\Sigma[n-1]}{\sum_\Sigma \mathcal{N}_\Sigma[n-1]} + \left(1 - \frac{\mathcal{N}_\Sigma[n-1]}{\sum_\Sigma \mathcal{N}_\Sigma[n-1]}\right)\right. \\ \left.\cdot\,\delta\left[\mathcal{N}_{measure}[t] - 1\right]\right) \end{aligned} \tag{13}$$

Basically, this expression calculates the value of the nobleness of a certain IoT component $\mathcal{N}_\Sigma[n]$ from the nobleness obtained from the aggregated observations $\mathcal{N}_{measure}[n]$. Then, this value is weighted by a ponderation factor which is equal to the unit if $\mathcal{N}_{measure}[n] = 1$ (all remote components are honest if the aggregated perception indicates that), but which depends on the past value of the nobleness for each component if $\mathcal{N}_{measure}[n] \neq 1$. In this context $\delta[\cdot]$ represents the Kronecker's delta function.

Once nobleness is obtained, the solidarity of the external IoT components must be evaluated.

Solidarity is obtained by employing a similar mathematical framework to which described in the case of nobleness, but implies a much simpler statistical analysis (as related information is directly obtained, not having to be inferred by means of estimation procedures). In particular, solidarity also follows an algebraic function (14) belonging to the sigmoid class (evolving from 0 -minimum value- to 1 -maximum value-).

$$S = \begin{cases} 1; & n < N_{th} \\ \sqrt{2}\frac{s}{\sqrt{1+s^2}}; & n > N_{th} \end{cases} \tag{14}$$

$$s = \sum_{j=0}^{n} w[-j] \cdot k^{j+1} \tag{15}$$

In order to obtain the solidarity value, a weighted ratio s representing the number of times the component behaved in solidarity is also defined (15). This ratio employs a geometric series to aggregate the past results related to solidarity. These results are represented by $w[j]$. This parameter is understood as the natural ratio of the number of times the component behaved in solidarity in every time slot. Particularly, $w[j]$ is defined (16) as the quotient between the times the component

answered in a positive manner to the requests of the IoT node under study (and it trust circle) in the j-th time slot a_j and the total number of requests performed in that time slot q_j.

$$w[j] = \frac{a_j}{q_j} \tag{16}$$

If information from various remote components is aggregated in a same high-level datum, then, the obtained evaluation of the solidarity parameter must be divided into all the involved modules as described before (13).

Finally, obtaining the relevance of an IoT module in a system is the simplest process. The relevance of an IoT module in a system for a certain IoT component under study depends on two variables: the presence of other components with the same functionalities in the system e; and nl, how the component is dependent on these capabilities. As both conditions must be present at the same time for a component to be relevant, relevance is obtained as the geometric mean of both amounts (17).

$$\mathcal{R}e = \sqrt{e \cdot nl} \tag{17}$$

The calculation of e parameter is very simple, as it is defined as a redundancy ratio (18). Where $\mathfrak{C}_\Sigma$ is the set of components with the same capabilities than Σ; and $\mathfrak{C}$ is the total set.

$$e = \frac{card\{\mathfrak{C}_\Sigma\}}{card\{\mathfrak{C}\}} \tag{18}$$

On the other hand, nl parameter is obtained as honesty and solidarity, by means of a natural quotient and considering the number of times the IoT component under study employs the capabilities of the remote module and the total number of performed transactions. The obtained result is weighted using a geometric series and is included as independent variable in a sigmoid function.

Relevance, usually, changes slowly, as important changes in the system should be developed: new components or new applications, disconnecting some parts of the system, etc. On the other hand, nobleness and solidarity may change in a very dynamic way, as more information is received and collected. Ratios in the geometric series control the effect of the past observations and, then, the convergence speed (see Section V).

C. GLOBAL OVERVIEW

Once the calculation of both types of reputation is described, in this section we are explaining the global meaning of the entire model.

First, it must be noted that $\mathcal{R}_\Sigma \in [0, 1]$, as every parameter in the calculation process, is also evaluated in the interval [0, 1]. In practice, it allows a very fast data interpretation and aggregation, as well as the integration of our proposal with other reputation management solutions which, usually, also define reputation in the interval [0, 1].

The proposed hybrid model works in this way. As time passes, IoT components collect information by means of direct and indirect observations, and implicit reputation is evaluated. However, as a certain amount of information is required and past events are also considered in this type of reputation, fast changes and aggressive attacks are perceived too late. To address this problem, IoT components may also publish recommendations about external IoT components. If the rate of received negative recommendations exceeds a certain limit, the explicit recommendation is degraded. In the worst case, explicit reputation is equal to zero, and (as reputation is defined as the geometric mean of both models -implicit and explicit), the entire reputation gets canceled.

IV. EXPERIMENTAL VALIDATION: A FIRST CASE STUDY USING SIMULATION TOOLS

An experimental validation based on a simulation scenario was planned to evaluate the proposed solution. The proposed validation includes two parts. In the first part, a security analysis is performed, using various attack scenarios to demonstrate the proposed technology's resilience to cyberattacks. In the second part, the performance of the proposed technique is evaluated (using simulation tools), and it is compared to previous similar proposals.

A. SECURITY ANALYSIS

Five different attack scenarios and classic security issues in IoT deployments [40] are discussed in relation to the proposed technology. In particular the following study cases are considered:

- Unauthorized device insertion: In this scenario an unauthorized device gets access to the systems at physical and network level, and starts injecting false sensing information. Although this device may be provided with valid credential and permissions, the proposed security solution must identify the new information as erroneous and isolate the new devices for being malicious.
- Data integrity: In this scenario, data integrity is comprised. Causes may be varied: from sensitization nodes that have been infected and their behavior modified, to an increase in the electromagnetic interferences or the packet loss rate. Basically, in this case, authorized devices are generating and providing information whose correctness is not guaranteed.
- Replay attack: This attack consists of a malicious device that impersonates an authorized node. Original data generated by the authorized node are intercepted and modified before being sent another time by the malicious device.
- Unauthorized data collection: This attack is the basic privacy violation problem. An unauthorized device gets access to the systems at physical and network level, but instead of injecting false information or affecting the data generated by other components, only collects information being transmitted in the system. The objective is to access to the users' private information.
- Forward and backward security: As the proposed solution is based on previous experiences to determine the

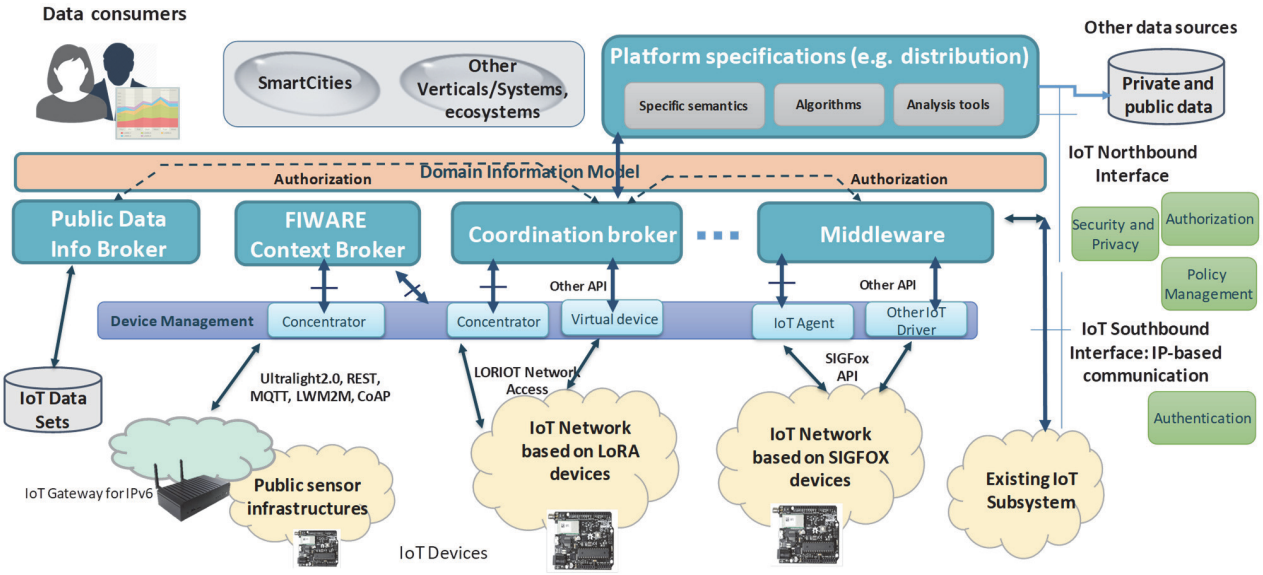

FIGURE 7. Simulation scenario.

reputation and trustworthiness of IoT components in the system, malicious components may try to attack the system by learning about its behavior. In this scenario, an attacker collects past information about the system, to use it as instrument to inject false information in the system but maintaining a good reputation of the malicious node.

B. PERFORMACE EVALUATION

Using advanced simulation techniques and the NS3 network simulator, a real IoT system was implemented including the proposed solutions. NS3 is a network simulator executing scenarios and network behavior by some logic programmed in C++.

The proposed simulation scenario is an adaptation of a real European deployment (see Figure 7). The selected deployment belongs to the FIWARE initiative [36]. The objective of the proposed system is to provide future IoT users with an environment to deploy their applications, guarantying the access to IoT platforms. Basically, it consists of a set of hardware devices whose data and control messages are managed together using a device management middleware. Then, using specific brokers and a semantic layer, it is possible (in the highest layer) to host user applications (those which try to obtain and calculate the reputation of hardware devices). Although particular details about the proposed architecture (Figure 7) are not significant, it is important to remark the idea that information is generated by hardware components, then it is processed by several components and modules (a device management middleware, different brokers, a semantic layer, etc.) and finally it is received by applications; exactly as described in the motivation scenario.

The proposed architecture contains four different information sources. These sources are sensor networks implemented using various radio access technologies and presenting different data access licenses. In particular, it is considered: two proprietary networks (based on LoRA and SIGFOX technologies), a network made of public sensors communicating using any of the existing free technologies (Bluetooth, ZigBee - although some implementations require paying an annual fee for usage-), and, finally, an interface to connect existing legacy systems (traditional industrial solutions) with other software and high-level components.

Different communication protocols and components are considered in order to represent a real heterogeneous IoT deployment. Nevertheless, presented results in Section V are independent from the employed communication technologies, as proposed algorithms operate at a very high abstraction level and independently from the underlying hardware platform (this is one of the advantages of the proposal, as very complex infrastructure may be protected using this technique).

Information sources in the simulation scenario are geographically sparse, so interferences and delays must be considered (in this case we are considering the standard configuration offered by the NS3 simulator for these non-ideal effects).

This proposed basic scenario change during simulations but at a quite low speed. Components with a good behavior turn malicious and vice versa. Besides, some components may get isolated by electromagnetic interferences, packet losses, etc. The proposed solution, as it is continuously monitoring the systems, must be able to detect these changes after a certain converge time (required to update the trust estimations of IoT components). Expressions (5) and (6) shows the actualization rules.

In this simulations scenario, the four low-level sensor networks were designed to present one hundred and fifty (150)

components each. Besides, inside each network, ten (10) components were forced to behave in a malicious way. Malicious components were infected in a random way. Any malicious component could become non-malicious in another moment, also, in a random way. The percentage of malicious components in a system at each moment during a simulation was always between 0% and 7%. In order to reach a malicious behavior, basically, various attacks were performed (an intruder was supposed to control the component, causing this component to provide erroneous information) and some harmful effects were activated: the precision of the instruments was reduced; the electromagnetic interferences were maximized, and the packet losses were strengthened.

In this context (Wireless Sensor Networks), the erroneous information is formally defined as information which does not represent the real current state of the physical world. In the proposed simulation scenario, the malicious components replace results obtained from the implemented physical model for temperature, humidity, etc., for random values. The objective of the validation is to prove that sources with this behavior are correctly identified using the proposed technology, which considers values generated by other devices, previous information accepted as correct, etc.

The duration of malicious behaviors was selected in a random way (during the setup of each simulation). Any case, this time period was between 0.5% and 100% of the simulation time. Sensors in these network generated data with a rate 1000 messages per hour.

The causes for a component to become malicious in real systems are very different: from cyber-attacks to hardware malfunctions. Any malicious behavior, any case, has a common characteristic: at the end, the result is the provision of false or uncertain information to user applications. In that way, the proposed simulation model for malicious behavior fulfills this requirement.

Although the performance of the proposed solution in a real environment may be different from the performance in a simulated scenario, the described simulation is close enough to a real deployment, being an acceptable first experimental validation. In particular, the most important and representative aspects of IoT systems are included in the proposed simulation (their heterogeneity and the high density of devices at their lowest level). These characteristics, in fact, must be replicated in a very precise way, as they are the most influencing variables in the proposal's performance. Actually, real IoT deployments, such as SmartSantander [36], present similar architectures, system heterogeneity, device density and employ (among other important aspects) equal wireless technologies to the presented simulation scenario.

Besides, in the proposed simulation scenario, we consider a final application containing only simple graphic algorithms calculating reputation values. This application was hosted in the ecosystem of the proposed scenario (based on FI-WARE European initiative [34]). Other four final applications (identical to the first one) were deployed to create the needed trust circle. Every final application in the simulation scenario was provided with the proposed hybrid reputation model and calculation method.

To be able to perform the described simulation, final applications must run in a different virtual machine from the one executing the simulation. Virtual machines must be embedded into the simulation scenario. To do that, different TAP bridges (or ghost nodes) [29] connecting virtual machines to the NS3 simulator were defined. Virtual machines were automatically generated and deployed through the Libvirt interface [37]. For all experiments, all machines implemented the operating system Linux Ubuntu 16.0.

The proposed simulation scheme employs the paravirtualization paradigm which allows a virtual machine (i.e. a NS3 node) to behave as an independent computer, providing all the configuration possibilities of a real machine. NS3 simulator provides support for the first three levels of this scheme. In that way, NS3 nodes can exchange messages with the real world and the host computer.

Using this scenario three different experiments were planned and performed. During the first experiment, the success rate on detecting malicious behaviors is evaluated. At the same time, the convergence speed is measured. During this experiment, parameters in the reputation model were fixed to the values indicated in Table 1.

Values in Table 1 have been selected to generate the results from the experimental validation, comparable to results obtained by previous proposals [12], [16], [19]. Other parameters (such as the ones related to the explicit reputation estimation algorithm) are configured to adapt to the average behavior of, for example, recommendation systems [39]. In future applications, different values for these parameters may be selected, according to the characteristics of the scenario.

In the second experiment, the success rate is evaluated, depending on the value of two parameters: the ratio in the geometric series defining the three parameters involved in implicit reputation calculation; and the token generation rate in the explicit reputation calculation algorithm.

Finally, during the third experiment, in order to evaluate the detection process, the proposed solution is compared to one of the existing proposals in the literature [16].

The proposed experiments are designed to, first, offer general results and proofs about the global performance of the proposed solution (i.e. the proposed contribution achieves its primary objective: detecting malicious components in IoT systems). In order to achieve these objectives first and second experiments are important. The first one evaluates if the proposed solution may detect malicious components successfully, and second one evaluates some quality parameters of its performance. Once the proposal is demonstrated to be a valid solution, then, its performance is compared to the most known and employed reputation models nowadays, in order to provide evidences that the proposed technique improves current state of the art. Third experiment is designed to support this objective.

TABLE 1. Configuration parameters during the experimental validation.

Parameter	Explanation	Value	Notes
μ_Λ	Reputation threshold to consider a component malicious (see final paragraph of Section III)	$\frac{1}{2}$	Threshold is placed in the medium point of the dynamic range
r_{neg}	Token generation rate for validating negative recommendations (see fifth paragraph of Section III.A)	$\frac{1}{3}\frac{\text{tokens}}{\text{h}}$	This value was fixed considering the speed of recommendation creation in social platforms
N_{neg}	Maximum capacity of token bucket for negative recommendations (see fifth paragraph of Section III.A)	15	Approximately, a burst of up to three recommendations per top application is validated (ignored)
r_{pos}	Token generation rate for validating positive recommendations (see fifth paragraph of Section III.A)	$\frac{1}{3}\frac{\text{tokens}}{\text{h}}$	This value was fixed considering the speed of recommendation creation in social platforms
N_{pos}	Maximum capacity of token bucket for positive recommendations (see fifth paragraph of Section III.A)	15	Approximately, a burst of up to three recommendations per top application is validated (ignored)
λ_Λ	Weight to aggregate all local reputation and obtain the global value (see expression 3)	$\frac{1}{5}$	All components in the system are weighted by the same factor
α, β, γ	Coefficients to calculate the implicit reputation from nobleness, solidarity and relevance (see expression 4)	$\frac{1}{3}$	All parameters are considered equally important in the implicit reputation calculation
N_{th}	Threshold to start the calculation process of nobleness and implicit reputation (see expression 5)	500	Approximately, data have to be collected during 15 minutes before running the implicit reputation calculation process
r	Common ratio to obtain the value of nobleness and implicit reputation (see expression 6)	$\frac{1}{2}$	Current observations and past observations have the same weight
μ_h	Threshold to consider a component hones or not (see expression 12 and next paragraph)	$\frac{1}{2}$	Threshold is placed in the medium point of the dynamic range
k	Common ratio to obtain the value of solidarity and implicit reputation (see expression 15)	$\frac{1}{2}$	Current observations and past observations have the same weight

V. RESULTS

In this section, results for the described experimental validation in Section IV are presented. The first subsection includes results for the security analysis, and the second subsection describes the obtained results for the performance evaluation.

A. SECURITY ANALYSIS

The first case to be studied is the unauthorized device insertion. Although the devices may be provided with valid credentials, the proposed solution is independent from the device configuration, permissions, type, etc., so this situation does not make easier the attack. On the other hand, these devices can never be essential for the system (as essential components will be directly deployed by the system administrator), so their importance is low. Besides, they are detected as untrustworthy devices, as they provide false information; and negative recommendations will grow as the number of authorized devices the malicious component interacts with (attacks will try to maximize this number to increase the damage to the system). In conclusion, both implicit and explicit reputation will decrease very fast, and the attack fails as inserted unauthorized devices would be isolated almost immediately.

With respect to the second scenario, data integrity, the proposed solution is independent from the causes that affect the correctness of data. In this case, data would be detected as untrustworthy by the nobleness calculation algorithm during the implicit reputation estimation. Besides, components communicating with the infected node generate negative recommendations and, after the convergence time, the component gets isolated for being malicious. The same behavior will be found if integrity is compromised by an increase in the packet loss rate in the electromagnetic interferences.

The third described attack, formally, is equivalent to the second one. However, in this case, the proposed technology behaves in a very intelligent manner. As reputation is calculated both, locally and globally, the replay attack (if only affecting one communication channel) will be stopped without removing the original and authorized component. In fact, once a component detects that false information is received though a communication channel, this one is being pruned. However, the source device is not isolated if globally the reputation is above the proposed threshold. In that way, components which related with the original components, but not with the malicious elements that impersonates it, may continue operating as regular.

With respect to the unauthorized data collection attack, all communications must be encrypted, as the convergence time required by trust-based and reputation-based techniques (as the proposed solution) enable attackers to collect a certain amount of information before stopping the attack. The proposed solution (as we are seeing in the next subsection) is faster than any previous proposal, but it still requires a certain convergence time. Components only collecting information are not solidary with other elements in the system and have no importance in the deployment. In that way, implicit reputation goes down, and as global reputation is obtained as the geometric average of explicit and implicit reputation, finally the component is detected as malicious.

Finally, the forward and backward security is guaranteed, as past events and information is weighted to reduce their importance (see expression (6)). In that way, although a malicious component tries to imitate an acceptable behavior in such a way it can inject false data but maintain a good reputation, as past events each time are less important, finally the reputation of these malicious elements falls below the security threshold.

In order to prove the resilience of the proposed solution to cyberattacks, Figure 8 shows the percentage of erroneous information and transactions in the IoT system at each time for the five relevant scenarios described above. In this work

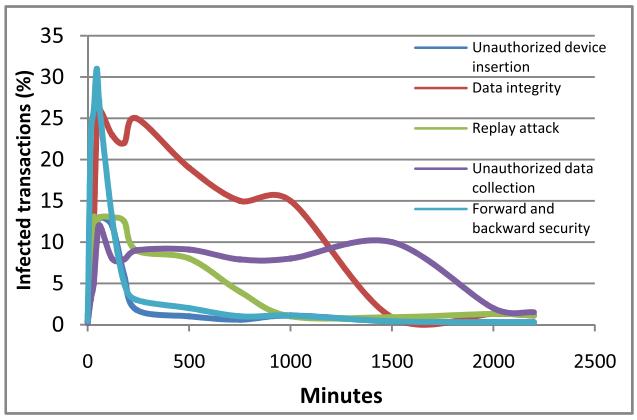

FIGURE 8. Resiliency to cyberattacks of the proposed framework.

we are considering a cyberattack is active if this percentage is above 2%. The employed configuration is the same as described in Section IV.B.

B. PERFORMACE EVALUATION

Each one of the described experiments was based on five simulations representing each one of them two hundred and forty (240) hours of operation.

Figure 9 shows the results of the first experiment. For the preparation of these results the concept of "success rate" was defined as the quotient between the number of times the proposed reputation model detected a real malicious component or behavior and the total number of times the algorithm was executed. As can be seen, the success rate is around 94%.

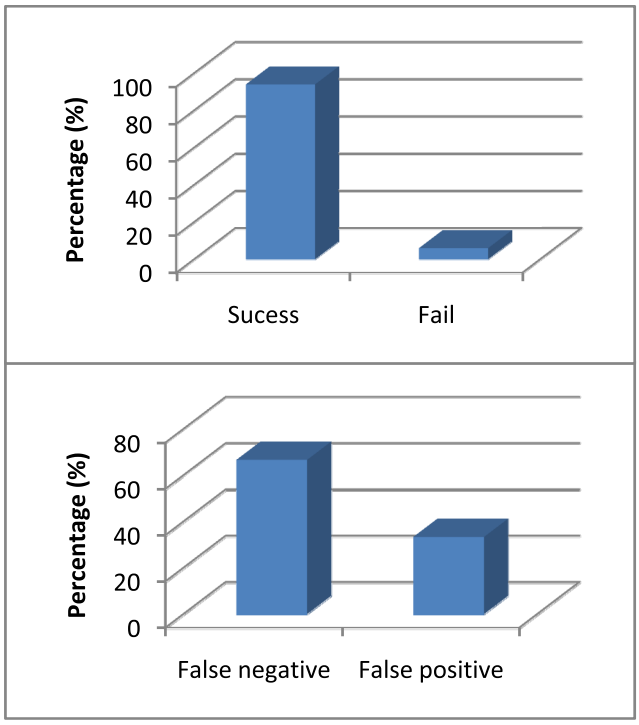

FIGURE 9. Results of the first experiment.

On the other hand, approximately in 6% of cases, the proposed algorithm fails. A study about the causes of these fails shows that most of them are due to fast attacks (i.e. components present a malicious behavior during a very limit amount of time, so the proposed solution cannot detect the situation fast enough). An explicit reputation model where a lower number of negative recommendations were admitted could be a possible solution. In fact, analyzing errors of the proposed algorithm shows that around 70% of fails are false negatives (corresponding to situation where too many negative recommendations were considered not relevant). False positives (around 30% of failures) usually correspond to situations where the statistical framework employed to calculate implicit reputation does not work properly. Particular causes and possible solutions should be analyzed in future works.

During this experiment, the convergence time has been also analyzed. Figure 10 shows that explicit reputation presents a much faster convergence time, as no information must be collected and accumulated (a reduction around 50% in the detection time is reached). A global reputation model based only on recommendations, however, is very vulnerable to all kinds of attacks [16], so implicit reputation must be considered in order to obtain a real picture about the system situation.

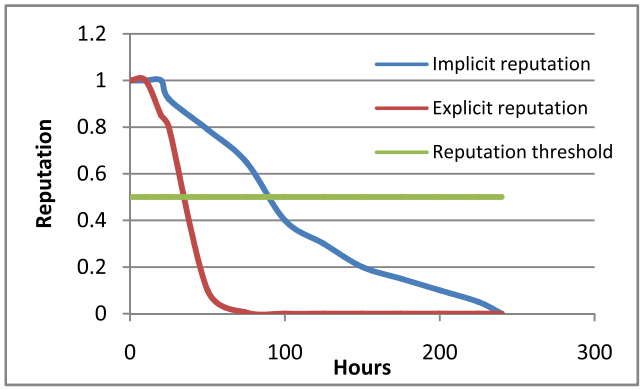

FIGURE 10. Study about the convergence time.

In our model, first, all components present a good reputation which is reduced if a malicious behavior is detected. The convergence speed would depend on the number of interactions the IoT components do, as well as the message creation speed (information is collected in few time). In order to compensate the effects of these variables, explicit reputation allows all the components in an IoT system to act in the same way as the component containing more information.

It is important to take into account that reputation, as defined in this work, is a normalized measure -see expression (2)-; however, as can be seen in Figure 10, implicit and explicit reputations are slightly over the unit (only some hundredths) in the first hours. This situation must be understood as a spurious effect caused by numerical errors in practical calculations.

Second experiment evaluates the influence of the two most critical parameters in our model in the convergence speed. Figure 10 shows the percentage of successfully performed detections depending on the ratio employed in the series to calculate the implicit reputation.

As can be seen, there is an optimum value for the common r around $r = 0,5$. In fact, for very low values of this parameter, past observations are practically negligible in comparison to the observations in the current time slot, and information is not considered in an adequate way. In a similar way, if past information has a very high influence (r is too high), new observations are almost not considered in the calculation process and the implicit reputation is not updated as desired. Consequently, see Figure 11, equilibrium between both information sources (past and present observations) must be reached in order to maximize the benefits of the proposed solution.

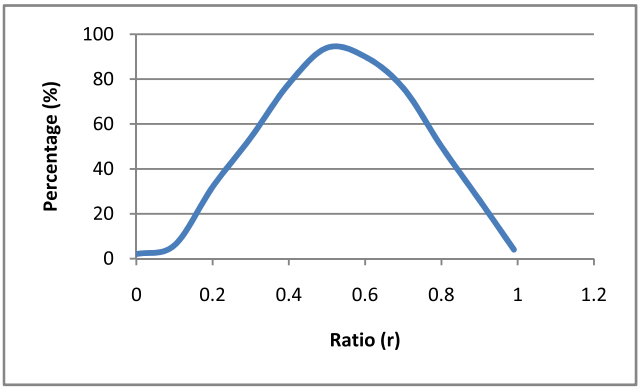

FIGURE 11. Results of the second experiment (study of the ratio in series).

Figure 12 shows the results of the study considering the token generation rate as independent variable. As can be seen, the ratio of successfully detected malicious components strongly depends on the common ratio but weakly depends on the token generation rate.

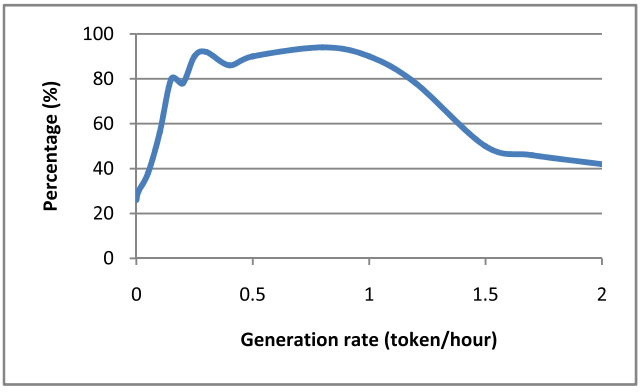

FIGURE 12. Results of the second experiment (study of token generation rate).

As can be seen, there is also an optimum value for the token generation rate (around 1 token per hour), but the graphic has a support on the value of 20% for the success rate. This phenomenon is due to the inclusion of two token buckets, one with positive tokens, and another one with negative tokens. Thus, if a decompensation or fluctuation appears in the system (causing all the components and elements to generate unprecise data, including recommendations) the explicit reputation algorithm is not affected. In fact, as two buckets are defined, their effects tend to cancel each other in this kind of situations (a conflict resolution procedure is triggered, as viewed in Section III.A). Finally, Figure 13 shows the results of the third experiment, where the proposed solution is compared to an existing proposal [16]. In particular, the work of Bao et al. is employed as reference, as it is the most relevant work about trust and reputation in IoT systems (see Section II).

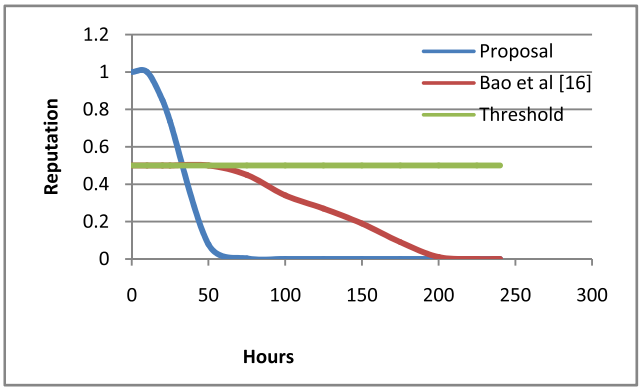

FIGURE 13. Results of the third experiment.

Both solutions behave in a similar way. However, if, in our proposal, all nodes are considered to have a good reputation by default, the considered previous proposal establishes the reputation threshold as default value. In that way, as previous proposals are based only on observations, the required time to detect a malicious behavior is higher. However, as the initial value is lower, the detection time (time required to obtain a reputation value for a malicious component below the trust threshold) is almost equal in both solutions. Nevertheless, as can be seen, an improvement close to 20% in the detection time is obtained by employing our solution.

VI. CONCLUSIONS

In this article, a hybrid reputation model based on both an implicit reputation calculation and an explicit definition of reputation is presented. Both amounts are considered in a geometric mean. Explicit reputation is obtained from explicit recommendations made by IoT components trusted by the module under study. Recommendations are processed through an algorithm based on the token bucket paradigm.

On the other hand, direct and indirect observations about external IoT components are employed to obtain an estimation of the implicit reputation. This second definition is calculated as the addition of three properties: solidarity,

nobleness and relevance. Using a statistical framework, these three parameters are estimated.

An experimental validation is also provided, evaluating the performance of the proposed solution. Obtained results show that the proposed technology is a valid approach to secure IoT systems, and prove the described algorithms are faster than similar previous exiting techniques. Results also show that the proposed solution must be complemented with other private information protection technologies, as the required convergence time by trust-based security policies allows the theft of a certain amount of information, until the components are identified as malicious. Besides, the proposed validation scenario only considers a quite permanent IoT deployment. It is important to consider that very dynamic systems might generate other results depending on the system change speed and the convergence time.

Future works should evaluate the performance of the proposed solution in very dynamic systems and collaborative environments, where devices composing the IoT deployment usually and randomly vary.

REFERENCES

[1] B. Bordel, R. Alcarria, T. Robles, and D. Martín, "Cyber–physical systems: Extending pervasive sensing from control theory to the Internet of Things," *Pervasive Mobile Comput.*, vol. 40, pp. 156–184, Sep. 2017.

[2] A. N. Sarkar, "Significance of smart cities in 21 st century: An international business perspective," *FOCUS, J. Int. Bus.*, vol. 2, no. 2, pp. 53–82, 2016.

[3] K. Hong, D. Lillethun, B. Ottenwälder, B. Koldehofe, and U. Ramachandran, "Mobile fog: A programming model for large-scale applications on the Internet of Things," in *Proc. 2nd ACM SIGCOMM Workshop Mobile Cloud Comput.*, 2013, pp. 15–20.

[4] B. B. Sánchez, R. Alcarria, D. Martín, and T. Robles, "TF4SM: A framework for developing traceability solutions in small manufacturing companies," *Sensors*, vol. 15, no. 11, pp. 29478–29510, 2015.

[5] R. Alcarria, T. Robles, A. M. Domínguez, and S. González-Miranda, "Flexible service composition based on bundle communication in OSGi," *KSII Trans. Internet Inf. Syst.*, vol. 6, no. 1, pp. 116–130, 2012, doi: 10.3837/tiis.2012.01.007.

[6] H. Lee, S.-K. Jo, N. Lee, and H.-W. Lee, "A method for co-existing heterogeneous IoT environments based on compressive sensing," in *Proc. IEEE 18th Int. Conf. Adv. Commun. Technol. (ICACT)*, Jan./Feb. 2016, pp. 206–209.

[7] B. Bordel, R. Alcarria, and D. Sánchez-de-Rivera, "Detecting malicious components in large-scale Internet-of-Things systems and architectures," in *Proc. World Conf. Inf. Syst. Technol.* Cham, Switzerland: Springer, 2017, pp. 155–165.

[8] T. Robles *et al.*, "An IoT based reference architecture for smart water management processes," *J. Wireless Mobile Netw., Ubiquitous Comput., Dependable Appl.*, vol. 6, no. 1, pp. 4–23, Mar. 2015.

[9] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for Internet of Things," *J. Netw. Comput. Appl.*, vol. 42, pp. 120–134, Jun. 2014.

[10] S. Geetha, "Social Internet of Things," *World Sci. News*, vol. 41, pp. 76–81, 2016.

[11] 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," *Comput. Sci. Inf. Syst.*, vol. 8, no. 4, pp. 1207–1228, 2011.

[12] F. Bao, I.-R. Chen, M. Chang, and J.-H. Cho, "Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection," *IEEE Trans. Netw. Service Manag.*, vol. 9, no. 2, pp. 169–183, Jun. 2012.

[13] M. Zhang, B. Selic, S. Ali, T. Yue, O. Okariz, and R. Norgren, "Understanding uncertainty in cyber-physical systems: A conceptual model," in *Modelling Foundations and Applications* (Lecture Notes in Computer Science), vol. 9764, A. Wąsowski and H. Lönn, Eds. Cham, Switzerland: Springer, 2016.

[14] C. C. Aggarwal, N. Ashish, and A. Sheth, "The Internet of Things: A survey from the data-centric perspective," in *Managing and Mining Sensor Data*. Boston, MA, USA: Springer, 2013, pp. 383–428.

[15] S. Hasan and E. Curry, "Approximate semantic matching of events for the Internet of Things," *ACM Trans. Internet Technol.*, vol. 14, no. 1, 2014, Art. no. 2.

[16] F. Bao and I.-R. Chen, "Trust management for the Internet of Things and its application to service composition," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2012, pp. 1–6.

[17] E. M. Daly and M. Haahr, "Social network analysis for information flow in disconnected delay-tolerant MANETs," *IEEE Trans. Mobile Comput.*, vol. 8, no. 5, pp. 606–621, May 2009.

[18] A. Boukercha, L. Xu, and K. EL-Khatib, "Trust-based security for wireless ad hoc and sensor networks," *Comput. Commun.*, vol. 30, nos. 11–12, pp. 2413–2427, 2007.

[19] F. Bao and I.-R. Chen, "Dynamic trust management for Internet of Things applications," in *Proc. ACM Int. Workshop Self-Aware Internet Things*, 2012, pp. 1–6.

[20] P. Resnick and R. Zeckhauser, "Trust among strangers in Internet transactions: Empirical analysis of eBay' s reputation system," in *The Economics of the Internet and E-Commerce*. Bingley, U.K.: Emerald Group Publishing, 2002, pp. 127–157.

[21] N. B. Truong, T.-W. Um, and G. M. Lee, "A reputation and knowledge based trust service platform for trustworthy social Internet of Things," in *Proc. Innov. Clouds, Internet Netw. (ICIN)*, Paris, France, 2016, pp. 1–8.

[22] H. Xiao, N. Sidhu, and B. Christianson, "Guarantor and reputation based trust model for social Internet of Things," in *Proc. IEEE Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Aug. 2015, pp. 600–605.

[23] L. Chen, Z. Yan, W. D. Zhang, and R. Kantola, "TruSMS: A trustworthy SMS spam control system based on trust management," *Future Generat. Comput. Syst.*, vol. 49, pp. 77–93, Aug. 2015.

[24] Y. Liu, Z. Chen, F. Xia, X. Lv, and F. Bu, "A trust model based on service classification in mobile services," in *Proc. IEEE Green Comput. Commun. (GreenCom)*, Dec. 2010, pp. 572–577.

[25] Y. Liu and K. Wang, "Trust control in heterogeneous networks for Internet of Things," in *Proc. Int. Conf. Comput. Appl. Syst. Modeling*, 2010, pp. V1-632–V1-636.

[26] R. Roman, P. Najera, and J. Lopez, "Securing the Internet of Things," *Computer*, vol. 44, no. 9, pp. 51–58, Sep. 2011.

[27] M. K. Khribi, M. Jemni, and O. Nasraoui, "Automatic recommendations for e-learning personalization based on Web usage mining techniques and information retrieval," in *Proc. 8th IEEE Int. Conf. Adv. Learn. Technol. (ICALT)*, Jul. 2008, pp. 241–245.

[28] J. Martinez-Romo and L. Araujo, "Updating broken Web links: An automatic recommendation system," *Inf. Process. Manage.*, vol. 48, no. 2, pp. 183–203, 2012.

[29] R. Alcarria, T. Robles, A. Morales, and E. Cedeño, "Resolving coordination challenges in distributed mobile service executions," *Int. J. Web Grid Services*, vol. 10, nos. 2–3, pp. 168–191, 2014.

[30] J. Heinanen and R. Guerin, *A Single Rate Three Color Marker*, document IETF RFC 2697, Sep. 1999.

[31] P. A. Stanwick and S. D. Stanwick, "CEO and ethical reputation: Visionary or mercenary?" *Manage. Decision*, vol. 41, no. 10, pp. 1050–1057, 2003.

[32] L. Gaines-Ross, *CEO Capital: A Guide to Building CEO Reputation and Company Success*. Hoboken, NJ, USA: Wiley, 2003.

[33] F. W. Scholz, "Maximum likelihood estimation," in *Encyclopedia of Statistical Sciences*, S. Kotz, C. B. Read, N. Balakrishnan, B. Vidakovic, and N. L. Johnson, Eds. New York, NY, USA: Wiley, 1985.

[34] T. Robles, S. González-Miranda, R. Alcarria, and A. Morales, "Web browser HTML5 enabled for FI services," in *Ubiquitous Computing and Ambient Intelligence*. Berlin, Germany: Springer, 2012, pp. 181–184.

[35] T. Zahariadis *et al.*, "FIWARE Lab: Managing resources and services in a cloud federation supporting future Internet applications," in *Proc. IEEE/ACM 7th Int. Conf. Utility Cloud Comput. (UCC)*, Dec. 2014, pp. 792–799.

[36] L. Sanchez *et al.*, "SmartSantander: IoT experimentation over a smart city testbed," *Comput. Netw.*, vol. 61, pp. 217–238, Mar. 2014.

[37] M. Bolte, M. Sievers, G. Birkenheuer, O. Niehörster, and A. Brinkmann, "Non-intrusive virtualization management using libvirt," in *Proc. Conf. Design, Automat. Test Eur.* Leuven, Belgium: European Design and Automation Association, 2010, pp. 574–579.

[38] Y. Moreno, M. Nekovee, and A. F. Pacheco, "Dynamics of rumor spreading in complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 6, p. 066130, 2004.

[39] Z.-D. Zhao and M.-S. Shang, "User-based collaborative-filtering recommendation algorithms on Hadoop," in *Proc. IEEE 3rd Int. Conf. Knowl. Discovery Data Mining (WKDD)*, Jan. 2010, pp. 478–481.
[40] Y. Y. Deng, C. L. Chen, W. J. Tsaur, Y. W. Tang, and J. H. Chen, "Internet of Things (IoT) based design of a secure and lightweight body area network (BAN) healthcare system," *Sensors*, vol. 17, no. 12, p. 2919, 2017.

BORJA BORDEL received the B.S. degree in telecommunication engineering and the M.S. degree in telecommunication engineering from the Universidad Politécnica de Madrid in 2012 and 2014, respectively, where he is currently pursuing the Ph.D. degree in telematics engineering with the Telecommunication Engineering School. His research interests include cyber-physical systems, wireless sensor networks, radio access technologies, communication protocols, and complex systems.

RAMÓN ALCARRIA received the M.S. and Ph.D. degrees in telecommunication engineering from the Universidad Politécnica de Madrid in 2008 and 2013, respectively. He is currently an Assistant Professor at the E.T.S.I Topography of the Universidad Politécnica de Madrid. He has been involved in several research and development European and national projects related to future Internet, Internet of Things, and service composition. His research interests are sensor networks, HCI, and prosumer environments.

DIEGO MARTÍN DE ANDRÉS received the B.Sc. degree in computer engineering and the M.S. degree in computer science from the Department of Informatics, Carlos III University of Madrid, Spain, and the Ph.D. degree in 2012. His main research areas are software process improvement, knowledge management and reutilization, and prosumer environments.

ILSUN YOU (SM'13) received the M.S. and Ph.D. degrees in computer science from Dankook University, Seoul, South Korea, in 1997 and 2002, respectively, and the Ph.D. degree from Kyushu University, Japan, in 2012. From 1997 to 2004, he was at THIN Multimedia, Internet Security, and Hanjo Engineering, as a Research Engineer. He is currently an Associate Professor with the Information Security Engineering Department, Soonchunhyang University. He is a fellow of IET.