A Decentralized Blockchain-Based Trust Management Protocol for the Internet of Things

Djamel Eddine Kouicem, Youcef Imine, Abdelmadjid Bouabdallah, Hicham Lakhlef

A Decentralized Trust Management Based Blockchain Protocol for Internet of Everything

Djamel Eddine Kouicem, Youcef Imine, Abdelmadjid Bouabdallah, Hicham Lakhlef

Abstract—Internet of Everything (IoE) is a network that integrates a variety of heterogeneous nodes, such as connected devices (sensors, robots, smart phones ...), connected cars, smart home appliances, etc. These smart objects communicate and collaborate between each other in a distributed and dynamic environments which are facing several security challenges. Trust management is one of the most important challenges in such environments. Existing trust management solutions do not fit with the new requirements introduced in IoE such as heterogeneity, mobility and scalability. In this paper, we propose a hierarchical and scalable blockchain based trust management protocol with mobility support in massively distributed IoE systems. In our protocol, smart objects disseminate trust information about service providers to the blockchain. Thus, all the objects will have a global view on each service provider in the architecture which speeds up the trust decision making. In addition, our system is resilient against the known malicious attacks such as bad-mouthing, ballot-stuffing and cooperative attacks. We confirm the efficiency of our proposal through theoretical analysis and extensive simulations. Finally, we show that our protocol outperforms existing solutions especially in terms of scalability, mobility support, communication and computation.

Index Terms—Trust management; Internet of Everything; Blockchain; Fog computing; Distributed IoE systems

1 Introduction

These last years, we are witnessing a real digital revolution of Internet that is becoming an Internet of Things (IoT) where huge number of physical objects are being connected to Internet. By 2020, the Gartner Institute expects more than 50 billion connected objects on the market, which will radically change our lifestyles through many applications [1]. Recently, a new paradigm called Internet of Everything (IoE) has been introduced by Cisco as an extension of IoT. This technology introduces a new heterogeneous and massively distributed network of people, smart objects, data and processes, which makes Internet smarter [25].

In order to efficiently manage the huge number of IoT objects and data in IoE environments, a new architecture called fog computing has been introduced recently. This architecture aims to extend cloud-computing services to the edge of the network. This extension is realized by using a large number of edge components such as routers, base stations, gateways, etc. Therefore, computation, communication, storage and control operations are performed closer to end users by pooling edge network's local resources.

IoE can be viewed as service centric architecture where each device, or thing in general, can request services from other devices and it may also provide services for other devices (service provider). Service centric based IoE applications face several security challenges such as trust management. Indeed, IoT service providers may behave dishonestly and maliciously for the purpose of promoting IoT devices (service requesters) to select them for one or many services on behalf of other trusted service providers. Furthermore, dishonest IoT service providers may perform discriminatory, bad-mouthing and ballot-stuffing attacks to disrupt the network and monopolize many provided services. Therefore, it is clear that a trust management protocol to evaluate the trustworthiness of IoT service providers, in a scalable and efficient way, is more than necessary.

To date, there is a large number of trust management protocols that have been developed for Wireless Sensor Networks, Social networks and P2P systems in general (eg. [3], [4], [6], [9], [20], [22], [24]). In these protocols, trust computation is often based on some information that includes: 1) the direct observations of each node regarding the others (which is gathered whenever the node encounters the IoT service providers) and 2) the indirect recommendations received from other nodes against the service providers. These solutions are still not scalable when it comes to massively distributed systems such as IoE. Indeed, in most solutions, a node needs to communicate with a large number of IoT devices so it would be able to accurately compute trust levels of IoT service providers. Moreover, other questions still arise on how trust information (direct observations and indirect recommendations) is disseminated and shared in a scalable way among different IoT objects in order to speed up the process of trust computation and make it more accurate. In addition, each node has to store this whole trust information about every encountered service provider.

Besides, in some cases an IoT device O_i needs to assess the trust level of a new encountering service provider Sp_j in a fast way, without necessarily performing a lot of exchanges. Thus, these solutions seem to be non convenient with such scenarios since without any previous exchange, a new encountering node Sp_j is assumed to have a trust value equal to 0.5, whereas it could be malicious.

Other clustering and centralized based trust management approaches have been investigated in several works

• *DE. Kouicem, Y. Imine, A. Bouabdallah, H. Lakhlef are with Sorbonne Universités, Université de Technologie de Compiègne, HEUDIASYC UMR 7253 CS 60319 60203 Compiègne Cedex France. E-mail: dkouicem, imineyou, bouabdal, hlakhlef@utc.fr*

Manuscript submitted April 24, 2018

(eg. [11], [13], [21]) in order to enhance the process of trust computation and optimization of IoT resources. Albeit these approaches allow for constrained IoT devices to assess trustworthiness of each other efficiently, devices have access only on trust data in their own cluster (no global view of trustworthiness). Furthermore, these protocols usually assume that cluster heads are pre-trusted nodes in terms of either provided trust information or behaviors. However, such assumption is not practical in most IoT applications.

Hence, this brings us back to an important question: how we can ensure a fully distributed and scalable trust management protocol, in which IoT devices can evaluate trustworthiness of any service provider in Internet, without the presence of any pre-trusted entity ?

2 Related work

In this section, we review some trust management protocols for IoT which are closely related to our work.

Very recently, Guo et al. [14] provided a comprehensive survey about the most recent works in the trust managements and computational trust models in IoT. They focused basically on service management in IoT dealing with the choice of IoT devices as service providers according to their trustworthiness. They discussed the five fundamental components of each trust management system, namely: trust composition, trust propagation, trust aggregation, trust update and trust formation.

Chen et al. [5] proposed a trust management model based on fuzzy reputation concept for IoT. However, they considered only some specific Wireless Sensor Networks (WSN) applications where nodes can establish limited trust relationships with other nodes. Actually, compared to WSN nodes, IoT devices are internet-enabled and can establish complex relationships with other IoT devices and owners.

Saied et al. [19] proposed a multi-service and context-aware trust management protocol for IoT systems, which deals efficiently with different malicious attacks. However, their protocol is based on centralized trusted servers that collect trustworthiness from IoT devices which is not viable in IoT. Similarly, Guo et al. [13] proposed a 3-tier hierarchical architecture based on cloudlets to disseminate trust information to a central cloud. Their architecture allows IoT devices to report trust information and also query trustworthiness of other devices directly from the local cloudlets. However, the proposed architecture refers always to the central cloud which is responsible to disseminate the trustworthiness information gathered from one cloudlet to the other cloudlets which can involve latency issues. Moreover, their trust model is still limited in the context of IoT, since distributed cloudlets are assumed to be honest in their architecture and they maintain only trust data in their geographical area.

The concept of social Internet of Things has been developed recently in many works. This concept consists on extending the world of IoT in such away, IoT devices will be able to establish autonomously social relationships between other devices and users. Many works have investigated the trust management problem in the context of social IoT [7], [15], [16], [18]. Chen et al. [7] proposed an adaptive trust management protocol for social and dynamic IoT systems. The main idea consists on distributing the computation of trust information among IoT devices. In their computational model, each device maintains its own trust assessment toward other users and devices. The trust assessment is based on the recommendations of the other devices, the direct observations and also the history of the interactions. The authors considered different classes of trust properties such as QoS, honesty and cooperativeness depending on the social relationships between IoT devices. However, their protocol is not scalable enough since each device must save all the trust pieces of information (that include its history and the recommendations of the other devices, etc.) related to its social friends (IoT devices and owners) in a lookup table. In [18], the authors proposed two trustworthiness computational models. 1) A subjective model which basically consists on the combination of the local trust parameters (direct observations) and also the received indirect recommendations. And 2) An objective model, where they proposed to disseminate trust assessments in a distributed Hash table maintained by a subset of trusted IoT devices. However, this last assumption is not actually practical in IoT environments. Moreover, their solution is still limited and it is applicable only in social based IoT applications. Other similar and recent works [2], [8], [12] have investigated the same problem by considering the same subjective trust management model as [7] which suffers from some scalability issues.

3 Our contributions

In this paper, we present a solution to the aforementioned limitations and address the above questions. We propose a new scalable trust management solution named ***BC-Trust***. Our solution is based on blockchain technology and fog computing paradigm, and allows IoT devices to accurately assess and share trust recommendations about other devices in a scalable way without referring to any pre-trusted entity. The blockchain is maintained by powerful fog nodes which offload lightweight IoT devices from trust information storage and heavy computations and save their bandwidth occupations. Indeed, fog nodes (router, base stations, etc.) are responsible for the management of trust information. Thus, IoT devices do not need to neither perform computation nor to communicate trust information with each other. Moreover, in our solution, some fog nodes could act maliciously without affecting trust management process, since the whole blockchain is trusted. We note that our protocol is far from being a simple implementation of existing blockchain-based solutions. Indeed, we introduce a new transactional system to fit with trust management settings. Moreover, contrary to cryptocurency based blockchains, our blockchain is private and permissioned where only powerful fog nodes and cloud are allowed to validate new blocks. In addition, we adopt a consensus algorithm that combines both proof-of-work (PoW) and proof-of-Stack (PoS) mechanisms, as used by ethereum [23]. This allows to substantially enhance the performance of our solution in terms of computing.

In summary, our trust management protocol offers the following advantages:

- **The scalability**: our architecture scales very well and deals efficiently with tremendous number of IoT devices. Indeed, IoT devices do not need to manage and exchange

trust information with each other, instead the whole process is devoted to fog nodes in a distributed way.
- **A global view of trust data**: in our architecture, trust data is disseminated and duplicated into the Blockchain, maintained by decentralized and powerful fog nodes that make it accessible from anywhere.
- **The mobility support**: given the nature of our architecture which is geographically distributed as well as the ubiquitous nature to access to trust data, mobile devices could assess trustworthiness of service providers in real time after few exchanges with fog nodes and service providers.
- **The optimization of IoT devices resources**: in our architecture, data storage and trust computation are offloaded to powerful fog nodes. Therefore, IoT devices optimize their storage and computation resources.
- **Fine-grained based service protocol**: IoT objects get recommendations about service providers not just according to the service they want, but also according to a set of requirements that these providers are able to satisfy.
- **Resiliency against cooperative attacks**: our proposed approach deals efficiently with cooperative bad-mouthing and ballot-stuffing attacks thanks to the history of the recommendations maintained in the blockchain.

To the best of our knowledge, there is no solution that tackles the problem of trust management for IoE using blockchain technology in fog computing architecture.

The remainder of the paper is organized as follows. We present in the following section some backgrounds about blockchain technology. In Section 5, we describe our security model. We discuss our trust management protocol named ***BC-Trust*** in Section 6. In Section 7, we present some theoretical analysis about the convergence of our protocol and its resiliency against trust-related attacks. Section 8 contains the performance evaluation of our protocol. Section 9 concludes the paper and outlines the future works.

4 Background on Blockchain

Blockchain is a new promising technology that revolutionized the world of cryptocurrency these last years. This technology was introduced first in 2009 with bitcoin by a group of anonymous called Satoshi Nakamoto in [17]. The main aim of this technology is to allow heterogeneous nodes to communicate and exchange assets (coins in the case of bitcoin and similar cryptocurrencies) between them in a completely distributed and secure way without relying to any trusted central entity. Basically, each node in the blockchain does not trust any other node however it trusts the whole blockchain network. Actually, blockchain is a distributed data base where data is replicated and maintained between several nodes that participate in the blockchain. These nodes communicate between each other over a highly distributed and scalable peer to peer network. In the blockchain, each node holds a pair of cryptographic keys (public and private keys) that allows it to generate transactions and interact with the other nodes in the network while preserving the privacy of users. The key advantage of blockchain technology is the transactions' immutability. Indeed it is hard to falsify any transaction once added to the blockchain.

In the distributed P2P blockchain netwrok, it's mandatory that the whole nodes reach a consensus state to validate each transaction. We note that, before adding a transaction to the blockchain, it must be verified and validated by the majority of the nodes. The process of validation is done by a subset of powerful nodes called the miners that must do heavy computations (Proof of Work in the case of bitcoin) in order to solve a mathematical buzzle associated to the block containing a set of transactions to be validated. Once the block is validated, it is simple for each node in the blockchain to verify whether the validation of the block is done correctly. This process allows all the nodes to establish a consensus about the validity of each block before being added to the blockchain. It is impossible in practice to falsify or update one block yet validated without redoing the same heavy validation process for this block and all its subsequent blocks in the blockchain.

5 Security model

In this Section, we define our security model by highlighting the main security attacks that may occur in our system. In our model, we assume that every IoT device may provide services for other devices and it may simply behave as service requester. Moreover, we consider dishonest service providers that act for their own benefits in order to be selected as service providers by other IoT service requesters. Thus, each malicious service provider can perform the following trust-related malicious attacks [14]:
- *Self-promotion attacks*: a malicious service provider can promote its importance and trustworthiness to other service requesters by sending good recommendations about itself, and then it may act maliciously by providing bad services.
- *Bad-mouthing attacks*: a malicious service provider can distrust the trustworthiness of other trusted service providers by providing bad and wrong recommendations about them to service requesters and therefore decrease their chances to be selected as service providers. These attacks could be performed in a collaborative way by a set of malicious service providers to ruin well-behaved nodes.
- *Ballot-stuffing attacks*: a malicious service provider can consolidate other malicious service providers and boost their trustworthiness by providing good recommendations. Therefore, this may increase their chances to be selected as service providers. Similarly to Bad-mouthing attacks, this attack could be performed in collaborative way by malicious nodes to recommend each other.
- *Opportunistic service attacks*: a malicious service provider can decide to provide opportunistically a good service to attract the service requesters and enhance its reputation regarding them. This malicious node could exploit this good opportunistic reputation to perform successful Ballot-stuffing and Bad-mouthing attacks.
- *On-off attacks*: in this kind of attacks, one node can decide to provide good and bad services in a random way to avoid the risk of not being selected as a SP. Once again, with good reputation, this malicious node can perform Ballot-stuffing and Bad-mouthing attacks with the collaboration of other malicious nodes.

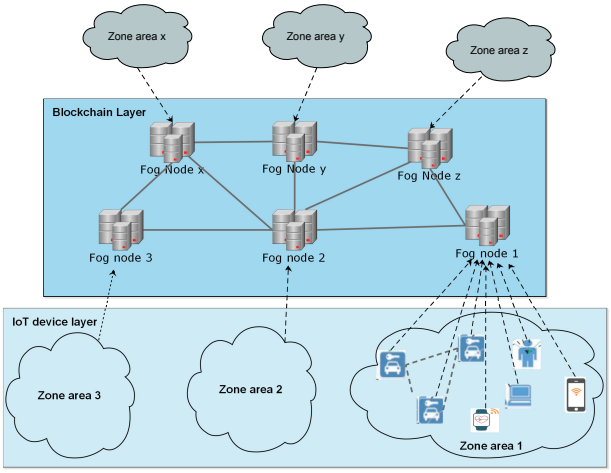


Fig. 1: Our system architecture

6 Our Trust Management Solution

In this Section, we introduce our architecture, then we define the main steps of our protocol which allows any entity in our architecture to measure the trustworthiness of any SP.

6.1 Our architecture

In our solution, we consider a trust management architecture, composed of the following components:

- **IoT devices** which communicate with any other component in the architecture, via Internet or other network protocols. Each device can offer services to other devices and therefore it is considered as a SP, or it can request services (service requester). We note the set of service providers by $Sp = \{Sp_1, Sp_2, ..., Sp_M\}$ and the set of service requesters by $D = \{O_1, O_2, ..., O_N\}$.
- **Fog nodes** which are responsible for a reliable management of trustworthiness in the system. Indeed, the set of fog nodes $FN = \{FN_1, FN_2, ..., FN_P\}$ maintain a Blockchain which stores the various trust values related to the IoT service providers. In addition, fog nodes provide to service requesters a global view on the trustworthiness of each SP. Note that these fog nodes are not assumed to be trusted. Indeed, since our solution is based on blockchain there is no need to trust any node as far as the whole blockchain is trusted.
- **Cloud provider** which is responsible for the identity management of IoT devices and fog nodes.

We illustrate in Figure 1 our architecture on which we base to propose our trust management protocol.

6.2 Our Trust model

In our trust model, we usually use the following appellations that we define as:

- Trust value $T_{ij}^{S}(t)$: is a real number in the range $[0, 1]$ which expresses the trust level of IoT device O_i toward IoT service provider Sp_j with respect to the service S at instant t. The max value 1 means that the node Sp_j (trustee) is full trusted with respect to the node O_i (trustor) and 0 indicates that service provider Sp_j is a bad or malicious node.

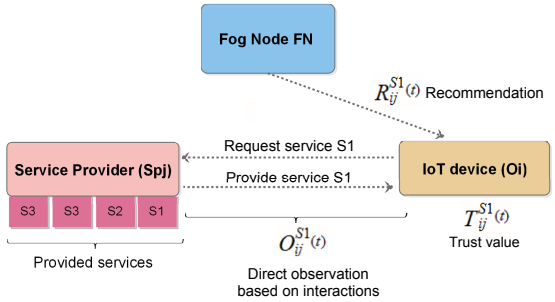


Fig. 2: Our trust model

- Recommendation $R_{ij}^{S}(t)$: is a real number in the range $[0, 1]$ computed by a fog node based on the trust values, which concern service provider Sp_j, reported by IoT devices. This value is sent to IoT device O_i.
- Direct Observation $D_{ij}^{S}(t)$: is a real number in the range $[0, 1]$. It represents the mean of satisfactions against the service S during the interactions between device O_i and service provider Sp_j.

Figure 2 illustrates our trust model, in which we define trust parameters used in our protocol. In addition, Table 1 summarizes the main notations used in this paper.

Notation	Description
O_i	The IoT device i
PK_i	The public key of IoT device O_i
SK_i	The private key of IoT device O_i
S_k	The service k
$T_{ij}^{S_k}(t)$	Trust of O_i toward Sp_j w.r.t. service S_k at time t
$D_{ij}^{S_k}(t)$	Direct observation of O_i toward Sp_j w.r.t. service S_k at time t
$R_{rj}^{S_k}(t)$	Recommendation of O_r toward Sp_j w.r.t. service S_k at time t
S_{ij}	Satisfaction level of O_i toward Sp_j
α_{ij}	Accumulated satisfaction level of O_i toward Sp_j
β_{ij}	Accumulated dissatisfaction level of O_i toward Sp_j
α	Weight on previous experiences
β	Weight on direct observation
γ	Weight on indirect recommendations
Δt_R	The period of time that separates two transactions

TABLE 1: Table of notations

6.3 The protocol BC-Trust for trust management

Our trust management protocol, is a real time, evolutionary and encounter-based assessment process, which provides trust information about any service provider. Indeed, in our protocol, *"honest"* IoT devices continuously evaluate and update trust information about the encountering IoT service providers whenever they request a service. In what follows, we explain the different steps of our protocol called ***BC-Trust***.

6.3.1 *Setup phase*

Our setup phase is composed of two main steps which are :

1) **Identification step**: in a massively distributed system of a very large number of heterogeneous IoT devices, the identification of IoT devices is one of the major challenges that must be addressed before developing a trust management protocol [14]. In our system, we assume that there is a public key infrastructure set up in the cloud which is responsible for cryptographic key generation. Therefore, PKI authority generates a public and private key pair for each IoT device and fog node in the architecture. The public keys are maintained in the blockchain by the fog nodes. Thus, once PKI authority generates the pair (PK_A, SK_A) for each entity A (IoT device or fog node, ie $A \in D \cup FN \cup Sp$), it sends a transaction containing PK_A to the blockchain. PK_A serves as an identifier of the entity A. Hence, at the end of identification step, all IoT devices and fog nodes are able to identify each other via the blockchain.
2) **Service indexing step**: in order to allow IoT devices to discover available services, service providers register their proposed services into their closest fog nodes. Thus, we propose to use a distributed hash table (DHT) to store the different services provided by different service providers. This DHT table, maintained by the fog nodes, is synchronized and updated via a distributed protocol similar to structured P2P networks [10].

6.3.2 Trust Dissemination Phase

In our solution, each *"honest"* IoT device O_i should periodically report its recommendations toward the encountered service providers every ΔT_R time units (ΔT_R is a system parameter). Device O_i's recommendations are reported to the closest fog node. For sake of optimization, each device O_i reports only the most fresh recommendations that have been updated during the last ΔT_R. Therefore, at the end of ΔT_R, the reported trust values are structured in separate transactions, where each transaction $Tx_R(O_i, Sp_j, S_k)$ contains the following pieces of information:

- The trustor node identifier: which is the public key PK_{O_i} of device O_i.
- The trustee node identifier: which is the public key PK_{Sp_j} of service providerSp_j.
- The service S_k that has been provided by node Sp_j to device O_i during the last ΔT_R.
- A set of criteria $C^{'} \subset C = \{C_1, C_2, ..., C_N\}$: that represents the criteria on which O_i has based its evaluation of service S_k.
- The trust value $T_{ij}^{S_k}$ that refers to the level of trustworthiness of the service provider Sp_j assessed by the device O_i with respect to the service S_k and criteria $C^{'}$.
- The timestamp $tsp_{ij}^{S_k}$ of the last updated trust value $T_{ij}^{S_k}$.
- The previous $\{R_{ij}, \Delta T = [t_1, t_2]\}_{SK_{FN_l}}$ signed by FN_l and computed based on trust values reported by IoT devices regarding service provider Sp_j. The computation of R_{ij} takes in consideration only the reported trust values in the interval $\Delta T = [t_1, t_2]$. Further explanations about the computation of R_{ij} are provided in phase 6.3.3.
- The approval of service S_k signed by the service provider Sp_j as: $\{approval, S_k, timestamp\}_{SK_{Sp_j}}$. This information is used as a proof that the service S_k has been accomplished and provided by Sp_j and thus it prohibits that service requester O_i can report a recommendation about the service provider Sp_j without requesting any service from it.

The device O_i signs the transaction $Tx_R(O_i, Sp_j, S_k)$ by its private key SK_{O_i} and sends it to the closest fog node. Upon receiving the transactions, the fog node periodically performs the following steps:

- It first verifies these transactions by verifying the signature of both service provider Sp_j (the approval signature) and service requester O_i (the transaction signature).
- It gathers only the valid transactions in one single block.
- It broadcasts the block to be validated to the whole fog nodes that maintain the blockchain as explained previously in section 4.
- One fog node FN_l validates the block using the Ethereum's Proof of Stack algorithm (PoS). Beside, the validation of the block, FN_l checks out if the computation of the previous $R_{ij}(\Delta t)$ has been well done by the fog nodes which signed these values. For more efficiency, these computations should be done offline by FN_l (and not at this step). In case of any incoherence in the computation of one $R_{ij}(\Delta t)$, the fog node that was responsible of this computation will be reported by FN_l in the blockchain as a malicious node.
- Finally, once the validation is done, the block will be added to the blockchain by all fog nodes.

6.3.3 Trust assessment process

Whenever, the node O_i requests the service S_k from service provider Sp_j at time t, it first queries for the available services from the distributed hash table (maintained by the fog nodes) to identify the potential IoT service providers it should interact with them. The choice of one service provider Sp_j among others is based on the trustworthiness level of each service provider at time t. The trustor IoT device O_i assesses or updates the trustworthiness of service provider Sp_j (trustee) as follows:

$$T_{ij}^{S_k}(t) = \begin{cases} \alpha T_{ij}^{S_k}(t - \Delta t) + \beta D_{ij}^{S_k}(t) + \gamma R_{ij}^{S_k}(\Delta t), \text{ if } P(i,j) \\ R_{ij}^{S_k}(\Delta t), \text{ otherwise} \end{cases} \quad (1)$$

Where $0 \leq \alpha, \beta, \gamma \leq 1$ and $\alpha + \beta + \gamma = 1$, are used to weigh the importance of each trust parameter. These weighs are adjusted dynamically by the trustor in order to maximize the accuracy of trust assessment as well as make the protocol more resilient to bad-mouthing and ballot-stuffing attacks. In equation 1, $P(i,j)$ is a predicate that is equal to *true* if the device O_i has interacted previously with the service provider Sp_j. Otherwise, $P(i,j) =$ *false*.

In the equation 1 above, we distinguish two main cases depending on the experience of the node O_i with the encountered IoT service provider Sp_j:

1) **Case 1**: if the device O_i has previously encountered the service provider Sp_j, it will assess its trustworthiness level based on $T_{ij}^{S_k}(t - \Delta t)$, $D_{ij}^{S_k}(t - \Delta t)$ and $R_{ij}^{S_k}(\Delta t)$. $T_{ij}^{S_k}(t - \Delta t)$ represents the last trustworthiness of service provider Sp_j. $D_{ij}^{S_k}(t - \Delta t)$ represents the direct observation measured till instant t. The last parameter denoted by $R_{ij}^{S_k}(\Delta t)$ refers to the indirect recommendations of the other IoT devices toward Sp_j.

2) **Case 2**: if the device O_i has not interacted previously with the service provider Sp_j and it does not dispose of any previous trustworthiness level $T_{ij}^{S_k}(t-\Delta t)$ about Sp_j, then it considers only the indirect recommendation $R_{ij}^{S_k}(\Delta t)$ as trustworthiness value $T_{ij}^{S_k}(t)$.

6.3.4 Computation of $D_{ij}^{S_k}(t)$

In our protocol, when O_i requests one service S_k from Sp_j, it measures the satisfaction level of the provided service. Let $S_{ij}(t)$ be the current satisfaction level, which is a real number in the range $[0, 1]$. The direct observation $D_{ij}^{S_k}(t)$ is defined through as follows:

$$D_{ij}^{S_k}(t) = \frac{\alpha_{ij}}{n} = \frac{\sum_{t_i \in \{t_1,..,t_n\}} S_{ij}(t_i)}{n} \quad (2)$$

Where:

- α_{ij} is the cumulative of the satisfaction levels and is continuously updated by $\alpha_{ij} = \alpha_{ij} + S_{ij}(t)$.
- $t_1 < t_2 < ... < t_n = t$ represent the instants where service S_k was requested.
- n is the number of experiences regarding the service S_k.

Algorithm 1 summarizes the different steps of trust assessment protocol, executed by IoT devices.

Algorithm 1 BCTrust: trust assessment-IoT devices level

```
Input: O_i: IoT device, Sp_j: IoT service provider
procedure ComputeAndReportTrust
    Requests a recommendation about Sp_j from the home fog node
    Fog node sends the recommendation R_ij^{S_k} to O_i
    if (T_ij^{S_k}, D_ij^{S_k}) ∈ loockup(O_i) then
        T_ij^{S_k} ← α × T_ij^{S_k} + β × D_ij^{S_k} + γ × R_ij^{S_k}
    else
        T_ij^{S_k} ← R_ij^{S_k}
    end if
    if T_ij^{S_k} < Threshold then
        Ignore the service provider Sp_j
        return false
    else
        Service S_k Done
        Evaluate the satisfaction S_ij(t) ∈ [0,1]
        α_ij ← α_ij + S_ij(t); n ← n + 1;
        D_ij ← α_ij / n
        Update the entry (D_ij, T_ij) in the loockup table
        Construct and send transaction Tx_R(O_i, Sp_j, S_k)
        return True
    end if
end procedure
```

6.3.5 Computation of $R_{ij}^{S_k}(\Delta t)$

As previously explained, our trust assessment is also based on recommendations provided by fog nodes. These recommendations are computed using trust values stored in the blockchain.

To provide indirect recommendation $R_{ij}^{S_k}(\Delta t)$, fog node FN_l starts by filtering out the most recent transactions, which have been occurred during the last Δt time units, available in the blockchain. We denote by L the list of IoT objects which have reported the filtered transactions. Next, from the list L, we distinguish two cases:

1) **Case 1 ($L \neq \emptyset$)**: fog node FN_l computes $R_{ij}^{S_k}(\Delta t)$ as follows:

$$R_{ij}^{S_k}(\Delta t) = sp \times Rs_{ij}^{S_k}(\Delta t) + (1-sp) \times Ro_{ij}^{S_k}(\Delta t) \quad (3)$$

Where:

- sp : the rate of service providers in the list L $(0 \leq sp \leq 1)$
- $Rs_{ij}^{S_k}(\Delta t)$: the average of the recommendations provided by service providers.
- $Ro_{ij}^{S_k}(\Delta t)$: the weighted average of the recommendations provided by IoT devices.

Overall, in equation 3, the computation of $R_{ij}^{S_k}(\Delta t)$ depends upon two different values $Rs_{ij}^{S_k}(\Delta t)$ and $Ro_{ij}^{S_k}(\Delta t)$. Indeed, in our solution, service provider Sp_j could be recommended by both IoT devices or other service providers.

Therefore, in the list L, fog node FN_l selects the subset L_O of IoT devices. Then, it computes $Rs_{ij}^{S_k}(\Delta t)$ as follows:

$$Ro_{ij}^{S_k}(\Delta t) = \frac{1}{(1-sp)|L|} \sum_{k \in L_O} T_{kj}^{S_k} \quad (4)$$

Where:

$L_O \subset L$: is a subset of L that contains only service requesters.

Equation 4 represents the average of all recommendations ($T_{kj}^{S_k}$) that were reported by all devices $O_k \in L_O$ and stored in the blockchain during the last period ΔT.

Likewise, fog node FN_l, selects the subset $L_S (L_S \subset L)$ of IoT devices. Then, it computes $RS_{ij}^{S_k}(\Delta t)$ as follows:

$$Rs_{ij}^{S_k}(\Delta t) = \sum_{k \in L_s} \frac{T_{ik}^{S_k}}{\sum_{k \in L_s - \{j\}} T_{ik}^{S_k}} \times T_{kj}^{S_k} \quad (5)$$

Equation 4 represents the weighted average of all recommendations $T_{kj}^{S_k}$ that were reported by all devices $Sp_k \in L_S$.

In fact, each recommendation value $T_{kj}^{S_k}$ provided by O_k is weighted by the ratio of the trust value reported by O_i toward Sp_k, to the sum of all trust values given by O_i toward each service provider in L_S. Hence, if the trust value $T_{ik}^{S_k}$ of O_i toward Sp_k is high, then the fog node will attribute a high weight to the recommendation $T_{kj}^{S_k}$. For sake of optimization, the fog node only considers the recommendation coming from service providers that device O_i grants them a minimum trust value. As an example, fog node considers the recommendations provided by the service providers if their trust value regarding O_i exceed 0.7 (i.e. $T_{ik}^{S_k} > 0.7$).

Finally, fog node FN_l computes the recommendation $R_{ij}^{S_k}(\Delta t)$, it responds the device O_i by sending $\{R_{ij}^{S_k}(\Delta t), \Delta t = [t_1, t_2]\}_{SK_{FN_l}}$ signed by its private key SK_{FN_l}. This information will be integrated in the next transaction that will be sent by the device O_i as explained previously in the Section 6.3.2. It allows the other fog nodes to detect any misbehavior from fog node FN_l during block validation step (Section 6.3.2).

2) **Case 2 ($L = \emptyset$)**: this case means that there have been no device which recommended Sp_j during the last ΔT time units. If service provider Sp_j has never been recommended by any IoT object in the architecture, then fog node FN_l returns a recommendation $R_{ij}^{S_k}(T) = 0.5$. Otherwise, fog node FN_l searches the most recent transaction Tx_R that has been reported prior interval $[t - \Delta T, t]$. Since Tx_R has not been reported in the last ΔT, it is still considered as an old transaction. Therefore, fog node FN_l will consider recommendation reported in transaction Tx_R with a small

penalty Pnl. In our solution, we consider a constant penalty Pnl equal to 0.05. Thus, let $R_{kj}^{S_k}(t^{'})$ be the recommendation reported in Tx_R ($t^{'} < t - \Delta T$), fog node FN_l computes the recommendation $R_{ij}^{S_k}(\Delta t)$ as follows:

$$R_{ij}^{S_k}(\Delta t) = (1 - Pnl) \times R_{ij}^{S_k}(t^{'}) \text{ , where } t^{'} < t - \Delta T$$

Algorithm 2 summarizes the different steps performed by fog nodes while computing recommendations.

Algorithm 2 BCTrust: trust assessment-Fog nodes level

procedure COMPUTERECOMMENDATION
  Init1: $L_S \leftarrow \{\}$; $L_O \leftarrow \{\}$
  Init2: $L \leftarrow$ *the* T*-th most recent recommenders that reported transactions in* $[t - \Delta T, t]$
  if $L = \emptyset$ **then** $R_{ij}^{S_k} \leftarrow 0.5$
    if $\exists R_{kj}^{S_k}(t^{'}) \in Blockchain$ && $t^{'} < t - \Delta T$ **then**
      $R_{ij}^{S_k} \leftarrow (1 - Pnl) \times R_{ij}^{S_k}(t^{'})$
    end if
    Send the recommendation $R_{ij}^{S_k}$ *to the device* O_i
    return $R_{ij}^{S_k}$
  end if
  for $O_k \in L$ **do**
    if O_k *is a service provider* **then**
      $L_S \leftarrow Sp \bigcup \{O_k\}$
    else
      $L_O \leftarrow Sr \bigcup \{O_k\}$
    end if
  end for
  Compute $Ro_{ij}^{S_k}$ *//recommendation of* L_O *(equation 4)*
  Compute $Rs_{ij}^{S_k}$ *//recommendation of* L_S *(equation 5)*
  $R_{ij}^{S_k} \leftarrow Sp \times Rs_{ij}^{S_k} + (1 - Sp) \times Ro_{ij}^{S_k}$
  Send $R_{ij}^{S_k}$ *to the device* O_i
  return $R_{ij}^{S_k}$
end procedure

We illustrate in Figure 3, in a comprehensive way, the different steps of our protocol.

6.4 Countermeasure against cooperative attacks

The most common attacks that are performed in IoE based trust management systems are basically bad-mouthing and ballot stuffing attacks. In these attacks, malicious nodes tend to report bad recommendations for honest service providers or good recommendations for malicious ones. For more effectiveness, in general, this kind of attacks is cooperatively performed by several attackers in order to promote each other or target some honest service providers. Cooperative bad-mouthing and ballot-stuffing attacks involve great damages on the whole IoE system. Moreover, these attacks are very hard to detect and overcome, at least for the following reasons:

- **Risk of false negative**: When a group of nodes give bad recommendations for one particular node A repetitively, it is hard to say for sure whether this group of nodes is malicious or because the node A is really malicious.
- **Risk of false positive**: It could be possible in some cases that a group of nodes request periodically one particular service from one service provider (the case for example of data aggregation). Therefore, reporting periodically the same recommendations for one service provider (the aggregator node for example) does not necessarily mean that this group of nodes is conducting a cooperative attack against this service provider.

Algorithm 3 BCTrust: Countermeasure against cooperative attacks

Input: O_i*: IoT device,* Sp_j*: IoT service provider*
procedure ONLINE COUNTERMEASURE
  Init: $Sp \leftarrow \{\}$; $Sr \leftarrow \{\}$; $Nbocc[T] \leftarrow \{0\}$
  $L \leftarrow$ *the most recent recommenders that reported transactions in* $[t - \Delta T, t]$
  $min_j(t) \leftarrow min_{i \in L}\{T_{ij}^{S_k}(t)\}$
  $max_j(t) \leftarrow max_{i \in L}\{T_{ij}^{S_k}(t)\}$
  if $max_j(t) - min_j(t) < Thr$ **then**
    $History \leftarrow$ *transactions produced during* $[t - n \times \Delta T, t]$
    for $O_k \in TopR$ **do**
      for $i := 1$ to n **do**
        if $T_{kj}^{S_k}(t - i\Delta T) \in History$ **then**
          $Nbocc[k] \leftarrow Nbocc[k] + 1$
        end if
      end for
    end for
    for $O_k \in L$ **do**
      if $\frac{Nbocc[k]}{n} > 0.8$ **then**
        $L \leftarrow L - \{O_k\}$
      end if
    end for
  end if
  $R_{ij}^{S_k} \leftarrow$ COMPUTEANDREPORTTRUST(L)
  return $R_{ij}^{S_k}$
end procedure

In this section, we propose a countermeasure solution to reduce the impact of cooperative attacks in the system. Our mitigation technique takes advantage of the history of the recommendations reported to the blockchain. The main idea of our solution consists to: 1) Analyze the history of the received recommendations to detect if there is a cooperative attack. 2) Trigger a mitigation technique to eliminate the recommendations provided by the group of malicious nodes in the case of any eventual cooperative attack.

We propose as a countermeasure, an online algorithm, which works in real time and is executed each time the trust recommendations are computed by fog nodes. As presented in algorithm 2, our mitigation algorithm works in the following steps:

- First, the fog node selects all the recommendations for one particular IoT service provider S_k (as discussed previously in our protocol). Let $L = \{O_1, O_2, ..., O_l\} \cup \{Sp_1, Sp_2, ..., Sp_m\}$ be the subset of IoT devices and service providers that have recommended Sp_k during the last ΔT.
- The fog node computes $min_k(t) = min_{i \in L}\{T_{ik}^{S_k}(t)\}$ and $max_k(t) = max_{i \in L}\{T_{ik}^{S_k}(t)\}$ which are respectively the minimum and the maximum of the recommendations provided by the devices of the list L. If the difference $max_k(t) - min_k(t)$ is bigger than Thr, then the sevice provider Sp_k may be subject of a cooperative attack. Indeed, having a large difference between $max_k(t)$ and $min_k(t)$ is a suspicious situation. In fact, there is at least one node who did not grant a good recommendation to Sp_k contrariwise to others. Thus, one of these sub-groups is malicious (see from step 3 to step 7 in algorithm 2).
- If an anomaly has been detected, the fog node consults the history of recommendations, available in the blockchain, which concern service provider Sp_k in the

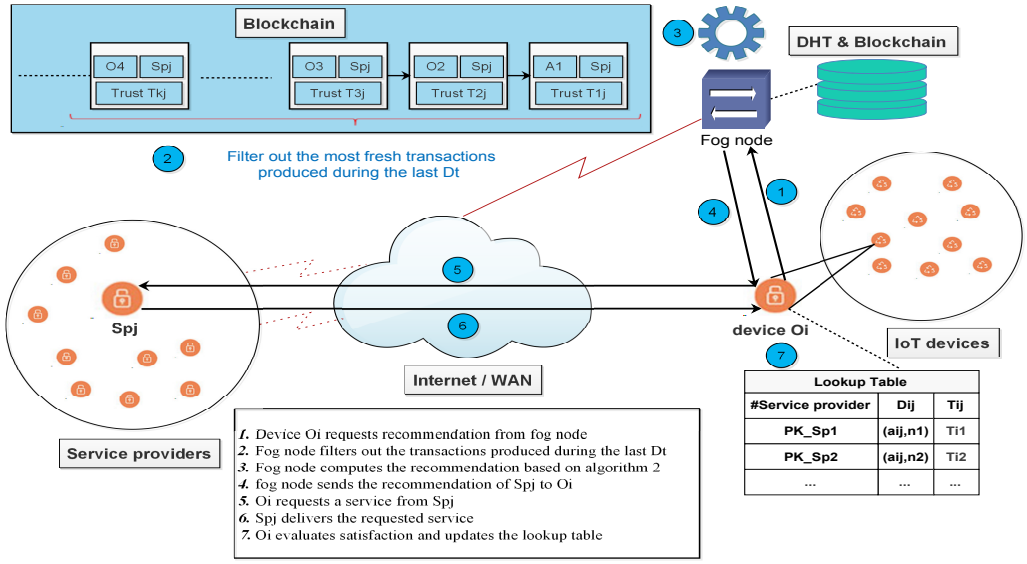


Fig. 3: work-flow of our trust management protocol ***BC-Trust***

last n time slots ΔT. The fog node ignores the recommendation of each node who frequently appears in the history (see from step 7 to step 16 in algorithm 2).

7 Theoretical Analysis

In this section, we will study the convergence of our protocol ***BC-Trust*** with respect to the parameters of our system. In this theoretical analysis, we give lower and upper bounds of trust values obtained by our protocol under bad-mouthing and ballot stuffing attacks, showing that our protocol is highly resilient to these attacks. We recall in Table 2 the symbols used in this section.

Notation	Description
L	The number of IoT devices (service providers and service requesters)
sp	The rate of service providers
λ	The rate of honest devices
m	The minimum satisfaction value that can be attributed to one honest service provider Sp_j by honest device O_i
T_n^{ij}	The trust value attributed to service provider Sp_j by an honest IoT device O_i at time n
H, M	The subsets of honest and malicious devices respectively
$E_h(T_j)$	The mean trust value of honest service provider Sp_j, measured by all IoT devices
$E_m(T_j)$	The mean trust value of malicious service provider Sp_j, measured by all IoT devices

TABLE 2: Table of symbols

In our solution, trust values $T_{ij}^{S_k}$ are updated at each time that device O_i requests a service S_k from service provider Sp_j. We define the set $\{T_0, T_1, T_2, ...\}$ as an ordered set of instants when O_i requests S_k. Hence, each T_n refers to the n^{th} service request. For sake of simplicity, we consider only one service in what follows. Thus, we note $T_{ij}^{S_k}(T_n)$ by T_n^{ij}.

Definition 1. *We define the sequence* $\mathcal{S} = (T_n^{ij})_{n\in\mathbb{N}}$ *by the set of trust values* $T_{ij}(t)$, $t \in [T_n, T_{n+1}]$, $n \in \mathbb{N}$.

Definition 2. *We define the sequence* $\mathcal{R} = (R_n^{ij})_{n\in\mathbb{N}}$ *by the set of recommendation values* $R_{ij}(t)$ *reported by fog nodes at each instant* $t \in [T_n, T_{n+1}]$, $n \in \mathbb{N}$.

Definition 3. *We define the sequence* $\mathcal{D} = (D_n^{ij})_{n\in\mathbb{N}}$ *by the set of direct observations* $D_{ij}(t)$, $t \in [T_n, T_{n+1}]$, $n \in \mathbb{N}$.

7.1 Study of the convergence of $\mathcal{S} = (T_n^{ij})_{n\in\mathbb{N}}$

Lemma 1. *Given a network of L devices. For each honest device* O_i *and honest service provider* Sp_j, *we have:*

$$\forall i, j \in \{1, ..., L\}, i \neq j, m \leq D_n^{ij} \leq 1 \tag{6}$$

Proof. From equation 2, we have:

$$D_n^{ij} = \frac{\alpha_{ij}}{n} = \frac{\sum_{t=1}^{n} S_{ij}(t)}{n} \tag{7}$$

Since Sp_j is a honest service provider, the satisfaction value $S_{ij}(t)$ at time t is at least equal to m and at most equal to 1. Therefore, we obtain from equation (7):

$$m \leq D_n^{ij} \leq 1$$

□

Lemma 2. *Given a network of L devices with a rate sp of service providers and* λ *the rate of honest devices. Under bad-mouthing attacks, for each honest device* O_i *and honest service provider* Sp_j, *we have:*

$$\forall i, j \in \{1, ..., L\}, i \neq j, R_n^{ij} \geq \lambda \times T_n^{min} \tag{8}$$

Where $T_n^{min} = min\{T_n^{kj}, k \in \{1, ..., L\}, and\ O_k \in H\}$

Proof. From equation 3, we have:

$$R_n^{ij} = sp \times Rs_{ij}(n) + (1 - sp) \times Ro_{ij}(n)$$

Given a set $L' = L'_S \cup L'_O$ composed of two subsets L'_S (service providers) and L'_O (IoT devices) that have recommended Sp_j. We distinguish two cases for each subset:

1) For the subset $L_O^{'}$, recommendation $Ro_{ij}(n)$ is expressed as follows:

$$Ro_{ij}(n) = \frac{1}{|L_O^{'}|} \times \sum_{k \in L_O^{'}} T_{n-1}^{kj} = R_H + R_M$$

Where:

$$R_H = \frac{1}{|L_O^{'}|} \times \sum_{k \in L_O^{'} \cap H} T_{n-1}^{kj}$$

$$R_M = \frac{1}{|L_O^{'}|} \times \sum_{k \in L_O^{'} \cap M} T_{n-1}^{kj}$$

In what follows, we study the lower bounds of the of (R_H and R_M).

Case 1: the sum R_H

$$R_H = \frac{1}{|L_O^{'}|} \times \sum_{k \in L_O^{'} \cap H} T_{n-1}^{kj}$$

By definition, for each $n \geq 0$, we have:

$$\forall i, j \in \{1, ..., L\}, T_n^{kj} \geq T_n^{min}$$

Hence, given that λ is the rate of honest devices in $L_O^{'}$, we can simplify R_H as follows:

$$\forall n \geq 1, R_H \geq \lambda \times T_{n-1}^{min} \tag{9}$$

Case 2: the sum R_M

Under bad-mouthing attacks, malicious devices report bad recommendations T_n^{ij} which are equal to 0 in the worst case. Therefore:

$$R_M = \frac{1}{|L_O^{'}|} \times \sum_{k \in L_O^{'} \cap M} T_{n-1}^{kj}$$

$$T_n^{ij} \geq 0 \implies R_M = \frac{1}{|L_O^{'}|} \times \sum_{k \in L_O^{'} \cap M} T_{n-1}^{kj} \geq 0 \tag{10}$$

From inequalities 9 and 10, we have:

$$Ro_{ij}(n) \geq \lambda \times T_{n-1}^{min} \tag{11}$$

2) For the subset $L_S^{'}$, recommendation $Rs_{ij}(n)$ is expressed as follows:

$$Rs_{ij}(n) = \sum_{k \in L_S^{'}} \frac{T_{n-1}^{ik}}{\sum_{k \in L_S^{'}} T_{n-1}^{ik}} \times T_{n-1}^{kj} = R_X + R_Y$$

Where:

$$R_X = \sum_{k \in L_S^{'} \cap H} \frac{T_{n-1}^{ik}}{\sum_{k \in L_S^{'}} T_{n-1}^{ik}} \times T_{n-1}^{kj}$$

$$R_Y = \sum_{k \in L_S^{'} \cap M} \frac{T_{n-1}^{ik}}{\sum_{k \in L_S^{'}} T_{n-1}^{ik}} \times T_{n-1}^{kj}$$

As previously, we develop bellow the sums R_X and R_Y.

Case 1: the sum R_X

Given λ the rate of honest service providers in $L_S^{'}$, we have :

$$R_X \geq \sum_{k \in L_S^{'} \cap M} \frac{T_{n-1}^{ik}}{|L_S^{'}| \times T_{n-1}^{ij}} \times T_{n-1}^{min} \geq \lambda \times T_{n-1}^{min} \tag{12}$$

Case 2: the sum R_Y

Under bad-mouthing attacks, malicious service providers report bad recommendations T_n^{ij} which are equal to 0 in the worst case. Therefore:

$$T_n^{ij} \geq 0 \implies R_Y \geq \sum_{k \in L_S^{'} \cap M} \frac{T_{n-1}^{ik}}{\sum_{k \in L_S^{'}} T_{n-1}^{ik}} \times 0 \geq 0 \tag{13}$$

From inequalities (12) and (13):

$$Rs_{ij}(n) \geq \lambda \times T_{n-1}^{min} \tag{14}$$

From inequalities (11) and (14), we find out:

$$R_n^{ij} \geq \lambda \times T_{n-1}^{min}$$

□

7.1.1 Resiliency against malicious attacks

Proposition 1. *Given a network of L devices with sp the rate of service providers and λ ($\lambda \leq 1$) the rate of honest devices. Under bad-mouthing attacks, for each honest device O_i and honest service provider Sp_j, we have:*

$$\forall i, j \in \{1, ..., L\}, i \neq j, T_h = \lim_{n \to \infty} T_n^{ij} \geq \frac{m \times \beta}{1 - \alpha - \gamma \times \lambda} \tag{15}$$

Proof. Given O_i and Sp_j are honest. By definition, we have: $\forall n \geq 0$, $T_n^{ij} \geq T_n^{min}$. Thus, we only need to study the convergence of the sequence $(T_n^{min})_{n \in \mathbb{N}}$.

Based on the result of lemma 1 and lemma 2, we have:

$$T_n^{min} \geq \alpha \times T_{n-1}^{min} + \beta \times m + \gamma \times \lambda \times T_{n-1}^{min}$$

Hence, we get:

$$\lim_{n \to \infty} T_n^{min} \geq (\alpha + \gamma \times \lambda) \times \lim_{n \to \infty} T_{n-1}^{min} + \beta \times m$$

Therefore:

$$\lim_{n \to \infty} T_n^{min} \geq \frac{\beta \times m}{1 - \alpha - \gamma \times \lambda}$$

Since $\forall n \geq 0$, $T_n^{ij} \geq T_n^{min}$, we have:

$$\lim_{n \to \infty} T_n^{ij} \geq \lim_{n \to \infty} T_n^{min}$$

Therefore,

$$T_h = \lim_{n \to \infty} T_n^{ij} \geq \frac{\beta \times m}{1 - \alpha - \gamma \times \lambda}$$

□

Proposition 2. *Given a network of L devices with sp the rate of service providers and λ ($\lambda \leq 1$) the rate of honest devices. Under ballot-stuffing attacks, for each honest device O_i and malicious service provider Sp_j, we have:*

$$\forall i, j \in \{1, ..., L\}, i \neq j, T_m = \lim_{n \to \infty} T_{ij}(n) \leq 1 - \frac{m \times \beta}{1 - \alpha - \gamma \times \lambda} \tag{16}$$

Proof. The proof is similar to the proof of proposition 1. □

Theorem 1. *Given a network of L devices with sp the rate of service providers and λ the rate of honest devices. Under bad-mouthing attacks, the mean trust $E_h(T_j)$ of honest service providers measured by all network devices is:*

$$E_h(T_j) \geq \lambda \times T_h \geq \frac{\lambda \times \beta \times m}{1 - \alpha - \gamma \times \lambda} \tag{17}$$

Proof. Let Sp_j be a honest service provider, we have:

$$\begin{aligned} E_h(T_j) &= \lim_{n\to\infty} \frac{1}{L}\sum_{i=1}^{L} T_n^{ij} \\ &= \lim_{n\to\infty} \frac{1}{L}\sum_{i=1}^{L} [Pr(O_i \text{ is honest}) \times T_n^{ij} + \\ & Pr(O_i \text{ is malicious}) \times T_n^{ij}] \\ &= \lim_{n\to\infty} \lambda \times T_n^{ij} + (1-\lambda) \times a \end{aligned}$$

Since $a \geq 0$, the worst value of a given by bad-mouthing attacker is 0. Hence, we have:

$$\begin{aligned} E_h(T_j) &\geq \lim_{n\to\infty} \lambda \times T_n^{ij} + (1-\lambda) \times 0 \\ &\geq \lambda \times T_h \geq \frac{\lambda \times \beta \times m}{1-\alpha-\gamma \times \lambda} \end{aligned}$$

□

Theorem 2. *Given a network of L devices with sp the rate of service providers and λ the rate of honest devices. Under ballot-stuffing attacks, the mean trust $E_m(T)$ of dishonest service providers measured by all network devices is:*

$$E_h(T_j) \leq \lambda \times T_m + 1 - \lambda \leq 2 - \lambda - \frac{\lambda \times \beta \times m}{1-\alpha-\gamma \times \lambda} \quad (18)$$

Proof. Let Sp_j be a dishonest service provider, we have:

$$\begin{aligned} E_h(T_j) &= \lim_{n\to\infty} \frac{1}{L}\sum_{i=1}^{L} T_n^{ij} \\ &= \lim_{n\to\infty} \frac{1}{L}\sum_{i=1}^{L} [Pr(O_i \text{ is honest}) \times T_n^{ij} + \\ & Pr(O_i \text{ is malicious}) \times T_n^{ij}] \\ &= \lim_{n\to\infty} \lambda \times T_n^{ij} + (1-\lambda) \times a \end{aligned}$$

Since $a \leq 1$, the best value of a given by a ballot-stuffing attacker is 1. Hence, we have:

$$\begin{aligned} E_h(T_j) &\leq \lim_{n\to\infty} \lambda \times T_n^{ij} + (1-\lambda) \times 1 \\ &\leq \lambda \times T_m + 1 - \lambda \leq 2 - \lambda - \frac{\lambda \times \beta \times m}{1-\alpha-\gamma \times \lambda} \end{aligned}$$

□

8 Performances Evaluation

In this section, we evaluate the effectiveness, resiliency and the benefits of our proposed ***BC-Trust*** approach through different experiments. In addition, we demonstrate how our experimental results feet with the theoretical analysis we presented in the previous section. Basically, we performed three initial experiments. The first one evaluates the effectiveness of our solution in terms of convergence time with respect to different parameters (α, β, γ). The second one evaluates the resiliency of our protocol against bad-mouthing and ballot-stuffing attacks. Finally, we evaluate the effectiveness of our countermeasure approach against cooperative attacks. Table 3 summarizes the main setting parameters related to our experiments.

parameters	values
Number of IoT devices (L)	100
Rate of service providers (sp)	20%
Default values of (α, β, γ)	$\alpha = \beta = \gamma = \frac{1}{3}$
Number of services	1
Number of criteria	5
Δt	5 seconds

TABLE 3: Test settings

8.1 Evaluation of the convergence of our protocol

The first bunch of experiments aims to measure the convergence time of our protocol, and to study the impact of parameters α, β, γ and m on both convergence value and time. In order to get a clear view on the behavior of our protocol, this first sequence of experiments is done in a safe area where all the nodes are assumed to be honest.

Figure 4 illustrates the evolution of the mean trust value of all the service providers seen by all IoT devices during the lifetime of the simulation. We clearly notice that the limit trust value depends on the parameter m (the minimum satisfaction level that can be attributed to honest service providers). Besides, this limit trust value converges to the value $\frac{m+1}{2}$ which exactly feet with the result of proposition 1. However, we notice that the convergence time does not depend on the parameter m. Indeed, even with two different m values, our protocol converges to almost the same time (convergence after about 70 time units).

Figure 5 depicts the mean trust value with respects to the parameters: α, β, γ. As we notice, these three parameters have an impact only on the convergence time of the mean trust value. However, these parameters do not affect the convergence value. Moreover, parameter β (the weight of direct observation) enhances significantly the convergence time compared to parameters α and γ. Indeed, with $\beta = \frac{2}{3}$ and $\alpha = \gamma = \frac{1}{6}$, the convergence time is reduced to around 40 time units, whereas with smaller value of β (i.e. $\beta = \frac{2}{3}$) the convergence time is significant (> 80 time units).

8.2 Effectiveness of our protocol against Malicious attacks

After studying the behavior of our protocol in normal circumstances, we evaluate in what follows its effectiveness under malicious attacks. We mainly focus on two kind of attacks: bad-mouthing and ballot-stuffing attacks.

As illustrated in Figure 6, the robustness of our protocol against bad-mouthing attacks has been evaluated with respect to the rate of honest nodes (λ). To do so, we vary the rate of honest nodes λ and the parameter m while the other parameters are kept constant and take their default values as shown in Table 3. Overall, we notice that the limit of mean trust value for honest service providers is reduced compared the result obtained in the case where there is no attack. As trivially expected, this limit value decreases with respect to the rate of malicious nodes $(1-\lambda)$. However, even with 20% of malicious nodes and $m = 0.9$, our protocol converges to a mean trust value which exceeds 0.75. This is due to our strategy of the computation of recommendations which favors trust values coming from honest nodes. Moreover, it is straightforward to see that

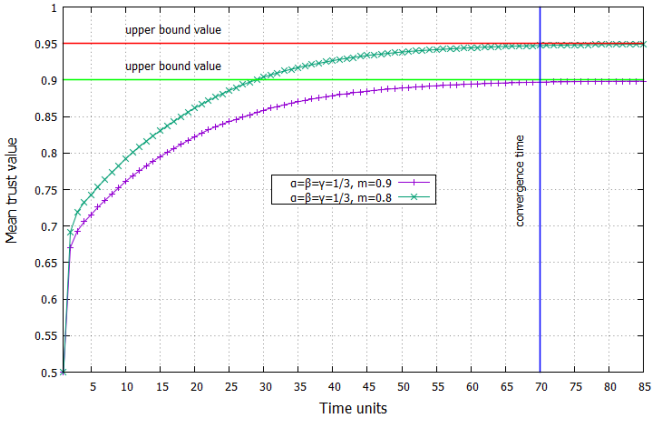


Fig. 4: Mean trust for the different service providers

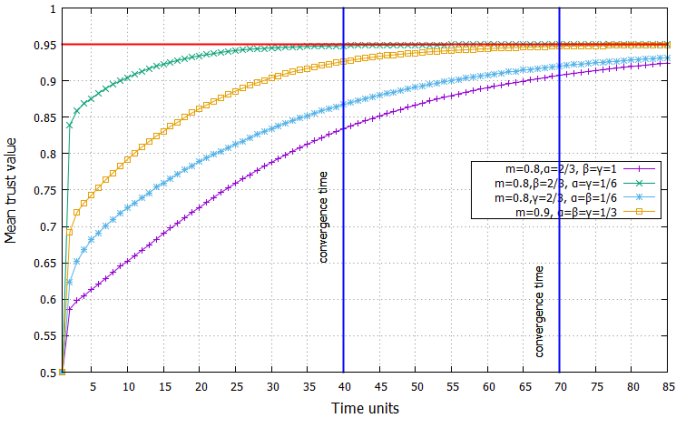


Fig. 5: Mean trust for the different service providers

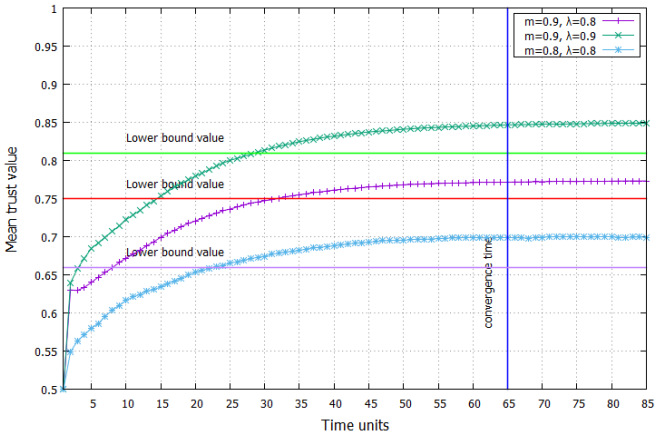


Fig. 6: Mean Trust under bad-mouthing attacks

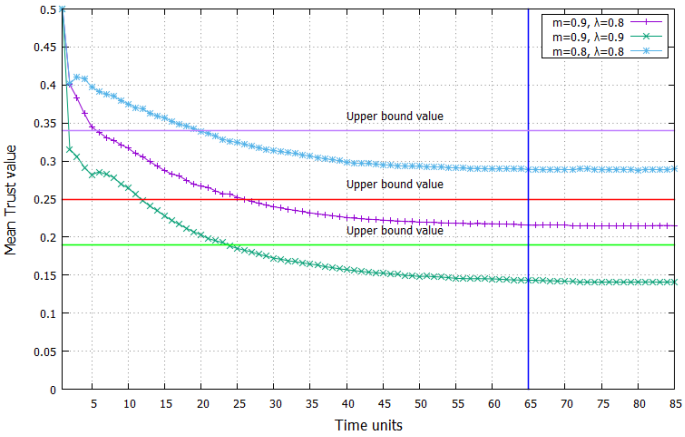


Fig. 7: Mean Trust under ballot-stuffing attacks

the limit mean trust value is always bigger than the lower bound obtained in the theoretical analysis (see proposition 1) with a small gap which is up to 4%.

On the other side, we evaluated the impact of ballot stuffing attacks on our protocol by varying the rate of honest nodes λ. Figure 7 illustrates the mean trust value of malicious service providers (evaluated by honest nodes) with respect to different values of λ and m. Despite the presence of significant malicious nodes ($1 - \lambda = 20\%$), we notice that the limit trust value is still small and reflects a correct reputation on these malicious nodes. Moreover, it is worth nothing that the theoretical analysis discussed in proposition 2 (upper bound limit of mean trust value of malicious nodes under ballot-stuffing attacks) are confirmed in the Figure 7.

Overall, the above results exhibit that ***BC-Trust*** shows its effectiveness and robustness to deal against bad-mouthing and ballot-stuffing attacks.

8.3 Effectiveness against Cooperative attacks

In order to evaluate the efficiency and robustness of our countermeasure approach against cooperative attacks, we performed a set of experiments, defined by the following scenarios:

- **Scenario 1**: We perform a cooperative bad-mouthing attack in which, all the malicious nodes target one service provider Sp_j and periodically report bad recommendations about it. The other honest nodes behave naturally, where they choose the service provider Sp_j randomly among other service providers and report real recommendations about it.
- **Scenario 2**: we perform a cooperative ballot-stuffing attack in which all malicious nodes periodically report good recommendations about a target malicious service provider Sp_k, whereas honest nodes provide real recommendations about Sp_k.

In both scenarios, we vary the rate of malicious nodes ($1 - \lambda$) to show the resiliency of our approach.

Figure 8 shows the evolution of mean trust value of the target honest service provider under bad-mouthing attacks. We notice that our online countermeasure algorithm significantly reduces the effect of collaborative attacks compared to the case where there is no countermeasure. Indeed, despite the presence of $1 - \lambda = 20\%$ of malicious nodes conducting bad-mouthing attacks, the mean trust value of the target service provider reaches the limit value 0.87. This last is significantly bigger than the reached limit value in the case where there is no countermeasure (0.59).

Similarly, in Figure 9, we show the results of experiments conducted on ***BC-Trust*** with the presence of ballot stuffing attacks by varying the rate λ. Our countermeasure algorithm also mitigates the trust computation process per-

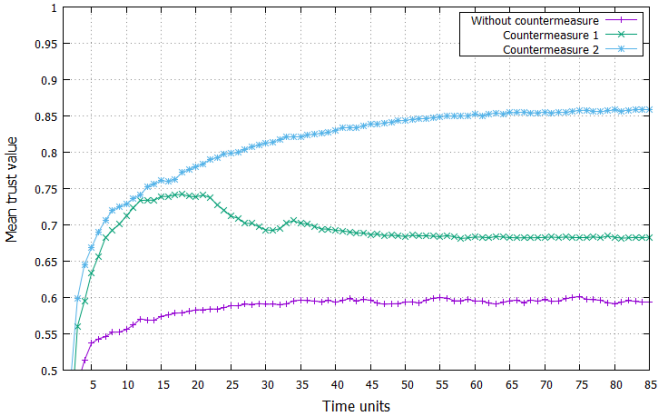


Fig. 8: Trust under cooperative bad-mouthing attacks

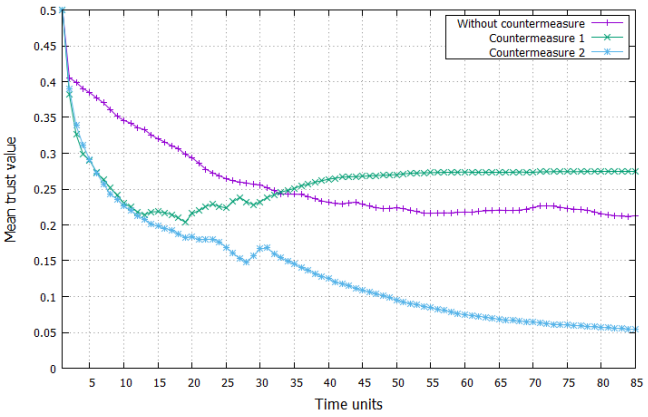


Fig. 9: Trust under cooperative ballot-stuffing attacks

TABLE 4: comparison in terms of trust evaluation cost

	Storage	Computation			Communication
		#Mult	#Add	#Exp	
[7]	$O(L^2)$	$O(L)$	$O(L)$	0	$O(L)$
[18]	$O(F)$	$O(1)$	$O(F)$	0	$O(F)$
Ours	$O(L)$	$O(1)$	$O(1)$	0	$O(1)$

In this table, we provide comparison in terms of computation, storage and communication. Note that L is the number of devices and F is the average number of friends in the social graph as presented in the work of Nitti et al. [18].

formed by fog nodes and it significantly reduces the impact of cooperative ballot-stuffing attacks. Indeed, with a rate $1 - \lambda = 20\%$ of malicious nodes, the limit trust value of the target malicious node reaches the value 0.05. This value is small comparing to the limit value 0.27 obtained in the case where there is no countermeasure.

8.4 BC-Trust vs Existing solutions

Table 4 shows a comparison of our solution and two other solutions (presented in related works section) in terms of storage, computation and communication overhead. We notice that our protocol ***BC-Trust*** reduces storage related to trust values compared to other solutions. Indeed, in our protocol, IoT device stores only trust data related service to providers which are basically its own direct observations. The amount of this data is at most equal to $8 \times \times L$ which depends linearly on the number of IoT devices L if we assume that trust values are encoded on 4 bytes. However, in other approaches, the storage overhead depends quadratically on the number of IoT devices L since each device must keep the recommendation of other nodes against each service provider. Moreover, contrary to other approaches, ***BC-trust*** reduces computation overhead (few additions and multiplication) which is independent of the number of IoT devices. Finally, the communication overhead, measured as the amount of data exchanged during ΔT, is also reduced in our protocol. Indeed, IoT devices need to exchange only with fog nodes to get recommendation about one service provider, whereas in other solutions IoT devices must exchange the recommendations between each others.

We present in Table 5 a qualitative comparison of our proposal with some previously presented related works. Our solution is very convenient with high mobility scenarios

	Scalability	Mobility	Node-failure	QoS	Convergence time	Global view
[7]	-	-	+	-	-	-
[18]	+	+	-	-	+	+
[15]	-	-	+	-	+	-
[5]	-	-	-	-	+	-
[19]	-	-	+	+	+	-
[2]	-	-	-	-	+	-
[8]	-	+	-	-	+	-
[9]	+	-	-	+	+	-
[6]	+	-	-	+	+	-
Ours	+	+	+	+	+	+

TABLE 5: Comparison between trust management protocols

and resists against node failures. Furthermore, our solution is QoS-aware protocol which reduces the latency during the computation of trust values and allows IoT devices to filter out service providers with respect to some QoS metrics thanks to fine-grained based service property. Contrary to other approaches, ***BC-Trust*** introduces other original properties such as global view of trustworthiness information and scalability support which are very important in IoE.

9 Conclusion and Future Work

In this paper, we proposed a new decentralized trust management protocol for Internet of Everything in fog computing architecture. Our protocol is distributed and each IoT object can assess trustworthiness of service providers and share it among IoT devices in a scalable way. Based on blockchain technology, our protocol offers a global view on the trustworthiness of each service provider in the architecture. In addition, our solution introduces the fine grained concept in trustworthiness computation.

Moreover, contrary to most existing works, our proposal deals efficiently with high mobility scenarios thanks to blockchain technology. Besides, we demonstrated through experiments the resiliency and robustness of our solution in front of malicious attacks. Then, we showed that our solution outperforms the existing ones, especially in terms of saving computation and storage resources. In addition, we confirmed our experimental result through an advanced theoretical analysis about the convergence of trust values under different malicious attacks. Furthermore, we shed the light on cooperative attacks where we proposed an efficient

countermeasure based on the analysis of recommendations' history reported by IoT devices to the blockchain.

For future work, we plan to extend our proposed mitigation approach by developing more efficient offline algorithms for malicious nodes detection using machine learning techniques.

10 Acknowledgments

This work was carried out and funded in the framework of the Labex MS2T. It was supported by the French Government, through the program "Investments for the future" managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02).

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys Tutorials*, 17(4):2347–2376, Fourthquarter 2015.

[2] H. Al-Hamadi and R. Chen. Trust-based decision making for health iot systems. *IEEE Internet of Things Journal*, 4(5):1408–1419, 2017.

[3] F. Bao, 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. on network and service management*, 9(2):169–183, 2012.

[4] B. Carminati, E. Ferrari, and M. Viviani. Security and trust in online social networks. *Synthesis Lectures on Information Security, Privacy, & Trust*, 4(3):1–120, 2013.

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

[6] I. R. Chen, F. Bao, M. Chang, and J. H. Cho. Trust management for encounter-based routing in delay tolerant networks. In *IEEE Global Telecommunications Conf. GLOBECOM*, pages 1–6, Dec 2010.

[7] I. R. Chen, F. Bao, and J. Guo. Trust-based service management for social internet of things systems. *IEEE Trans. on Dependable and Secure Computing*, 13(6):684–696, Nov 2016.

[8] R. Chen, J. Guo, and F. Bao. Trust management for soa-based iot and its application to service composition. *IEEE Trans. on Services Computing*, 9(3):482–495, 2016.

[9] J.-H. Cho, A. Swami, and R. Chen. Modeling and analysis of trust management for cognitive mission-driven group communication systems in mobile ad hoc networks. In *Int. Conf. on Computational Science and Engineering*, volume 2, pages 641–650. IEEE, 2009.

[10] B. Cohen. Incentives build robustness in bittorrent. In *Workshop on Economics of Peer-to-Peer systems*, volume 6, pages 68–72, 2003.

[11] X. Fan, L. Liu, M. Li, and Z. Su. Grouptrust: Dependable trust management. *IEEE Trans. on Parallel Distributed Systems*, 28(4):1076–1090, April 2017.

[12] F. Gai, J. Zhang, P. Zhu, and X. Jiang. Multidimensional trust-based anomaly detection system in internet of things. In *Int. Conf. on Wireless Algorithms, Systems, and Applications*, pages 302–313. Springer, 2017.

[13] J. Guo, I. R. Chen, and J. J. P. Tsai. A mobile cloud hierarchical trust management protocol for iot systems. In *5th IEEE Int. Conf. on Mobile Cloud Computing, Services, and Engineering*, pages 125–130, April 2017.

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

[15] U. Jayasinghe, N. B. Truong, G. M. Lee, and T. W. Um. Rpr: A trust computation model for social internet of things. In *in Int. IEEE Conf. on Smart World Congress*, pages 930–937, July 2016.

[16] Z. Lin and L. Dong. Clarifying trust in social internet of things. *arXiv preprint arXiv:1704.03554*, 2017.

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

[18] M. Nitti, R. Girau, and L. Atzori. Trustworthiness management in the social internet of things. *IEEE Trans. on Knowledge and Data Engineering*, 26(5):1253–1266, May 2014.

[19] Y. B. Saied, A. Olivereau, D. Zeghlache, and M. Laurent. Trust management system design for the internet of things: A context-aware and multi-service approach. *Computers & Security*, 39:351–365, 2013.

[20] A. A. Selcuk, E. Uzun, and M. R. Pariente. A reputation-based trust management system for p2p networks. In *IEEE Int. Symposium on Cluster Computing and the Grid*, pages 251–258. IEEE, 2004.

[21] R. A. Shaikh, H. Jameel, B. J. d'Auriol, H. Lee, S. Lee, and Y.-J. Song. Group-based trust management scheme for clustered wireless sensor networks. *IEEE Trans. on parallel and distributed systems*, 20(11):1698–1712, 2009.

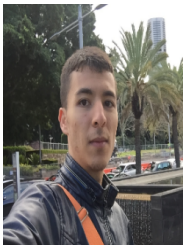
[22] Z. Su, L. Liu, M. Li, X. Fan, and Y. Zhou. Servicetrust: Trust management in service provision networks. In *IEEE Int. Conf. on Services Computing*, pages 272–279, June 2013.

[23] G. Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151:1–32, 2014.

[24] L. Xiong and L. Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Trans. on Knowledge and Data Engineering*, 16(7):843–857, 2004.

[25] L. T. Yang, B. Di Martino, and Q. Zhang. Internet of everything. *Mobile Information Systems*, 2017, 2017.

Djamel Eddine Kouicem is a PhD student at the University of Technology of Compiegne (UMR CNRS 7253) since October 2016. In September 2016, he gained a MSC diploma in Computer Science from Pierre Marie Curie University (Paris 6) in France. In July 2015, he received Engineer Diploma in computer science from the High National School of Computer Science in Algiers. His research interests are in security and privacy in internet of things and Networking.



Youcef Imine is a PhD. student at the University of Technology of Compiègne (France) under the supervision of Pr. Abdelmadjid Bouabdallah, where he works on "Cloud Computing security". In june 2016, he received his master degree in Networking and distributed systems from Abou Bekr Belkaid University (Algeria). His main works concern data security in emerging technologies such as Cloud and Fog computing, which include several security challenges such as data confidentiality, authentication, and cryptographic access control.

Abdelmadjid Bouabdallah received the Master (DEA) degree and Ph.D. from university of Paris sud Orsay (France) respectively in 1988 and 1991. From 1992 to 1996, he was Assistant Professor at university of EvryVal d'Essonne (France) and since 1996 he is Professor at University of Technology of Compiegne (UTC), where he is leading the Networking and Security research group and the Interaction and Cooperation research of the Excellence Research Center LABEX MS2T. His research Interest includes Internet QoS, security, unicast/multicast communication, Wireless Sensor Networks, and fault tolerance in wired/wireless networks. He conducted several large scale research projects funded by Motorola Labs., Orange Labs., ANRRNRT, CNRS, and ANR-Carnot.

Hicham Lakhlef is an associate professor at the University of Technology of Compiegne (UMR CNRS 7253). During the year 2015/2016 he was a temporary researcher in IRISA, University of Rennes 1 (UMR CNRS 6074) in France. During the year 2014/2015 he was a temporary teaching assistant and researcher at the University of Franche-Comté/FEMTO-ST institute (UMR CNRS 6174) in France. He co-authored more than 30 international publications. He obtained his Ph.D degree from the University of Franche-Comté in 2014 in France. His research interests are in parallel and distributed algorithms, WSNs, security, self-reconfiguration, routing, and internet of things.