Intelligent Automation And Soft Computing, 2019

Vol.25,no.1,155 –170

Trust provision in the Internet of Things using transversal blockchain networks

Borja Bordel[a], Ramon Alcarria[b], Diego Martín[a] and Álvaro Sánchez-Picot[a]

[a] Department of Telematics Systems Engineering. Universidad Politécnica de Madrid, Avenida Complutense nº 30. 28040 - Madrid (Spain)
[b] Department of Topographic Engineering and Cartography. Universidad Politécnica de Madrid. Campus Sur, 28031 Madrid (Spain)

ABSTRACT
The Internet-of-Things (IoT) paradigm faces new and genuine challenges and problems associated, mainly, with the ubiquitous access to the Internet, the huge number of devices involved and the heterogeneity of the components making up this new global network. In this context, protecting these systems against cyberattacks and cybercrimes has turn into a basic issue. In relation to this topic, most proposed solutions in the literature are focused on security; however other aspects have to be considered (such as privacy or trust). Therefore, in this paper we define a theoretical framework for trust in IoT scenarios, including a mathematical formalization and a discussion about the requirements which should fulfill a solution for trust provision. An analysis of these requirements shows that blockchain technology meets them perfectly, so a first trust provision system based on blockchain networks is also provided. An experimental validation is also proposed and performed in order to evaluate the described solution.

KEY WORDS: Blockchain, Cyber protection, Embedded Systems, Internet of Things, Trust provision

1 INTRODUCTION

NOWADAYS, society is really concern about the risks of networked devices. In fact, the number of protection solutions for computers, tablets and other traditional devices has grown in a very fast and incredible way in the last five years (and now it is one of the most promising markets) (NIST, 2016). However, new trends in the technological world present new additional challenges and problems which, in general, are invisible for the regular users (Bordel, 2017). One of these new technological systems which require special attention is the Internet of Things (IoT).

The IoT proposes to embed communication capabilities in every object, scenario or situation (from daily living objects such as appliances, to industrial systems and agriculture scenarios). As a main characteristic, this paradigm implies to be provided with a ubiquitous Internet access which makes complicated to apply traditional cyber-protection instruments such as traffic engineering or firewalls (Coley et al., 1998) (as the number of traffic sources and sinks may increase exponentially). Moreover, the current Internet architecture (proposed in the 90s and named as the "Internet-of-computers" (Coetzee et al., 2011)) is based on the connection of hundreds of millions of high capability machines, with power supply, which show a similar behavior and, even, a uniform hardware infrastructure. Nevertheless, each one of the new IoT scenarios is composed by thousands of heterogeneous devices, causing that, globally, various billions of devices are currently connected (expecting in the future to reach more than 50 billions) (Evans, 2011). Additionally, as we have said, these devices are very heterogeneous (from traditional computers to very small capability sensor nodes), so regular cyber-protection mechanisms based on the Internet-of-computers assumptions (total interoperability, employment of the OSI reference model, password-based designs, etc.) are not suitable at all. In fact, the Hewlett Packard company reported in 2015 that more than 70% of IoT devices present weaknesses despite using common cyber-protection solutions (HPE Fortify, 2016).

CONTACT Borja Bordel bbordel@dit.upm.es

In conclusion, the high level of heterogeneity and the limited computing power of devices, together with the large scale of the IoT systems (which generates severe scalability problems) and the great diversity of application scenarios (connecting humans, machines and robots in any combination) hinder the use of regular cyber-protection solutions. However, in order to get the full acceptance of users, companies and governments it is required to define valid cyber-protection policies. In particular, solutions related to three different aspects are needed: security, privacy and trust (Weber, 2010 and Feng et al., 2010). As can be seen on Figure 1(a), security and privacy terms include very well-known concepts such as authentication and encryption. Thus, most works on protection management for IoT deal with these topics, as they are consolidated knowledge (Atzori et al., 2010 and Domingo, 2012).

On the contrary, trust is a complex notion about which there is no consensus. Furthermore, important issues such as its definition, metrics or evaluation methodologies are rarely addressed (Sicari et al., 2015). Then, although most authors agree trust is a key element in the IoT scenarios, there is a lack of discussions about trust provision (apart from the legislative initiatives which do not address the technical challenges).

Therefore, the objective of this paper is to stablish a theoretical framework for trust in IoT scenarios, including its most important properties. The proposed framework is composed by a mathematical description of trust and a list of the requirements which should fulfill a solution for trust provision (including its assessment and evaluation). An analysis of the proposed formalization and requirements shows that blockchain technology (Pilkington, 2016) fits them perfectly, so a first trust provision system based on blockchain networks is also provided.

The rest of the paper is organized as follows. Section 2 describes the state of the art in trust provision systems and trust management solution for IoT scenarios. Section 3 presents the proposed formalization, the requirement analysis and the described solution based on blockchain technology. Section 4 includes the experimental validation. Finally Section 5 shows the obtained results and Section 6 concludes the paper.

2 STATE OF THE ART

ALTHOUGH works on trust in IoT systems are not very numerous, various proposals may be found (Xu, 2014). Moreover, many papers on collateral topics (such as multi-party computation (Shaikh et al., 2010) or privacy preservation (Evans et al., 2012)) have been also reported. Various surveys about trust have been also communicated (Daubert, 2015; Yan, 2014; Roman, 2011) which may be used to understand the current state and future challenges of trust management.

First, some authors have proposed works focused on trust evaluation (Zhao et al. 2016). In these proposals, methodologies in order to estimate the value of the properties influencing the trust level associated with the IoT entities in the system are defined. In general, however, it is complicated to evaluate trust in a quantitative way. Thus, solutions used to employ Quality-of-Service concepts in order to make calculations. However, some works focused on the estimation of parameters such as privacy, friendship, nobleness or sensitivity (Daubert, 2015; Bernabe 2016; Bordel 2017b) may be also found. Bao (2012), for example, defines honesty, cooperativeness and community-interest using network parameters such as the packet loss rate. These three quantities may be measured in a direct or indirect way, and are employed to compose a general estimation of trust in the system. Scalability and adaptability problems related to this solution were also studied (Bao et al., 2013). Advanced models considering adaptive parameters for trust calculation (Chen, 2016) have been reported as well. Furthermore, some trust properties (such as trust accuracy or resiliency) have been also investigated (Bao, 2012b; Chen, 2016b). As a main problem, these proposals require a long time (tens of hours) in order to converge to real value of trust; the proposed technology allows calculating trust in ephemeral ad hoc connections. In a similar way, Nitti (2012) proposes a reputation-based model for trust in the IoT, arguing that IoT entities might stablish social relationships in an independent way (therefore, he proposes the term Social Internet-of-Things). The proposed framework is able to detect the malicious behaviors and protect the system using security tools. On the other hand, trust evaluation models for simplified scenarios including only sensor nodes may be also found. For instance, Chen (2011) proposes a trust evaluation system for sensor networks based on a fuzzy reputation concept which considers QoS parameters. Finally, solutions including users' trust have been described (Liu et al. 2010). These proposals, using a service classification, evaluate the user perception in order to detect malicious entities.

In all the previously cited proposals, however, trust models and methodologies are focused on IoT entities (hereinafter, we are using the word "entity-centric"). In this approach, IoT components evaluate the trustworthiness of the other components with which they communicate. Depending on the elements employed to analyze the trustworthiness, different types of trust may be defined (see Figure 1(b)). In particular, cited works describe behavior-based (Junfeng et al. 2016; Chen et al. 2011b) solutions (where trust level depends on the past experiences), although in commercial solutions certificate-based solutions (such as HTTPS) are preferred.

In any case, some relevant problems still make difficult the practical employment of entity-centric solutions in IoT systems. In fact, entity-centric

solutions require very stable and static networks, as communication links should be maintained enough time to acquire a representative amount of information. However, most recently proposed IoT systems do not fit this characteristic. First, because IoT systems are very dynamic: hardware devices stablish ad hoc connections (Bordel et al. 2016b); software components may be deployed and disassembled very quickly, services and applications change depending on the users' needs (along the prosumer principles (Alcarria et al., 2012)), etc. Thus, relationships among components are very ephemeral and usually it is impossible to accumulate enough information in order to make the estimated trust value to converge to a stable value. Moreover, identity management is an unresolved issue in the IoT (Vermesan et al., 2011). In particular, it is unclear if components in a system can be provided with a unique identifier (Bandyopadhyay et al., 2011). Then, implementing an entity-centric trust evaluation mechanism turns a very complicated task. Finally, in a very common situation, services are provided by means of brokers which make independent the different layers in the IoT system. Furthermore, components in a certain layer usually execute services in a collaborative ad hoc way, and the execution scheme is unknown by the rest of the elements in the system (Bordel et al., 2016). That means that when an execution order is received by a component in an IoT system, it may delegate or work together with other components belonging to the same level in the architecture to solve that order; and no component in the system (even the original element whish sent the execution order) is able to know what components were involved in the execution or how the result was obtained. In conclusion, many times components cannot know the entities with which they communicate for real, so trust cannot be measured, even if indirect techniques are considered.

In order to address these problems, we propose a data-centric trust evaluation method. In our solution the trustworthiness of every received datum by a certain IoT entity is evaluated in an independent way, without being necessary to know all the components in the system or accumulate information about the past behaviors of the other IoT entities. In order to do that, we define the concept of "chain of custody" (CoC) of a datum, as well as the notion of "warranty level" associated with the CoC.

Other works are focused on the description of obligations and policies which IoT systems must implement in order to be trustworthy (Wu et al., 2011 and Dell'Amico et al. 2013). Languages such as WS-policy (Weerawarana et al. 2005) or XACML (Moses, 2005) are usually analyzed. The main problem of these languages is the difficulty to describe general low-level policies considering the huge differences from IoT scenario to IoT scenario (as target applications affect in a very strong way the type of hardware devices to be deployed in the system, and no common characteristics could be find between two arbitrary IoT systems).

In comparison to these proposals, the presented solution is based on a mathematical framework, which is adaptable to every application, system or scenario. Moreover, as the proposed framework is generic, it may be applied in the same way to both, low-level and high-level IoT entities.

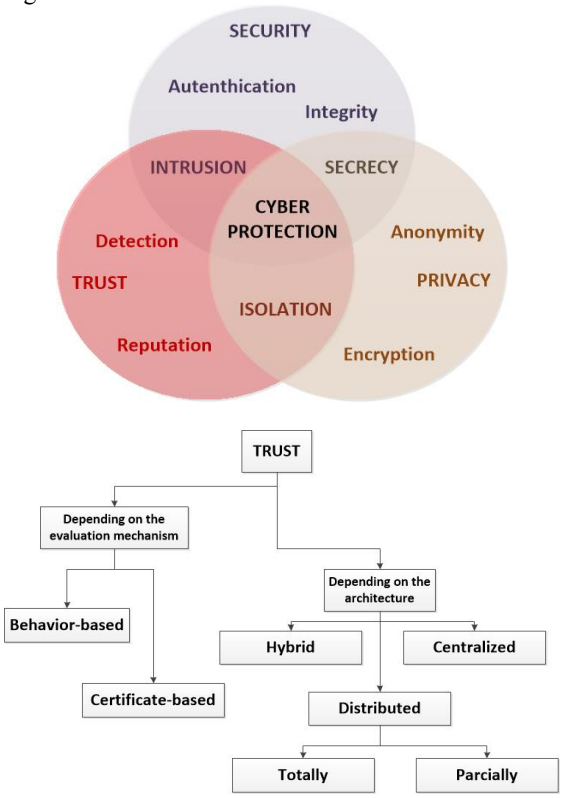


Figure 1. (a) Topics related to cyber-protection (b) Trust taxonomy

Finally, various architectures for IoT systems focused on improving trust management have been reported. Most of these proposals are based on the inclusion of special new functional components focused on trust evaluation (Suo et al., 2012). Xiong (2011), for example, proposes the inclusion of five different functional modules (such as a trusted user module or trusted network module), although the described solution it is not evaluated in practice. Other work (Zhou et al., 2012) describes one or various entirely new layer(s), such as middleware, focused on trust provision. The main problem of this approach is that only the components of the adjacent level may provide and obtain information about trust. Other solutions present modifications to existing layers (such as the network layer) in order to include trust provision. Thus, modified routing protocols (Dong et al., 2012), handover processes (Martinez-Julia et al.,

Table 1. Comparison among the trust provision and management systems

Article	Described solution	Applicable to different layers, applications…	Require a convergence time	Unique identifiers required	Deep modifications in the system architecture are needed	Ad hoc solutions are tolerated	Employ a validated technology
Bao (2012)	Trust model	With restrictions**	Yes	Yes	No	With restrictions	Yes
Nitti (2012)	Trust model	With restrictions	Yes	Yes	No	No	Yes
Chen (2011)	Trust model	With restrictions	Yes	Yes	No	With restrictions	Yes
Liu (2010)	Trust model	No	Yes	Yes	No	No	Yes
Description languages (Weerawarana et al. 2005 and Moses, 2005)	Trust description languages	No	No	Yes	No	No	Yes
Xiong (2011)	Trusted architecture	No	Yes	Yes	Yes	No	No
Zhou (2012)	Trusted architecture	No	Yes	Yes	Yes	With restrictions	No
Enhanced technologies (Dong et al., 2012; Martinez-Julia et al., 2013 and Liu et al., 2013)	Trusted enhanced technologies	No	Sometimes	Yes	Yes	No	Yes
Dólera (2014)	Adaptable trust architecture and models	With restrictions	Yes	Yes	Depending on the system architecture	With restrictions	Yes
Proposed solution	Transversal solution*	Yes	No	No	No	Yes	Yes

*It includes all the elements related to trust (model, description language, etc.) and affects every layer in the IoT systems
** Applicable only to layers and applications from a list or presenting certain characteristics

2013) and session establishment procedures (Liu et al., 2013) have been reported. Additionally, architectures including an adaptable trust model may be also found. In particular, Dólera (2014) proposes a trust provision solution including an engine selector being able to apply the most adequate trust calculation model at each moment. As a final idea, hardware technologies for trust management (Xu, 2014) have been also reported, although they are not very common as they require special hardware devices which are not commercial elements.

The main disadvantage of the previously described frameworks is the need of modifying the existing architectures (optimized to the application scenarios) in order to include the trust provision system. IoT deployments, in general, present complex schemes which cannot be easily modified, so these new architectures, usually, are not employed in practice. In order to address this challenge, the proposed solution is based on transversal blockchain networks which do not require modifying the existing architectures; have been exhaustively validated and allow providing trust information to every entity in the system.

Table 1 compares the proposed technology and the reviewed previous works.

3 PROPOSED SOLUTION

IN this Section the technical solution is described. In the first subsection the basic definitions and the mathematical formalization are presented. In the second subsection, requirements of trust provision systems in the IoT scenarios are reviewed and the proposed solution based on blockchain networks is described.

3.1 *Trust definition and mathematical formalization*

Mathematically, the reception of data by a certain IoT entity $\boldsymbol{e}$ may be modeled as a stochastic process $\boldsymbol{Y_e}$, which is the result of the processing tasks developed by a set of $\boldsymbol{K}$ different IoT entities $\boldsymbol{S_e} = \{\boldsymbol{s_1}, \boldsymbol{s_2}, \dots, \boldsymbol{s_K}\}$ called data sources. Data sources, besides, might consider additional inputs which (as any other data reception) may be also modeled as a collection of $\boldsymbol{M_X}$ stochastic processes $\boldsymbol{X_e} = \{\boldsymbol{X_1}, \boldsymbol{X_2}, \dots, \boldsymbol{X_{M_X}}\}$ (in the general case, $\boldsymbol{K} \neq \boldsymbol{M}$ as not every data source presents a unique input). Moreover, a generalized collection of $\boldsymbol{M_Z} = \boldsymbol{M_X} + \boldsymbol{M_W}$ data flows $\boldsymbol{Z_e} = \{\boldsymbol{Z_1}, \boldsymbol{Z_2}, \dots, \boldsymbol{Z_{M_Z}}\} = \{\boldsymbol{X_1}, \dots, \boldsymbol{X_{M_X}}, \boldsymbol{W_1}, \boldsymbol{W_2}, \dots, \boldsymbol{W_{M_W}}\}$ may be defined, not only considering the inputs to the data sources $\boldsymbol{X_e}$, but also the $\boldsymbol{M_W}$ intermediate states of the received datum $\boldsymbol{W_e} = \{\boldsymbol{W_1}, \boldsymbol{W_2}, \dots, \boldsymbol{W_{M_W}}\}$.

At each time instant $\boldsymbol{t} = \boldsymbol{t_i}$, the stochastic process $\boldsymbol{Y_e}$ turns into a random variable $\boldsymbol{Y_e^{t_i}}$ on the sample space $\boldsymbol{\Omega_Y}$ which contains all the possible received messages ($\boldsymbol{\Omega_Y}$ is, then, a discrete set). The same

consideration can be made about the set of stochastic processes $\boldsymbol{X_e}$.

Then, it can be defined a multiple variable function $\boldsymbol{F_e}$: $\left(\boldsymbol{\Omega_{X_1}} \times \boldsymbol{\Omega_{X_2}} \times \ldots \times \boldsymbol{\Omega_M} \times \mathbb{R}^+\right) \rightarrow \boldsymbol{\Omega_Y}$ (called processing function) which represents the composition and processing actions performed by the data sources, and depends on time in two ways: through the stochastic processes (which are time-dependent) and explicitly. Thus, $\boldsymbol{Y_e = F_e(X_1, X_2, \ldots, X_M, t)}$. In general, however, each one of the data sources performs a different action or set of actions, following an incremental process. Then, in general, $\boldsymbol{F_e}$ may be understood as the generalized composition of $\boldsymbol{N = K + J}$ functions $\boldsymbol{f_i}$ representing both, on the one hand the actions of each data source $\boldsymbol{h_i}$ (which adds a total of $\boldsymbol{K}$ functions) and, on the other hand, the potential malicious effects of cyber-attacks on the system operation $\boldsymbol{g_i}$ (represented by $\boldsymbol{J}$ functions) (1). Figure 2(a) shows a graphic representation of this scheme.

$$\boldsymbol{F_e = f_1 \circ f_2 \circ \ldots \circ f_N = h_1 \circ h_2 \circ \ldots \circ h_K \circ g_1 \circ g_2 \circ \ldots \circ g_J} \quad (1)$$

Definition 1 (Trust): Given a certain IoT entity $\boldsymbol{e}$ and a received message $\boldsymbol{\omega_Y} \in \boldsymbol{\Omega_Y}$ at $\boldsymbol{t = t_i}$, trust is the assumption by the entity $\boldsymbol{e}$ that the received datum $\boldsymbol{\omega_Y}$ comes from the processing and composition of legitimate data by legitimate data sources.

First, it is important to note that, in entity-centric trust definitions (Chen et al. 2011), trust is a continuous variable (which evolves as information about the behavior of other entities is acquired). However, the proposed data-centric definition does not consider past events or behaviors, so there is no converge time, but trust turns into a binary variable: an IoT entity either relies on the message and accepts it, or it does not and discard it.

On the other hand, traditional entity-centric trust definitions, which are usually behavior-based approaches, are focused on finding out if the processing function $\boldsymbol{F_e}$ hides a malicious behavior (Lize, 2014). Mathematically, these approaches try to decompose the processing function $\boldsymbol{F_e}$ into its elemental component functions $\{\boldsymbol{f_i \,/\, i = 1, \ldots, N}\}$, in order to stablish if any of them is affected by a cyber-attack (i.e. if there is a function $\boldsymbol{g_i \neq Id \; with \; i \in [1, \ldots, J]}$, being $\boldsymbol{Id}$ the identity function). However, as it is very complicated to decompose a function, so most times (in practice) entity-centric solutions only evaluate certain aspects of the global processing function. Considering the results, and using a previously defined decision tree, it is stablished if a hidden malicious component is present or not.

The proposed data-centric approach is quite different. Instead of evaluating the processing function (which may be a complex task), trust depends on the collection of data sources $\boldsymbol{S_e}$ and the generalized collection of data flows $\boldsymbol{Z_e}$. The obtained information about these elements, by only analyzing the received message, $\boldsymbol{\omega_Y}$, is usually insufficient (the conditional entropies $\boldsymbol{H(Z_i \,|\, Y_e = \omega_Y)}$, which measures the remaining uncertainty once known the received message, present values much higher than zero). Then, it is required additional information in order to determine if a datum $\boldsymbol{\omega_Y}$ is trustworthy, i.e. if it is the product of processing legitimate data by legitimate data sources (a question which, one more time, has a binary answer). Legitimate data flows $\boldsymbol{Z_e^{leg}}$ are usually a subset of the generalized data flows $\boldsymbol{Z_e^{leg} \subseteq Z_e}$, as well as legitimate data sources $\boldsymbol{S_e^{leg}}$ are a subset of the general data sources $\boldsymbol{S_e^{leg} \subseteq S_e}$. Basically, if these sets are equal two by two, then the datum $\boldsymbol{\omega_Y}$ is trustworthy. The advantage of this approach with respect to entity-centric solutions lies in the fact that it is very difficult to trace the processing functions of the IoT entities (some complex algorithms are required). It is always easier to store information about the data sources and the data flows proving a certain datum is trustworthy. In this context, it will be guaranteed a message is the product of the processing and combination of legitimate data by legitimate data sources, if it is verified the chain of custody (CoC) of that message.

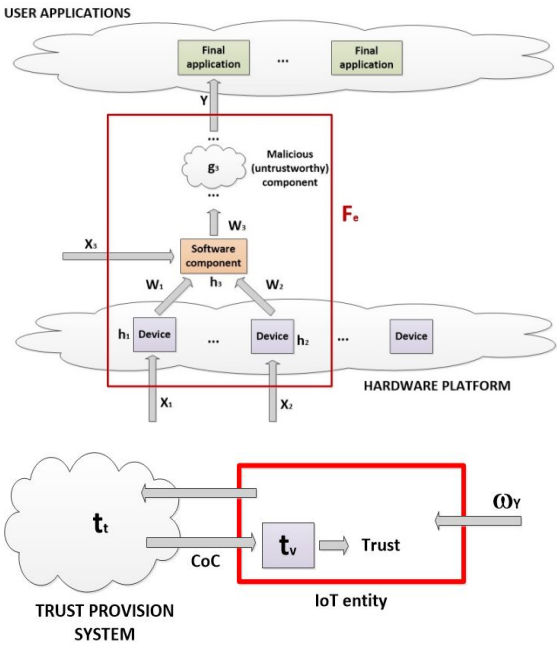


Figure 2. (a): General representation of the reception of data by IoT entities (b) Evaluation of the trust function in a generic scenario

Definition 2 (Chain of Custody -CoC- of a datum): The CoC of a datum is a registration of its lifecycle, designed in order to guarantee the received information has not suffered alterations, substitutions, contaminations or destructions. In particular, verifying

the CoC of a datum implies to access and validate the information about four basic phases in the data lifecycle: (1) the generation of the original low-level data, (2) the possible storage of the data until they are consulted or employed, (3) the transmission of the data to other IoT entities, and (4) the analyses and transformations applied to the data.

The way in which the information about the CoC is generated and stored depends on the employed technology: low-level elements usually employ binary data formats, while intelligent components manage complex information representations such as XML or semantic files. Nevertheless, it is usually registered by the same IoT entities which generate, process or compose the datum (i.e. the data sources). These entities, then, must be provided with the needed credentials to be able to access to the storing system.

The concept of CoC is employed in various contexts, from magic spectacles to legal investigations. However, the amount of required information to validate a CoC is different depending on the scenario. This idea is also valid in IoT systems. For instance, in some applications identifying the IoT entities transmitting a datum may be enough to create a valid CoC; while in other cases, it may be also necessary to register the time, the data length or any other relevant information. In any case, no system is able to provide a total knowledge about the received data (or the IoT entities, if an entity-centric solution is considered). Thus, it always exists a certain probability of trusting in an untrustworthy datum, $\boldsymbol{p_{err} > 0}$.

Definition 3 (Warranty level): The warranty level is a number (usually an integer) representing the amount of information (parameters), the level of detail and/or the granularity required to a CoC to be valid. In practice, it represents the "suspicion level" of the IoT entities: as the warranty level grows, more warranties (proofs) are required by the entities to trust in data. As a consequence, as the warranty level grows, the probability $\boldsymbol{p_{err}}$ goes down.

In general, different applications or services in a same IoT system would show different warranty levels: critical services would require a great warranty level, while trivial application could present low values for this parameter. Thus, an IoT entity may discard a received message (i.e. the entity does not trust on the message because the associated CoC cannot be verified) either because the required warranty level is not reached, or because the received information about the CoC shows that the datum it is not trustworthy.

All the previous discussions, besides, may be expressed mathematically. It can be defined, then, the trust function $\boldsymbol{T_e}: \boldsymbol{\Omega_Y} \rightarrow \mathbb{Z}_2$ of a certain IoT entity $\boldsymbol{e}$, which collects the information about the CoC of each received datum and verifies it. The obtained binary output determines if the data is trustworthy or not. In particular, the trust function may be understood as the composition of two functions (2).

$$\boldsymbol{T_e = t_v \circ t_t = t_v(t_t(\cdot))} \quad (2)$$

The first function is the tracking function $\boldsymbol{t_t}: \boldsymbol{\Omega_Y} \rightarrow \mathbb{R}^{\boldsymbol{p(w)}}$. For each received datum $\boldsymbol{\omega_Y}$ it generates a $\boldsymbol{p(w)}$-dimensional real vector, being $\boldsymbol{p}: \mathbb{N} \rightarrow \mathbb{N}$ an integer monotonically increasing positive defined function and $\boldsymbol{w}$ the warranty level required by the entity $\boldsymbol{e}$. This vector represents the information about the CoC of the message $\boldsymbol{\omega_Y}$. In general, as the CoC may include a lot of information, it will be stored in specialized external components. The tracking function, then, searches this information and acquires it. Moreover, as $\boldsymbol{w}$ grows, more information must be acquired which is coherent with the proposed definition. If enough information to construct the $\boldsymbol{p(w)}$-dimensional vector is no available, as we said, the trust function $\boldsymbol{T_e}$ returns a negative result immediately.

The second function is the verification function $\boldsymbol{t_v}: \mathbb{R}^{\boldsymbol{w}} \rightarrow \mathbb{Z}_2$. This function is performed by the IoT entities. It receives the real vector representing the information about the CoC of the datum $\boldsymbol{\omega_Y}$, and tries to verify it, determining if the datum is trustworthy or not. Any desired condition may be imposed, such as the original message to be generated by an authorized device or the timestamp to be later than a certain value. The selected policy depends on the considered service and may be described using any of the available languages (XACML, for example). Figure 2(b) represents the generic evaluation of the trust function.

Imaging a trivial solution, a database could support the described framework. However, a hidden practical problem must be considered, which makes our proposal very different from usual databases or data traceability schemes. As we said, IoT entities cannot store the information about the CoC of every datum in the system (as they are not prepared to support this functionality), so a specialized external component is needed. However, in this context, the IoT entity $\boldsymbol{e}$ has to receive (through the tracking function) a new message containing the information about each CoC. Thus, in a recursive problem, it should be evaluated if the received information about the CoC is trustworthy.

In order to solve this complex problem, two solutions could be designed. The first solution implies to deploy a chain of trust (CoT). In this scheme, it is supposed that components storing the information about the CoC are much more secure and trustworthy by default, so IoT entities should require a lower warranty level to the messages received from them. In that way, the components storing the information about the CoC of the original CoC are much more trustworthy, so IoT entities require a much lower warranty level, etc. At the end, it will be found a totally trustworthy component by default (so no

warranties are required, the trust function always returns a positive result), which validates the entire chain.

Mathematically, every component in an IoT system has the same probability of transmitting an untrustworthy message, $\boldsymbol{p_0}$ (assuming that all have the same level of security and, therefore, the same probability of suffering a cyber-attack). The creation of a CoT implies to incorporate components whose a priori probability of transmitting an untrustworthy message is lower. This probability is continuously going down with each iteration (a total of $\boldsymbol{n}$ steps are considered) until it cancels (3).

$$\boldsymbol{p_0 > p_1 > \cdots > p_n > 0} \tag{3}$$

This solution is very common, with modifications it is (for example) the base of the HTTPS protocol. However, it is a very time consuming mechanism as various tracking actions (each time more complicated as components as more secure) and verifications must be performed. Moreover, no component is totally secure or trustworthy by default, so (actually) there is a hidden risk when supposing the last component validates the entire CoT (quantified in the probability $\boldsymbol{p_{err}}$). Therefore, a second option is preferred.

Definition 4 (cumulative property of trust): Given a certain message $\boldsymbol{\omega_Y}$received by a certain IoT entity $\boldsymbol{e}$, the trust of the entity $\boldsymbol{e}$ on the message $\boldsymbol{\omega_Y}$grows following an exponential law, as statistically independent evaluations of the trust function with positive result are cumulated. As the physical independence implies statistical independence, in practice in order to cumulate trust, various independent registrations of the CoC should be maintained (thus, in each evaluation of the trust function, the information about the CoC will be tracked in a different register).

In fact, mathematically, if every component in an IoT system has the same probability of transmitting an untrustworthy message, $\boldsymbol{p_0}$, then the probability of a message to be untrustworthy when $\boldsymbol{n}$ independent registrations of the CoC prove it is trustworthy may be described as an exponential function (4).

$$\boldsymbol{p_U = \prod_{i=1}^{n} p_0 = p_0^n} \tag{4}$$

Then, the second solution consists of acquiring in an independent way the information about the CoC of the received datum, from a collection of $\boldsymbol{n}$ independent registers $\boldsymbol{\mathcal{R}_e = \{r_1, r_2, \ldots, r_n\}}$. The intuitive idea behind this proposal is that if a same information is provided by various independent sources, the probability of it to be true is higher (a principle very used, for example, in journalism). Described in that way, this second solution also requires performing various tracking processes (as in the first solution), however two important facts may be considered. First, as all registers are independent but equally secure, the accessing time is the same in every case (so, finally, for a certain number of tracking phases, this second option requires less time). And, second and much more important, as all registers implement the same interfaces, protocols and secure policies, they can communicate and the tasks of storing various independent copies of the CoC and (later) checking the consistence among all of them may be delegated on such collection of registers (which must be implemented with an appropriate technology). In that way, the IoT entities only have to perform one evaluation of the trust function and the entire process accelerates. As a consequence of this delegation, usually, the value of the $\boldsymbol{n}$ parameter is part of the system design and it is the same for every IoT entity, service or application.

3.2 Trust provision using blockchain networks

Various works (Yan et al., 2014) have studied the objectives and requirements of trust provision systems. Considering these previous analyses and the framework presented in Section 3.1, we obtain the following list of requirements for data-centric trust provision systems:

- REQ#1, Generality: Data-centric trust provision systems must be generic, to be easily and widely applied to any type of service or application supported by the IoT infrastructure.
- REQ#2, Trust in metadata: Trust provision systems should be also applicable to control information in the IoT deployment (such as QoS data) if required.
- REQ#3, Self-protection: Trust provision systems should effectively detect attacks against their infrastructure, especially attempts to modify the stored information about the CoC.
- REQ#4, Privacy preservation: Information about the CoC cannot contain any data about the users' identity, personal information, etc.
- REQ#5, Distributed: The trust provision system must be made of a collection of distributed independent nodes, being able of storing each one a copy (complete or partial, but coherent with the others) of the CoC of the data in the IoT system.
- REQ#6, Storage capacity: The trust provision system may be able to store all the information about the CoC of the data in the system, with the required granularity (warranty level).

An analysis of the previously described requirements clearly shows that blockchain technology (Pilkington, 2016) is the most adequate to implement data-centric trust provision systems. The objective of this paper is not to explain in detail how blockchain networks work; however, a brief overview is provided below in order to show all requirements are perfectly met.

Blockchain networks (Tapscott, 2016) are composed by a collection of nodes, maintaining each one a copy (partial or complete) of the information blocks which are stored in the network. The objective of these networks is to maintain trustworthy information, divided into chained blocks which are distributed among all the independent nodes which made up the network. In that way, REQ#6 is natively supported. On the other hand, any new block added to the network is, by default, sent by the receiver node to other nodes to be stored in various independent locations (REQ#5). Blockchain networks, besides, are agnostic in respect to the content of the blocks (which, even, may be heterogeneous). Thus, they can store information (CoC) about both data and metadata (REQ#2) and about any type of service or application (low-level devices and top-level applications may incorporate information to the blockchain network in the same way). REQ#1 gets, in that way, perfectly met. Additionally, each block is signed by means of a hash function, which protects the stored content. If any illegal change is applied, the block gets corrupted and every posterior block chained with it also gets invalid. If the hash fields were recalculated to create valid blocks, the affected node would consult the copies of the modified blocks maintained in other nodes. If the modified blocks are not coherent with the information stored in the network (at least $\boldsymbol{n}$ nodes must confirm it), the changes are discarded (so information about CoC is always supported by, at least, $\boldsymbol{n}$ independent registrations, as required by the cumulative property of trust). Furthermore, only authorized users (provided with the appropriate key) may incorporate information to the blockchain network. In that way, REQ#3 is fulfilled. Finally, blockchain networks are not a backup system, so only information about the CoC of data in the IoT system is stored (never the proper data). Personal information or identities, then, are not maintained and REQ#4 is also met.

Blockchain technology, besides, presents a good behavior in relation to other important variables and challenges such as the security in the securing system, reliability, availability or investment. In particular, as a network made of several peers where information is extensively replicated, availability is guaranteed in blockchain systems. Besides, peers may be implemented using standard software techniques and regular hardware equipment, so investment has not to be very high. Reliability and security in the securing system are guaranteed by the architecture of blockchain systems. As information it is not supported only by one machine, but by several hosts geographically sparse and belonging to very different people, the solution is highly reliable by default, and security is provided by means of traditional techniques as in any other computational system.

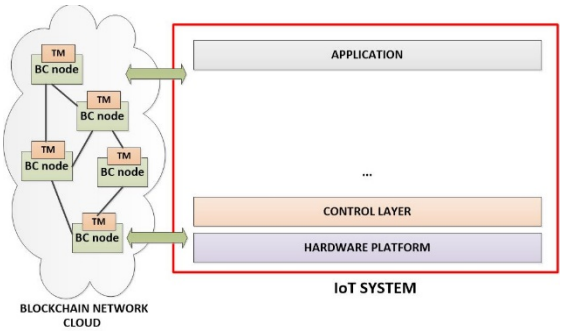


Figure 3. Trust provision system based on blockchain technology

Figure 3 presents the proposed trust provision system. As can be seen, the basic element in the system is a transversal (or vertical) blockchain network, which communicates with every layer in the IoT system (and, potentially, with every IoT entity), but which remains totally independent from the IoT deployment.

The proposed blockchain network is made of a collection of blockchain nodes (BC node, in Figure 3) which are provided with a tracking module (TM) being able to collect all the information about the CoC of data from the blocks stored in the network, when requested by IoT entities. Traditionally, these nodes are connected using secure traditional internet techniques such as HTTPS, TCP, TLS, etc. Each BC node has associated a certain set of IoT entities. These entities are provided with the credentials which allow them to register and obtain information in/from the network through the associated BC node (each node may have different credentials, in order to improve the node independence). The use of these credentials ensures that only legitimate IoT components write and read the CoC.

Two different actions may be performed by IoT entities in respect to the blockchain-based trust provision system: write information about the CoC and obtain the CoC of a datum.

In order to write information about the CoC of data in the blockchain network, IoT entities periodically send a report about the activities performed in the last time slot. The level of detail of the report, as well as the format of this document, depends on the IoT entity writing the information. In some occasions data will be uniquely identified (by means of a transaction identifier, the transport sequence number, etc.), while in other cases generic notes will be written (for example: “at T time, algorithm A was applied to all pending data”). If desired, both elements (level of detail and data format) can be coordinated in all entities, layers, etc. or can be totally independent in each component. In some cases, even, only some selected entities (for example the border entities such as gateways) write information in the blockchain network. The only requirement which must be taken into account is that the tracking module (TM) must be

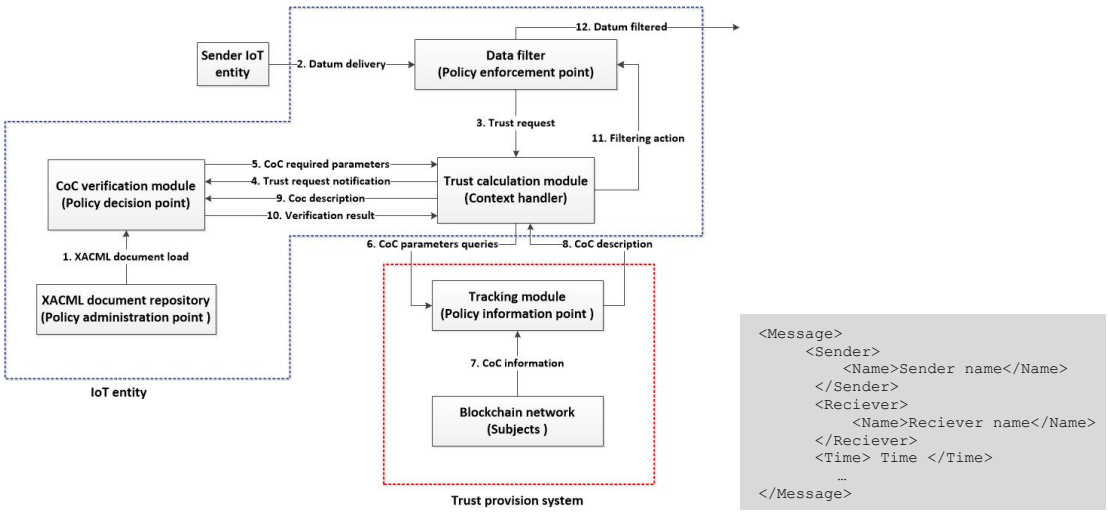


Figure 4. **(a)Trust provision system based on blockchain technology (b) Example of a XML document in the proposed solution**

able to understand all the data formats employed in the data blocks. In the proposed implementation (see Section 4) a XML-based homogeneous data format is employed. Reports from authorized IoT entities are directly encapsulated into data blocks and stored in the network (see Figure 4(b)).

Reading process is slightly more complicated. As each time, and depending on the warranty level, IoT entities may need different parameters (and, in general, a different amount of information) about the CoC, it is required a flexible tracking solution. This solution is based on XACML language. IoT entities needing trust information about data (usually top-level applications as they have more computational power, although other entities could also use this service) maintain a XACML description about the parameters of the CoC (including, of course, the four basic phases, see Definition 2) which want to be known (origin, generation time…). Then, a XACML modified data-flow model is followed in order to obtain the required information and verify the calculated CoC (see Figure 4(a)).

On Figure 4(a), all the elements participating in the trust evaluation process are identified. In brackets, generic names as indicated in the XACML data-flow model are written. As main name, the denomination employed in this work is included. The process is as described below:

1. The XACML document describing the required parameters and information in a valid CoC is created by users and loaded by the IoT entity.
2. A new datum or message is received by the IoT entity. Its processing is stopped in a *data filter* until it is determined if the received information is trustworthy.
3. The *data filter* requests the *trust calculation module* about the trust associated with the received datum
4. *Trust calculation module* asks to the *CoC verification module* for the required information about CoC in order to make a decision about the new message
5. *CoC verification module* sends the parameters to be obtained from the trust provision system.
6. *Trust calculation module* sends a request to the corresponding BC node and its associated *tracking module*.
7. The blockchain network reads the information blocks, tracking the origin of the received datum and all the transformation process it suffered.
8. The *tracking module* creates a description of the CoC with the obtained information from the blockchain network
9. The CoC description is verified by the *CoC verification module*. As we are explaining later, two basic types of verification could be made: a light one and a heavy one. In both cases, policies and rules to be applied to the CoC are also described in the XACML document created by users.
10. The result of the verification is returned
11. If, finally, the datum is trustworthy the processing keeps going, if not, the *data filter* it is ordered to delete the message
12. The filtering decision is applied

Although this process seems to be very time consuming, in fact it requires very few resources. Even, blockchain networks may operate at real-time, as their use to support BitCoin transactions has proved (Yelowitz et al., 2015). However, as we said, an additional mechanism to accelerate the calculations is planned. Most times, applications trust on data if their CoC may be tracked with the required warranty level: particular values of parameters are not important. In fact, if the CoC of a message may be perfectly traced with the information contained in the blockchain network, it is guaranteed the received information has

not suffered alterations, substitutions, contaminations or destructions (see Definition 2). In these circumstances, it can be employed the light verification. In this type of verification, tracking modules do not send a complete report about the CoC. If all the required parameters can be obtained, a message indicating a positive result is sent, but the complete CoC description is discarded. At the end, this type of verification implies every IoT entity in the system provided with blockchain credentials is a legitimate data source. This approach reduces the transmission and processing time in the IoT entity, which accelerates the entire process (quantitative analyses are performed in Section 4 and 5).

On the other hand, if additional rules or policies are defined (for example, every datum obtained from hardware devices in a certain geographic area is untrustworthy), the heavy verification must be executed. In this case, the entire CoC description is evaluated in the *CoC verification module* in order to determine if it fulfills the requirements to consider the message trustworthy.

Finally, it is important to note that traditional blockchain networks are implemented using physical devices (computers). However, nowadays, virtualization techniques allow the creation of more dynamic and flexible solutions by means of the called Network Functions Virtualization (NFV). Therefore, in the proposed trust provision system, a transversal blockchain network is implemented using cloud computing and virtualization techniques.

4 EXPERIMENTAL VALIDATION

AN experimental validation was designed and conducted in order to evaluate the performing of the proposed solution. In particular, two different experiences were developed. During the first part of the validation, the characteristics and capacities of the proposed technology are exhaustively measured. During the second part, the proposed data-centric trust solution is compared with traditional entity-centric proposals, in a common application scenario.

Using the Cloud Services of the Technical University of Madrid, a virtual blockchain network was created and deployed. Each node was provided with a tracking module, being able of understanding XML language. Although information about CoC may be stored in heterogeneous formats, for simplicity (and as this fact does not affect significantly the obtained results) in this first deployment all the IoT entities are generating reports using the XML language. Ten independent virtual nodes were deployed, using OpenStack as management application. The credentials of each BC node consisted of a hash, generated from a private key. Credentials were directly programmed on IoT entities, although in real application solution for the secure distribution of keys should be considered. SHA-256 hash algorithm was employed for both, generating the credentials and signing the data blocks. Nodes in the proposed blockchain network were based on Linux (Ubuntu systems) with Intel i5 processors and 4GB of RAM.

On the other hand, an IoT system was deployed in a laboratory of the Technical University of Madrid. The deployed infrastructure consisted of an autonomous system for the dynamic calculation of evacuation plans. Various sensor nodes where deployed, being connect among them through a publication/subscription system (P/S) and to the Internet by means of a collection of concentrators. LCD displays and sound actuator were also included. This entire physical infrastructure was connected with top-level application by means of a multi-modal interface, including (among other technologies) a web interface and a telemetry interface. A rule provision module was also included in order to stablish certain policies about the evacuation plans. Figure 5 shows the described architecture. A detailed description of the behavior of this system was reported by Morales (2014).As a novelty, all the entities in the system were provided with a new interface to communicate with the blockchain network. Each sixty (60) seconds every entity generated an XML report describing the activity of the last minute. Using the provided credentials, these reports were generated in the trust provision system. An XACML description document is also provided to top-level applications. The content of these documents varied depending on the performed experiment.

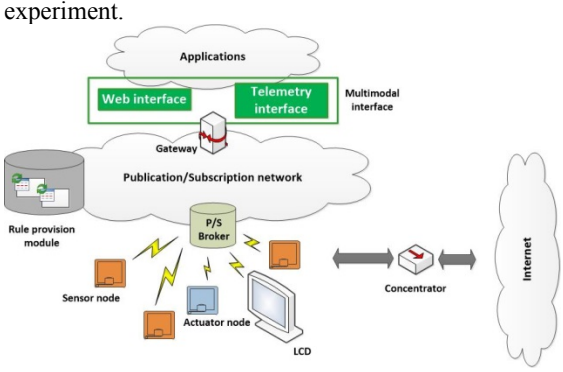


Figure 5. Architecture of the deployed IoT infrastructure

During the first part of the experimental validation, five different experiments were conducted.

The first experiment evaluated the percentage of malicious messages correctly detected, depending on the number of data generated per minute in the IoT system. In order to do that, light verification was activated, and two different policies about the CoC registration were considered. Firstly, every IoT entity was able to register the generated XML reports in the blockchain network. Later, only border components (brokers, concentrators and gateways) were allowed to write information about the CoC. In order to develop this experiment, 25% of the total number of generated messages were untrustworthy.

The second experiment, in the same circumstances than the first one, evaluated the percentage of malicious messages correctly detected, depending on the percentage of untrustworthy messages generated. A data generation rate of 50 data per minute was considered during this experiment.

The third experiment was focused on determining the influence of the XACML description document in the performance of the trust provision system. In the same circumstances than the first one, considering all entities in the IoT system may add information to the blockchain network, two different XACML were created. The first one only required to the information about the CoC to be found (i.e. light verification was activated). The second one was designed to only admit as trustworthy data generated from messages created by certain sensor nodes (i.e. heavy verification has to be performed). Information about the percentage of malicious messages correctly detected was acquired.

The fourth experiment is an extension of the third one (exactly the same conditions are considered). It is focused on evaluating the reduction in the trust calculation time when light verification is activated.

Finally, the fifth experiment evaluates the percentage of malicious messages correctly detected, depending on the warranty level required by top-level applications. For simplicity, each time the warranty level grew in one unit; ten new parameters were included as mandatory information for any CoC in the XACML document. Light verification was activated and every IoT entity in the system was able to incorporate information in the blockchain network.

On the other hand, during the second part of the experimental validation, only one experiment was performed. One the most famous entity-centric proposals (Bao et al. 2012 and Bao et al., 2013) was also implemented in the same IoT infrastructure. Light verification was activated and every IoT entity in the system was able to incorporate information in the blockchain network. In these conditions, both solutions were compared in various situations. Namely:

a) Situation A: A fixed component presents a malicious component
b) Situation B: An ad hoc connected component presents a malicious component
c) Situation C: New components without support for trust evaluation are included in the IoT system

5 RESULTS

RESULTS of the experimental validation are showed on Figure 6 to 9. Figure 6 to 8 present the results of the first part of the experimental validation, while Figure 9 describes the results of the second part.

The evolution of the percentage of malicious messages correctly detected, depending on the number of data generated per minute may be seen on Figure 6. Four different curves are drawn. "False trustworthy" curves represent the messages labeled as trustworthy, which, in fact, are untrustworthy. "False untrustworthy" curves represent the opposite, untrustworthy messages considered as trustworthy. As can be seen, all curves have an exponential-like evolution. In three situations, the proposed solution behaves in a very similar way. As the number of generated data per minute grows, more information (CoC) is stored in the blockchain network, and the tracking and verification algorithms have more problems in order to calculate the CoC (errors can be committed, such as not finding a certain information or confusing the value of a field). However, when all entities may incorporate information to be trust provision system, the asyntactic value for the error probability is, as maximum, $\boldsymbol{p} \approx \mathbf{10}\%$. A value that is very similar to which reached by the "false trustworthy" curve when only border entities may incorporate information. The value which, in fact, changes its behavior in a very drastic way is the "false untrustworthy" probability when only border entities create the CoC. In fact, as only partial information is available, sometimes all parameters cannot be found. Then verification algorithm tends to determine as untrustworthy valid data.

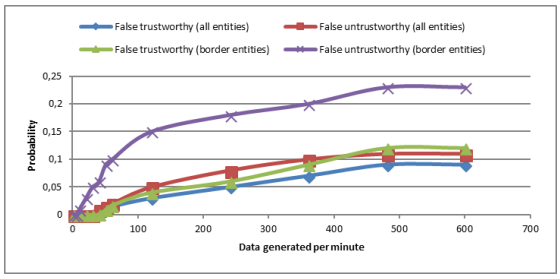


Figure 6. Results of the first experiment (first part of the experimental validation)

A similar situation occurs in experiment two (see Figure 7(a)). In general, as the number of messages generated per minutes was relative low (50 data per minute), error probabilities (calculated following the Laplace's definition, i.e. as a ratio over the total number of transmitted messages) are also low (all of them blow 2%). A slightly evolution, nevertheless, can be observed: "false trustworthy" curves tend to be growing and "false untrustworthy" curves are decreasing (in a more significant way). Once more time, however, the most remarkable evolution is which presented by the "false untrustworthy" curve, when only border entities may incorporate information to be blockchain network. As can be seen, error probability is near 10% if no untrustworthy messages are generated, but it starts descending very fast form the beginnings (at a rate of, more or less, 25%). This is due to the fact that, as the number of invalid messages grows, more often the lack of information represents a real untrustworthy message.

The third experiment shows very similar results to which showed for experiment one (see Figure 7(b)). "False trustworthy" curve for light verification remains below the value of $\boldsymbol{p} \approx \mathbf{10}\%$. On the other hand, the other three curves present a slight modification. This effect is especially significant when heavy verification is considered. As more conditions are applied, it is more probable to not be able to locate all the required parameters and messages are labeled as untrustworthy. Thus, heavy verification must be considered very carefully, and probably is only advisable in general applications. On the other critical services might incorporate this type of verification as, as can be seen, the "false trustworthy" curve converges to a lower error probability (around $\boldsymbol{p} \approx \mathbf{7}\%$).

Previously discussions about heavy verifications are much more remarkable considering the result of experiment four (see Figure 8(a)). As can be seen, as more data per minute are generated, more time is required to evaluate trust. However, in its maximum value, a reduction of 40% in the calculation time can be observed. It must be considered that, in these experiments "normalized time" was obtained dividing each value per the maximum obtained value during the experimental validation.

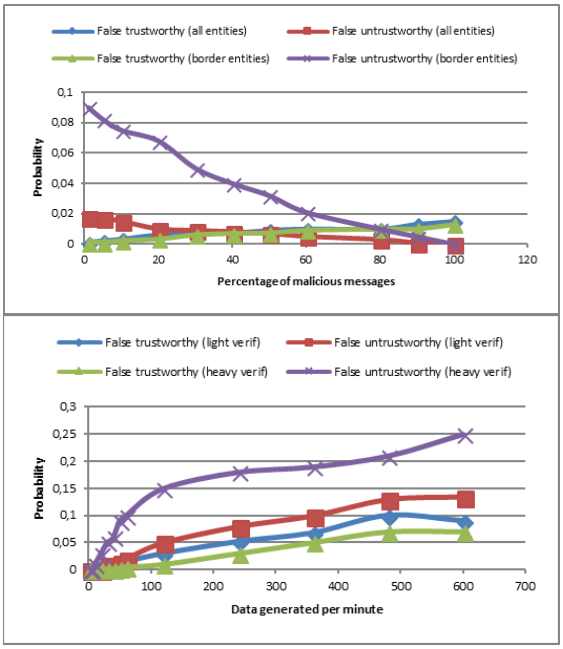


Figure 7. (a) Results of the second experiment (first part of the experimental validation) (b) Results of the third experiment (first part of the experimental validation)

Moreover, the evolution is quite different depending on if heavy or light verification is considered. In the case of light verification, a linear evolution may be observed. As CoC descriptions are not transmitted to the CoC verification module, no saturation state is reached and, as seen, the evolution is linear. On the other hand, heavy verification requires a complex process of calculation; thus as the number of data to be processed grows, some trust estimations could not be performed (especially if queues on IoT entities are saturated) but time remains constant. As conclusion, heavy verification should be only employed in circumstances when the required level of protection justifies these worse QoS parameters.

Experiment five, showed that there is an optimum warranty level (see Figure 8(b)). As can be seen for low values of the warranty level, CoC contains few information and the probability of labeling an untrustworthy message as trustworthy is very high. On the other hand, for high values of the warranty level, probability of occurring a "false untrustworthy" grows (in a similar phenomenon to which described in the case of heavy verification). As a result, there is an optimum warranty level, located in the point where both curves cross. Depending on the characteristics of the IoT and trust provision systems this value may change, but in the case of the proposed validation it is $\boldsymbol{w} = \mathbf{8}$ (warranty level always must be integer).

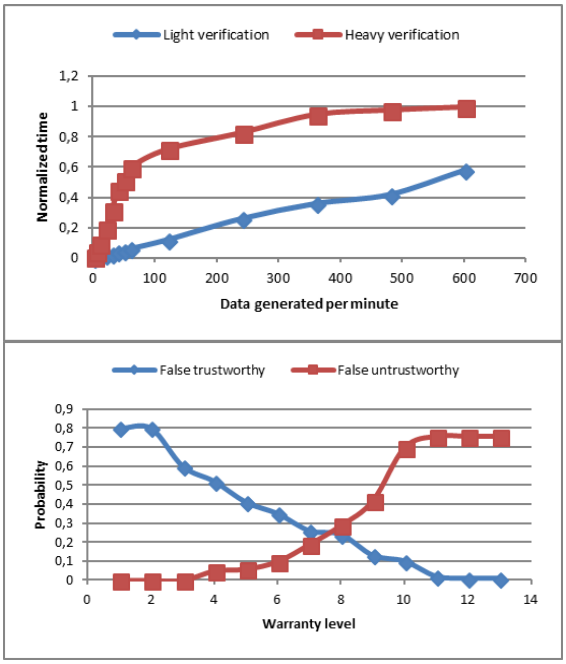


Figure 8. (a) Results of the fourth experiment (first part of the experimental validation) (b) Results of the fifth experiment (first part of the experimental validation)

Finally, we are reviewing the results of the second part of the experimental validation, where the proposed solution is compared with previous entity-centric proposals (Bao et al., 2011). See Figure 9. It is very difficult to compare data-centric proposals with entity-centric proposals, as different very parameters and flow diagrams describe their behavior. Thus, the number of malicious data finally accepted in three different remarkable situations is the best way of

comparing (even if only partially and limited) both types of solutions.

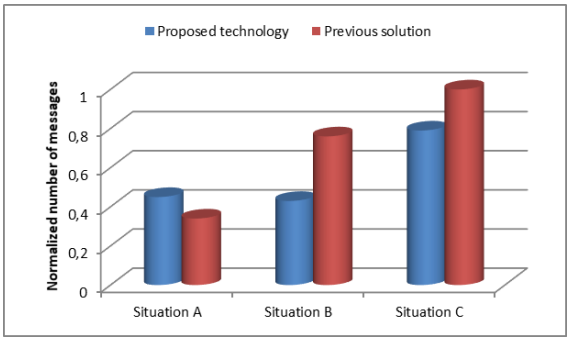


Figure 9. Results of the second part of the experimental validation

As can be seen, traditional entity-centric trust provision systems are more effective than the proposed data-centric technology if malicious components maintain a fixed position. In fact, in these situations, as once the convergence time has passed and the malicious component has been isolated no more messages are admitted so, long-term, fewer malicious messages are processed. However, in the other two situations, where components do not maintain a same situation or identity, as entity-centric solutions need a certain converge time before detecting any malicious conduct, there is a risk of never reaching a stable value (and, the, all messages, trustworthy or untrustworthy, are admitted). Thus, data-centric solutions behave in a better way in dynamic scenarios.

6 CONCLUSIONS

IN this paper a solution for trust evaluation in IoT scenarios is described. The proposed data-centric solution is based on a mathematical formalization and the concepts of Chain of Custody and Warranty level. Moreover, it is focused on dynamic scenarios where IoT entities stablish ad hoc connections and/or identities are not permanent (i.e. ephemeral configurations).

The practical implementation of the proposed solution is based on the blockchain technology as it meets both, the described formalization and the usual requirements for trust provision systems.

Basically, our proposal consists of a blockchain network, where meta-information about the received data is stored. This information is protected by hash functions and divided into chained data blocks which are maintained in various independent nodes in order to protect the system against cyber-attacks. This information is employed to create CoC descriptions which are used to determine the trustworthiness of the received data.

The experimental validation showed that the proposed scheme is a useful solution, with a maximum error probability (in a regular scenario) of p ≈10%. A heavy verification algorithm is also provided, which is valid for highly protected infrastructures, but whose characteristics do not advise to employ it extensively. The concept of warranty level is also evaluated, stablishing there is an optimum value for this parameter. Additionally, a comparison between the proposed data-centric solution and traditional entity-centric proposals determines the main use of the proposed technology is associated to ephemeral scenarios, as entity-centric solutions present a better behavior in fixed infrastructure. Then, as a conclusion, both proposals are not contradictory but complementary.

In any case, the proposed technology enables the calculation of trust without being necessary to monitor the communication links or IoT entities during large time periods, as in previous entity-centric proposals. In that way, new and most recent IoT systems based on ad hoc technologies and ephemeral connections could be provided with trust management solutions.

7 DISCLOSURE STATEMENT

NO potential conflict of interest was reported by the authors

8 FUNDING

THE research leading to these results has received funding from the Ministry of Economy and Competitiveness through SEMOLA project (TEC2015-68284-R) and from the Autonomous Region of Madrid through MOSI-AGIL-CM project (grant P2013/ICE-3019, co-funded by EU Structural Funds FSE and FEDER). Borja Bordel has received funding from the Ministry of Education through the FPU program (grant number FPU15/03977)

9 REFERENCES

R. Alcarria, Robles, T., Morales, A., López-de-Ipiña, D., & Aguilera, U. (2012). Enabling flexible and continuous capability invocation in mobile prosumer environments. *Sensors*, 12(7), 8930-8954.

L. Atzori, Iera A., Morabito G., (2010) The internet of things: a survey, *Computer Networks*. 54 (15) 2787–2805.

D. Bandyopadhyay & Sen, J. (2011). Internet of things: Applications and challenges in technology and standardization. *Wireless Personal Communications*, 58(1), 49-69.

F. Bao, Chen, R., Chang, M., & Cho, J. H. (2011, June). Trust-based intrusion detection in wireless sensor networks. *Proceedings of 2011 IEEE International Conference on Communications* (ICC), 1-6

F. Bao and Chen I. (2012) Trust management for the Internet of Things and its application to service composition. *Proceedings of the IEEE*

International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM) 1–6.

F. Bao & Chen, I. R. (2012, September). Dynamic trust management for internet of things applications. In Proceedings of the 2012 international workshop on Self-aware internet of things (pp. 1-6). ACM.

F. Bao, Chen I., Guo J. (2013) Scalable, adaptive and survivable trust management for community of interest based Internet of Things systems. *Proceedings of the IEEE eleventh International Symposium on Autonomous Decentralized Systems (ISADS)* 1–7

J. B. Bernabe, Ramos, J. L. H., & Gomez, A. F. S. (2016). TACIoT: multidimensional trust-aware access control system for the Internet of Things. Soft Computing, 20(5), 1763-1779.

B. Bordel Sánchez, Alcarria R., Sánchez-de-Rivera D., and Sánchez-Picot A. (2016) Enhancing Process Control in Industry 4.0 Scenarios using Cyber-Physical Systems. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, 7(4), 41-64

B. Bordel, de Rivera, D. S., & Alcarria, R. (2016, July). Plug-and-play transducers in Cyber-Physical Systems for device-driven applications. In *10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, 2016 (pp. 316-321). IEEE.

B. Bordel, Alcarria, R., Robles, T., & Martín, D. (2017). Cyber–physical systems: Extending pervasive sensing from control theory to the Internet of Things. *Pervasive and Mobile Computing*, 40, 156-184.

B. Bordel, Alcarria, R., & Sánchez-de-Rivera, D. (2017, April). Detecting Malicious Components in Large-Scale Internet-of-Things Systems and Architectures. In *World Conference on Information Systems and Technologies* (pp. 155-165). Springer, Cham.

D. Chen, Chang G, Sun D, Li J, Jia J, Wang X. (2011) TRM-IoT: a trust management model based on fuzzy reputation for Internet of Things. *Computer Science and Information Systems*. 8 (4):1207–28.

H. Chen, & Zhongchuan, F. (2011). A novel trust routing scheme based on node behaviour evaluation for mobile AD hoc networks. *Intelligent Automation & Soft Computing*, 17(8), 1063-1074.

R. Chen, Guo, J., & Bao, F. (2016). Trust management for SOA-based IoT and its application to service composition. *IEEE Transactions on Services Computing*, 9(3), 482-495.

R. Chen, Bao, F., & Guo, J. (2016). Trust-based service management for social internet of things systems. *IEEE Transactions on Dependable and Secure Computing*, 13(6), 684-696.

L. Coetzee, & Eksteen, J. (2011, May). The Internet of Things-promise for the future? An introduction. *2011 IST-Africa Conference Proceedings*, Gaborone, 2011, pp. 1-9.

C. D. Coley & Wesinger Jr, R. E. (1998). U.S. Patent No. 5,826,014. Washington, DC: U.S. Patent and Trademark Office.

J. Daubert, Wiesmaier, A., & Kikiras, P. (2015, June). A view on privacy & trust in IoT. In *2015 IEEE International Conference on Communication Workshop (ICCW)*, (pp. 2665-2670). IEEE.

M. Dell'Amico, Serme M.I.S.G., de Oliveira A. S., Roudier Y. (2013), Hipolds: a hierarchical security policy language for distributed systems, *Information Security Technical Report*. 17 (3) 81–92.

G, Dólera Tormo, Marmol F.G., Perez G.M. (2014), Dynamic and flexible selection of a reputation mechanism for heterogeneous environments, *Future Generation Computer Systems*.

M. C. Domingo. (2012), An overview of the internet of underwater things, *Journal of Network and Computer Applications* 35 (6) 1879–1890.

P. Dong, Guan J., Xue X., Wang H. (2012), Attack-resistant trust management model based on beta function for distributed routing in internet of things, *China Communications*, 9 (4) 89–98.

D. Evans. (April 2011). "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything". Cisco.

D. Evans, Eyers D.M. (2012). Efficient data tagging for managing privacy in the Internet of Things. *Proceedings of the IEEE international conference on green computing and communications (GreenCom)* 244–8.

H. Feng, Fu W., (2010) Study of recent development about privacy and security of the internet of things, *Proceedings International Conference on Web Information Systems and Mining (WISM)* 91–95.

HPE Fortify and the Internet of Things, homepage. http://go.saas.hpe.com/fod/internet-of-things (Accessed 31 December 2016)

T. Junfeng & Hongqiang, J. (2016). A kind of dynamic software behavior trust model based on improved subjective logic. *Intelligent Automation & Soft Computing*, 22(4), 621-629.

M. Liu, Zhang N. (2010) A solution to privacy-preserving two-party sign test on vertically partitioned data (P22NSTv) using data disguising techniques. *Proceedings of the International Conference on Networking and Information Technology (ICNIT)* 526–34.

T. Liu, Guan Y., Yan Y., Liu L., Deng Q. (2013), A WSN-oriented key agreement protocol in internet of things, *Proceedings of 3rd International Conference on Frontiers of Manufacturing Science and Measuring Technology*, 1792–1795.

G. Lize, Jingpei, W., & Bin, S. (2014). Trust management mechanism for Internet of Things. *China Communications*, 11(2), 148-156.

P. Martinez-Julia, Skarmeta A. F. (2013), Beyond the separation of identifier and locator: building an identity-based overlay network architecture for the future internet, *Computer Networks* 57 (10), 2280–2300.

A. Morales, Alcarria, R., Martin, D., & Robles, T. (2014). Enhancing evacuation plans with a situation awareness system based on end-user knowledge provision. *Sensors*, 14(6), 11153-11178.

T. Moses. (2005). Extensible access control markup language (xacml) version 2.0. Oasis Standard, 200502.

M. Nitti, Girau R., Atzori L., Iera A., Morabito G., (2012) A subjective model for trustworthiness evaluation in the social internet of things. *Proceedings of IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications*. Australia, Sydney 18–23.

NIST Special Publication 800-160 (November 2016). Accessible online: http://csrc.nist.gov/publications/drafts/800-160/sp800_160_second-draft.pdf (Accessed 31 December 2016)

M. Pilkington. (2016). Blockchain technology: principles and applications. *Research Handbook on Digital Transformations*, edited by F. Xavier Olleros and Majlinda Zhegu. Edward Elgar.

R. Roman, Najera, P., & Lopez, J. (2011). Securing the internet of things. *Computer*, 44(9), 51-58.

Z. Shaikh, Mishra DK. (2010) A study on secure multiparty computation problems and their relevance. *Proceedings of the second international conference on Computational Intelligence, Modelling and Simulation (CIMSiM)* 95–9.

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

H. Suo, Wan J, Zou C, Liu J. (2012) Security in the Internet of things: a review. *Proceedings of the International Conference on Computer Science and Electronics Engineering (ICCSEE)*, 3, 648–51.

D. Tapscott & Tapscott, A. (2016). Blockchain Revolution: How the technology behind Bitcoin is changing money, business, and the world. Penguin.

O. Vermesan, Friess, P., Guillemin, P., Gusmeroli, S., Sundmaeker, H., Bassi, A., ... & Doody, P. (2011). Internet of things strategic research roadmap. 1, 9-52.

R. H. Weber. (2010) Internet of things - new security and privacy challenges, *Computer Law & Security Review* 26 (1) 23–30.

S. Weerawarana, Curbera, F., Leymann, F., Storey, T., & Ferguson, D. F. (2005). Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more. Prentice Hall PTR.

Z. Wu, Wang L., (2011) An innovative simulation environment for crossdomain policy enforcement, *Simulation Modelling Practice and Theory* 19 (7), 1558–1583.

I. Xiong, Zhou X, Liu W. (2011) Research on the architecture of trusted security system based on the Internet of Things, *Proceedings of the International Conference on Intelligent Computation Technology and Automation (ICICTA)*, 2, 1172–5.

T. Xu, Wendt, J. B., & Potkonjak, M. (2014, November). Security of IoT systems: Design challenges and opportunities. In *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design* (pp. 417-423). IEEE Press.

Z. Yan, Zhang, P., & Vasilakos, A. V. (2014). A survey on trust management for Internet of Things. *Journal of network and computer applications*, 42, 120-134.

A. Yelowitz & Wilson, M. (2015). Characteristics of Bitcoin users: an analysis of Google search data. *Applied Economics Letters*, 22(13), 1030-1036.

B. Zhao, He, J., Zhang, Y., Liu, G., Zhai, P., Huang, N., & Liu, R. (2016). Dynamic trust evaluation in open networks. *Intelligent Automation & Soft Computing*, 22(4), 631-638.

Q. Zhou, Gui F, Xiao D, Tang Y. (2012), Trusted architecture for farmland wireless sensor networks. *Proceedings of the IEEE 4th international conference on cloud computing technology and science (CloudCom)*, 782–7.

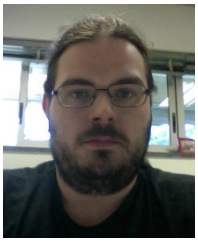
10 NOTES ON CONTRIBUTORS

Borja Bordel received the M.S. telecommunication engineering in 2014 from Technical University of Madrid. He is currently pursuing the Ph.D. degree in telematics engineering at Telecommunication Engineering School, UPM. His research interests include cyber-physical systems, communication protocols and complex systems.

Ramon Alcarria received his M.S. and Ph.D. degrees in Telecommunication Engineering from the Technical University of Madrid in 2008 and 2013 respectively. Currently, he is an assistant professor at the E.T.S.I

Topography of the Technical University of Madrid. His research interests are Service Architectures, Sensor Networks and Prosumer Environments.

Diego Martín received his doctoral degree in 2012, holds a B.Sc in Computer Engineering and an M.S. in Computer Science from the Department of Informatics at the Carlos III University of Madrid, Spain. His main research areas are Software Process Improvement, Knowledge Management and Reutilization and Prosumer Environments.



Alvaro Sanchez-Picot received his M.S. degree in Telecommunication Engineering from Technical University of Madrid in 2014. Currently he is a Ph.D. student in the Department of Telematics Systems Engineering. His research interest is focused on Sensor Networks, Simulation of Network Communications, Wireless Communications and Web development.