A Robust and Adaptive Trust Management System for Guaranteeing the Availability in the Internet of Things Environments

Xu Wu[1, 2]
[1]School of Computer, Electronics and Information, Guangxi University
Nanning, Guangxi 530004, P.R. China
[2]Key Laboratory of Multimedia Communication and Network Technology, Guangxi University
Nanning, Guangxi 530004, P.R. China
[e-mail: xrdz2006@163.com]
*Corresponding author: Xu Wu

Received Septebmer 21, 2017; revised November 15, 2017; accepted December 23, 2017; published May 31, 2018

Abstract

Trust management is one of the most challenging issues for the highly heterogeneous Internet of Things (IoT). In the context of the IoT, it is difficult to evaluate the node's trustworthiness in the same trust model when a node provides different services. Guaranteeing the availability of the trust management service is another significant challenge because of the dynamic nature of IoT environments. With these issues in mind, this paper propose a robust and adaptive trust management system for the IoT that is able to measure the trustworthiness of nodes based on feedbacks collected from participants in a specific context and ensure the availability of trust management services. The main contributions of our system are: 1) Proposing a partly decentralized trust management framework, which improves the resiliency of the trust mechanism; 2) Proposing an adaptive trust evaluation scheme and a three-dimensional context representation makes trust evaluation more accurate and specific; 3) Enhancing the adaptive trust evaluation scheme by incorporating a bad behavior factor in trust estimation, which efficiently distinguishes misleading feedbacks from On-Off attacks. Simulation results show the good performance of the proposed system and especially show effectiveness against On-Off attacks compared to other trust mechanisms.

Keywords: Internet of Things, trust model, trust management, adaptive, On-Off attacks

1. Introduction

There are a large number of smart sensor nodes in the Internet of Things (IoT), which provide information and application services to end users through communication network protocols and unique addressing schemes [1]. These sensor nodes in IoT often are resource-constrained nodes, so they have a greater need to collaborate with one another for providing advanced service and applications. For example, the car driver wants to know about the situations of roads towards her destination. Thus, the sensors installed in her car will request a collaborative task with other sensors met along the road. However, building the collaborative task may make nodes exposed to certain types of malicious attacks. Thus, constrained IoT nodes have a greater need to collaborate with one another in order to establish secure communications or to resolve coverage and packet delivery problems. For these reasons, some techniques are being proposed for many networking services in the field of modern wireless communications. Examples of such technologies are presented in [18-22].

To best satisfy the collaborative service requester and maximize application performance, it is crucial to develop a trust service platform to evaluate the trust between nodes in IoT. Trust has been defined and considered from different perspectives based on multi-trust metrics [2]. The basic idea of trust management is to establish trust between two individual nodes. Trust management is a mechanism that also allows identifying malicious, selfish, and compromised nodes. Much research [1] has been done to deal with trust management issues in IoT environments.

Yet, most of these trust management method don't focus on investigating the multiservice characteristic of IoT [6]. In IoT environments the smart nodes can provide different types of services by using their different resources. In the context of the IoT, it is difficult to evaluate the node's trustworthiness in the same trust model when a node provides different services.

In addition, guaranteeing the availability of the trust management service is another significant challenge because heterogeneous sensor nodes in the IoT are vulnerable to attacks, and distributed in different communities. Hence, nodes may lack the motivation to provide reliable trust feedbacks; instead, malicious ones may intentionally give misleading feedbacks to specific victims in order to fake their decisions. Until now, huge amount of work about trust mainly focused on defining and evaluating the trust relationships among nodes and proposing the trusted frameworks and algorithms; rather than the development of a robust model for ensuring the availability of trust management service.

With these issues in mind, this paper aims at developing a robust trust management system that evaluates the trust between two nodes and ensures the availability of trust management service in IoT environments. In particular, we distinguish the following key issues of the trust management in IoT:

Context-aware and multiservice approach: In a multiservice IoT, nodes can provide different cooperative services. Nodes can have dynamic interactions with other nodes, which may involve different cooperative services. Each cooperative service has nodes' resource consumption related to a different cost. Undoubtedly, a trust management for the IoT should consider a multiservice approach, where trust is context sensitive. The trust evaluation is based on how well the entity will behave for providing certain service in a specific context.

Collaborative Services Protection: It is not unusual that one collaborative service experiences internal attacks from its partners. When collaboration happens, a malicious

partner can easily launch an internal attack by giving multiple misleading feedbacks (i.e., collusion attacks) or by behaving well and badly alternatively (i.e., On-Off attacks) [14]. Indeed, the detection of such malicious behaviors poses several challenges. Firstly, new nodes join and old ones leave IoT environment at any time. The mobility of the nodes makes the detection of malicious behaviors (e.g., feedback collusion) a significant challenge. Secondly, on-off attacking nodes behave well and badly alternatively. The on-off attacking behavior is similar to the behavior of a malfunctioning node. So it is difficult to distinguish the on-off attacking nodes from the malfunctioning nodes.

Trust Management Service's Availability: A trust management service can provide effective trust management to measure the trustworthiness of node from its past behaviors when it is selected as a partner. However, guaranteeing the availability of trust management system is a difficult problem due to the characteristics of heterogeneity and multiservice in IoT environments. The approaches satisfying the specific requirements of IoT are appropriate in IoT environments. Trust management system should be adaptive and flexible in IoT environments.

Therefore, we propose a novel trust management system for the IoT that is able to measure the trustworthiness of nodes based on feedbacks collected from participants in a specific context and ensure the availability of trust management services. Depending on the system, a requesting node can select the best partners to provide collaborative service. Our system exploits techniques to ensure the availability of trust management service. The main contributions of our system are:

An adaptive System. Providing dynamic trust evaluation for certain node is an important requirement to the trust management service. Therefore, we propose an adaptive trust evaluation scheme, where several contextual metrics make trust evaluation more accurate and specific. Unlike previous work such as [14, 15], we choose the service, capability and community interest as the main contextual metrics to evaluate the trustworthiness of nodes in the different contexts. In addition, we also measure context similarity investigated in [14], but our method is a three dimensional context representation instead of a two dimensional representation.

A Robust System. It is difficult to identify the truly malicious nodes in the context of the IoT. Sometimes a benevolent node might behave bad temporarily because of unexpected accidents. For example, a node might temporarily unable to provide assistance to their peers due to exhaustion of their resource capabilities. In the above circumstance, a malfunctioning node is often qualified as malicious node. In order to solve the problem, we further enhance our proposed adaptive trust evaluation scheme by incorporating a bad behavior factor. Our method efficiently distinguishes misleading feedbacks from On-Off attacks.

An Availability System. The trust management service in IoT environments has high availability requirement. The trust information computed in a full decentralized approach would result in communication overhead and consuming the limited resource of sensor nodes. Though a centralized server can solve the problem of communication overhead, it has the shortcoming of single point of failure. If the server is controlled by malicious nodes, the entire trust management system will collapse. We combine the advantages of centralized and distributed approaches and propose a partly decentralized trust management framework, where several power nodes covering different communities of IoT are spread to handle trust computational load in a decentralized way.

This paper is organized as follows. Section 2 describes related work. Section 3 discusses the proposed partly decentralized trust management framework. Section 4 presents the proposed

adaptive trust evaluation scheme. Section 5 describes the test scenario and simulation results. Finally, we conclude with a summary of our results and directions for new research in Section 6.

2. Trust Management for the Internet of Things

In the literature, many trust management frameworks and methods are proposed in order to solve trust issues in IoT environments.

Basically, the basic idea of trust management mechanisms is to measure, build and manage the trust relationships between smart objects in IoT environments. There exits four design dimensions of the trust computation techniques in [7]: trust composition, trust propagation, trust aggregation and trust update. The advantages and shortcomings of each dimension's options are analyzed. The authors also show the effectiveness of trust computation techniques in terms of resisting malicious attacks. **Table 1** shows the recently work about trust management for the Internet of Things.

Table 1. Recently work about trust management for the Internet of Things

Trust management model or method	Management framework	shortcoming
[4]	centralized	Not indentifying malfunctioning nodes
[5]	decentralized	Not indentifying malfunctioning nodes
[6]	Centralized	Using ontology in the method isn't a suitable solution
[9]	decentralized	Not taking into account the characteristic of resource limited nodes
[10]	decentralized	Not indentifying malfunctioning nodes
[11-12]	centralized	Encouraging to send dishonest trust recommendations
[15]	centralized	Not a lightweight mechanism

A group-based and collaborative method [4] is proposed in order to ensure the security of RFID systems in IoT environments. The proposed method focuses on offering adaptability and scalability to support the application of RFID systems. In addition, it also detects incorporated malware to provide an extra protection. The simulation results show that the method has better efficiency than other protocols.

The authors in [10, 11] study the different metrics including its collaboration to offer a recommendation in a trust management system, and discuss the service collaboration problem. Yet, there are some disadvantages in the method. For example, a node may be encouraged to send dishonest trust recommendations in order to getting a high trust value.

In [9], the proposed trust management methods in IoT are evaluated based on three parameters including trust management protocol, scalable solutions and context-aware assessment. The paper comprehensively analyzes these methods from the three perspectives. The results show that it is a future research direction to build a scalable and context-aware trust management system in IoT environments.

The work in [2] is a very recently work on trust management in IoT environments. A trusted service platform is established, which provides trust evaluation based on three trust metrics. These metrics include Reputation, Recommendation, and Knowledge. The idea of the

proposed method comes from modeling human trust relationship. However, using ontology in the method isn't a suitable solution due to the characteristic of limited resource of sensor nodes. Moreover, the authors don't explain how their protocol can face trust related attacks and don't propose solution to ensure the scalability of the IoT network.

The work in [8] presents a trust model, where a reputation score is related to the object and stored in its own database. The reputation score is managed by a server. This model uses a punishment way to decrease the malicious behaviors of nodes. However, the main shortcoming is that the characteristic of resource limited nodes are not taken into account in the method.

The work presented in [5, 14] considers a context-aware and multiservice approach for IoT. The trust in [10] is computed in a centralized trust manager. The simulation experiments are done under different attack environments. Compared with [14], the authors [5] propose a distributed trust management scheme, which assigns different scores to the collaborative nodes and the malicious nodes. The proposed scheme is evaluated under on-off attacks. We also consider that a multiservice scenario is prevalent in IoT. Our proposed scheme is inspired by [5, 14], but we use a partly decentralized approach instead of a full distributed strategy or a centralized approach.

In the literature, trust management issues have long been investigated in Mobile Ad hoc Networks (MANETs) and Wireless sensor networks (WSNs). Yet, the proposed approaches lack flexibility and adaptability to both the specific requirements of IoT and all these trust management schemes don’t deeply investigate how to identify truly malicious nodes from malfunctioning nodes. Specifically, it is difficult to differentiate between the malfunctioning nodes and the on-off attacking nodes [12, 13]. The behavior of a malfunctioning node can be similar to behavior of an on-off attacking node. An example is where a node always reports correct feedback data but might sometimes also reports incorrect feedback data due to a computation error. Thus, the malfunctioning node is qualified as malicious. In addition, a malicious node might keep good behaviors in ordinary situations; but make bad behaviors in important circumstances such as a large scale trade et al., so it is hard to be found.

Based on new IoT requirements and identified shortcomings of the related work,we propose a partly decentralized trust management framework for the IoT that is able to induce from nodes past behaviours in distinct cooperative services how much trust can be put into a node for accomplishing a required task. Eventually, only the best partners with respect to a sought cooperative service are proposed to a requesting node. In this work, we further enhance our proposed adaptive trust evaluation scheme by incorporating a bad behavior factor in order to efficiently distinguish misleading feedbacks from On-Off attacks.

3. The Partly Decentralized Trust Management Framework

The main aim of the proposed solution is to design an available trust management system which manages cooperation in a heterogeneous IoT architecture involving nodes who provide different service. The way of trust management framework such as centralized or decentralized way should be considered carefully before designing such a trust management system. A full decentralized trust management system would bring the problem of communication overhead. Also the resource limited nodes haven’t enough memory to store the trust information. In IoT scenarios, most nodes are limited resource sensor nodes which have limited computing power, memory, radio range and battery. In a centralized strategy, the system usually used a trust management server to solve the problem of communication load.

Yet it has the shortcoming of single point of failure. We combine the advantages of centralized and distributed approaches in the paper. Thus, the proposed system is a partly decentralized trust management framework based on feedbacks collected from participants. In particular, the trust management system spread several power nodes managing feedback given by nodes handle trust computational load in a decentralized way. The power nodes expose interfaces to sensor nodes, so that sensor nodes can give their feedbacks or inquire the trust results. **Fig. 1** depicts the framework, which consists of two different layers, namely the IoT Node Layer and the Trust Management Service Layer.

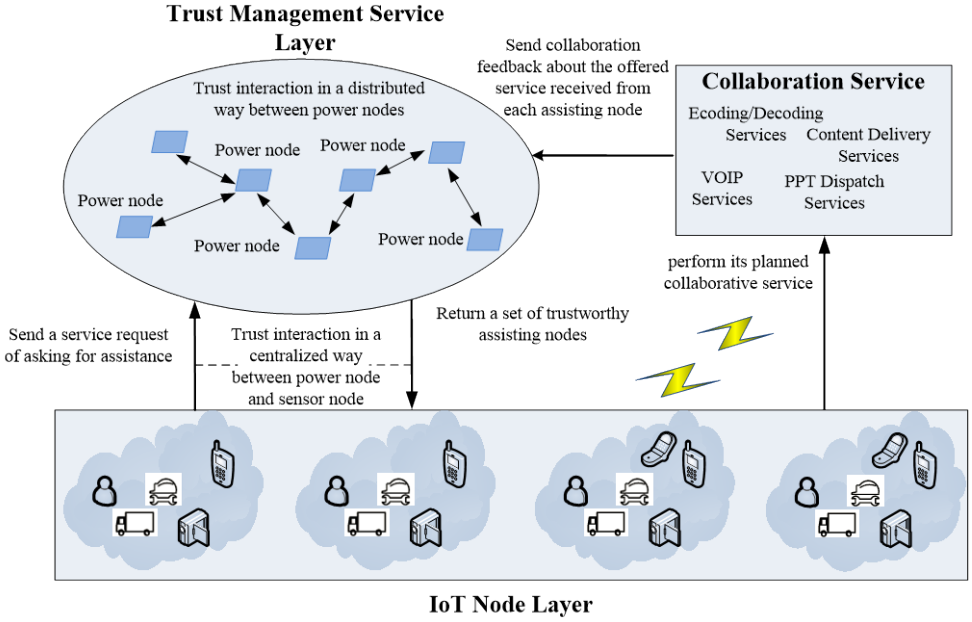


Fig. 1. Architecture of partly decentralized trust management framework

The Trust Management Service Layer. This layer consists of several distributed power nodes, which are hosted in different IoT communities because the sensor nodes in similar communities likely collaborate with each other for completing a collaborative task. These power nodes communicate with each other in a full distributed way. The sensor nodes send trust feedback and inquiry trust value with power node of their own community in a centralized way. Upon receiving a collaboration request from node of its own community, the power node would start the partner selection process and return some trusted assisting nodes to the requester. Interactions for this layer include: 1) receiving collaboration request which come from node of its own community; 2) selecting and returning the potential candidates to the requester with an adaptive trust evaluation scheme; 3) updating the trust of assisting nodes and storing their trust information.

The IoT Node Layer. This layer consists of resource limited nodes. In the layer, these resource limited nodes would collaborate with others in order to finish a common goal. Interactions for this layer include: 1) sending a collaboration request to the power node of its own community; 2) giving collaborative feedbacks to the power node.

The different phases of partly decentralized trust management framework are presented in the following:

Initialization phase: All nodes are grouped into different community based on community interest [3]. At the beginning, there isn't any feedback information. In order to solve the problem, a few collaboration tasks are assumed between some nodes, then the feedback information from requesting nodes are stored in power nodes and used as trust evaluation evidence.

Partner selection phase: the phase is similar to [15], where a resource-limited node will send a collaboration request to its power node. Upon receiving the collaboration request, the system goes into the partner selection process and returns some trusted assisting nodes to the requester. It is crucial to develop a mechanism that helps determine the optimal number of power nodes because more nodes residing at various communities means higher overhead (e.g., cost and resource consumption) while lower number of nodes means less availability. In the paper, we use the workload threshold $e_w(N_{pn})$ that can automatically adjust the number of power nodes based on the power workload factor. The power workload for a particular power node is presented as $P(N_{pn})$. It is calculated as the mean of the Euclidean distance (i.e., to measure the distance between a particular power node workload and the mean of the workload of all power nodes) and the power node workload (i.e., the percentage of trust feedbacks handled by this node) as follows:

$$P(N_{pn}) = \frac{1}{2} * \left(\sqrt{\left(\frac{\gamma(N_{pn})}{\gamma(all_{pn})} - \frac{\gamma(mean_{np})}{\gamma(all_{np})}\right)^2} + \frac{\gamma(N_{pn})}{\gamma(all_{pn})}\right) \tag{1}$$

where the first part of the equation represents the *Euclidean distance* between the workload of node N_{pn} and the average workload of all nodes where $\gamma(mean_{np})$ denotes the mean of feedbacks handled by all nodes. The second part of the equation represents the ratio of feedbacks handled by a particular node over $\gamma(N_{pn})$ the total number of feedbacks handled by all nodes $\gamma(all_{np})$. Each node hosting a power node instance reports its power workload. The number of power nodes $Number_{pn}$ is adjusted as follows:

$$Number_{pn} = \begin{cases} Number_{pn} + 1 & if\ P(N_{pn}) \geq e_w(N_{pn}) \\ & or\ \ Number_{pn} < 1 \\ Number_{pn} & otherwise \end{cases} \tag{2}$$

Trust updating phase: Once the requester gives the interaction feedbacks to the power node, the system would update the trust of the assisting nodes. Finally, the system stores the trust information of assisting nodes.

4. Adaptive Trust Computing Scheme

In this paper we propose an adaptive trust evaluation scheme for social IoT systems. The adaptive trust evaluation scheme can dynamically evaluate the trust of node in different contexts. There exists a wealth of trust metrics available in IoT systems, but we choose service, capability and community interest as the main metrics due to the features of IoT architecture. In the scheme, a three-dimensional context representation makes trust evaluation more accurate and specific. **Fig. 2** shows the details of three-dimensional context representation. X-axis, Y-axis and Z-axis denotes service, capabilities and community interest, respectively.

the three-dimensional context model measures context similarity between the previous collaboration service and the present requested collaboration service.

4.1. Adaptive Trust Evaluation

The service trust property represents the circumstances of executed service. It can be used to discriminate the interactions and consider trusted a node only for a certain type of service. We select service as a trust property because a trusted node for a particular collaboration task may not be reliable for other collaboration tasks in an IoT system. It is indeed important to know in which scenario the interaction feedback has been obtained in an IoT application. A node depends on the execution time and the status of the measured node to evaluate the service trust property of another node.

The capability trust property represents the resource amounts of candidate node that is needed to provide a collaboration service. When the node provides assistance for different collaboration service, resource-demanding requirements should be differentiated. The resource-demanding of a node is different for a large scale trade and an ordinary scale trade. Node capabilities may be measured from a multifaceted perspective such as processing power, memory and energy level. In the paper, energy consumption is used to quantify node capabilities.

Generally, the community interest is related to social relationship of node (e.g. co-location or co-work relationships [3]). Nodes are in the same social communities likely have similar community interest. Therefore, two nodes with similar community interest trust have high probability of collaborating with each other, and thus can bring a better service performance.

The trust of a node is calculated by the weighted sum of the interaction feedback value received. The interaction experience value is issued by a node for the collaboration service provided by the assisting node. We refer to the interaction feedback E_{kj}^{m}, i.e., the node k towards node j, where m = service, capability and community interest. The value can be expressed in a binary way ($E_{kj}^{m} \in [1,0]$, i.e., where 1 indicates complete trust, and 0 distrust).

The most important interaction feedbacks are those that come from the same context with requesting collaboration service of node i. So the system would only collect the interaction feedbacks about node j from those that have similarity of service, capabilities and community interest with requesting collaboration service. We would measure context similarity between the previous collaboration service and the present requested collaboration service. **Fig. 2** shows the interaction feedback history of node j. X-axis, Y-axis and Z-axis denotes service, capabilities and community interest, respectively. In **Fig. 2**, previous interaction feedbacks E_{kj}^{m} (S_k, C_k, I_k and V_k) are stored at the trust management system, given by all nodes k evaluating the quality of service provided by a common assisting node j.:

S_k : Service provided by node j

C_k : Capability of node j

I_k : Community interest of node j

V_k : Satisfaction score given by any node k to j for evaluating the offered service. Node k rates 1 if it is satisfied with the service and 0 otherwise.

The X-axis on the graph shows the different service provided by node *j*. The Y-axis and Z-axis respectively shows the capabilities and community interest of the node *j* when collaborating for these services. Each graph is characterized by the requesting interaction feedback $E^{m}_{request}(S_{request}, C_{request}, I_{request})$ depicted as a black diamond on **Fig. 2**. $E^{m}_{request}$ refers to the interaction feedback about the present requested service, which would be issued by node *j*, after the node *j* provides the requested collaborative service. $S_{request}$ is the present requested service provided by node *j*. $C_{request}$ is the present requested capability of node *j*. $I_{request}$ is the present requested community interest of node *j*. In order to measure context similarity between the previous interaction and the present requested interaction, the Euclidean Distance is computed between E^{m}_{kj} and $E^{m}_{request}$.

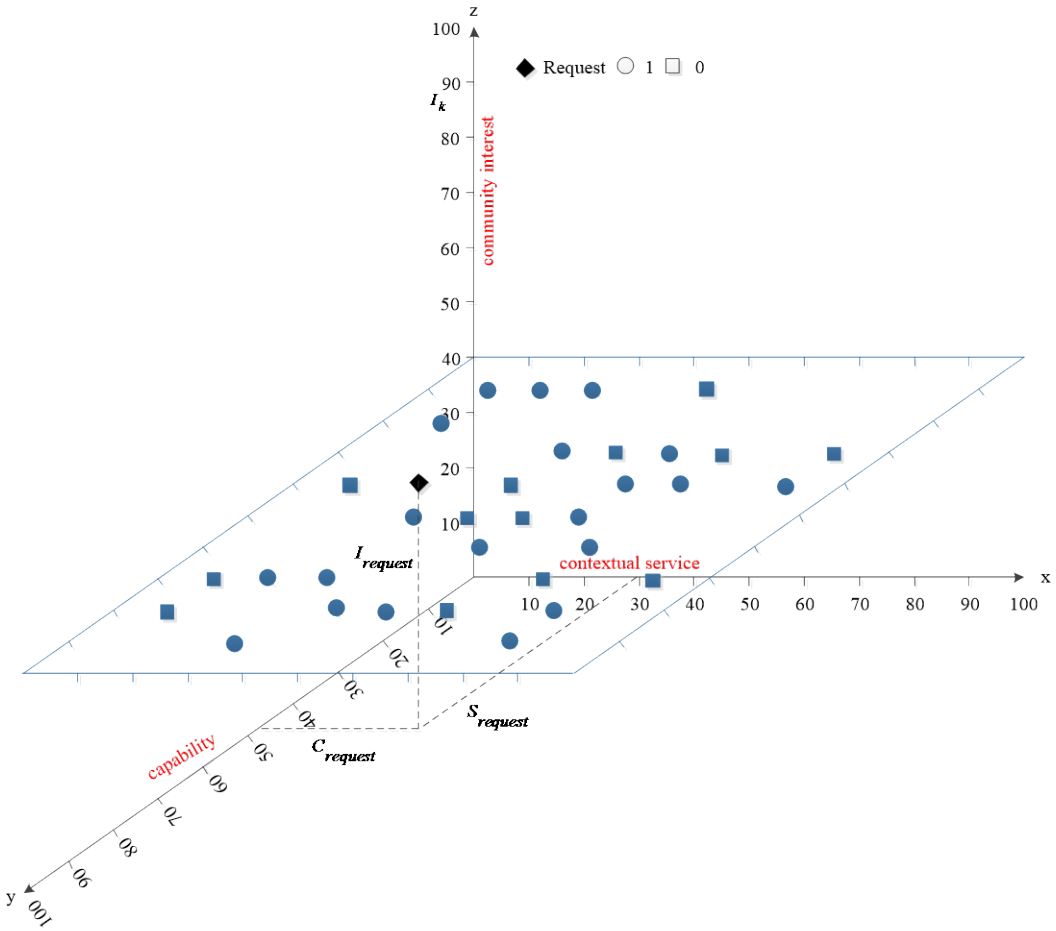


Fig. 2. Three dimensional trust property representation of node *j*

The distance is denoted as d_{kj}. We then obtain d_{kj} as

$$d_{kj} = \sqrt{(dS_k^2 + dC_k^2 + dI_k^2)} \tag{3}$$

where dS_k , dC_k and dI_k is presented as the difference between $S_{request}$ and S_k , between $C_{request}$ and C_k , and between $I_{request}$ and t I_k , respectively.

$$dS_k = \left| S_{request} - S_k \right| \tag{4}$$

$$dC_k = \left| C_{request} - C_k \right| \tag{5}$$

$$dI_k = \left| I_{request} - I_k \right| \tag{6}$$

This computation is well measuring the similarity of a previous collaboration service to a present requested collaboration service.

Finally, we adjust d_{kj} distance as follows: a retained interaction feedback of node *j* should have a distance $d_{kj}(E_{kj}^m, E_{request}^m) < \chi$. χ is an adjustable threshold, which decides the number of collected feedbacks. Compared with [14], our computing method of context similarity is a lightweight mechanism that fits resource-limited nodes in IoT environments.

Finally, the system can combine the collected interaction feedbacks about the node *j*. The global trust R_j^m of the node *j* is eventually obtained as follows:

$$R_j^m = \sum_{k \in S} \left(\frac{\beta_k}{\sum_{k \in s} \beta_k} E_{kj}^m \right) = \frac{\sum_{k \in s} \beta_k E_{kj}^m}{\sum_{k \in s} \beta_k} \tag{7}$$

where R_j^m is the trust value of node j . *S* is the set of node k with whom node *j* has conducted collaboration interactions and satisfies with $d_{kj} < \chi$. E_{kj}^m is the feedback of node *j* rated by peer *k*, and β_k is the aggregation weight of E_{kj}^m and to consider trust decay over time. A node may change its behavior over time: recent interaction feedbacks are thus more meaningful than feedbacks obtained for a long time. The aggregation process runs multiple iterations until each R_j^m converges to a stable trust rating for node *j*. The algorithm of calculating the trust value of node *j* is shown in **Fig. 3**.

Input: the requesting interaction feedback $E_{request}^m$ $(S_{request}, C_{request}, I_{request})$
Output: the trust value R_j^m of node *j*
1. Calculating the Euclidean Distance d_{kj} between E_{kj}^m and $E_{request}^m$;
2. Deciding the number of collected feedbacks based on $d_{kj}(E_{kj}^m, E_{request}^m) < \chi$
3.Combining the collected interaction feedbacks about the node *j*
4. runing multiple iterations until each converges to a stable trust rating for node *j*
5. Getting the trust value R_j^m of node *j*;

Fig. 3. The algorithm of calculating the trust value of node *j*

4.2. Adaptive Trust Evaluation

The proposed trust management system assesses the services provided by node in different contexts. But, it is difficult to identify the truly malicious nodes in the trust management system. The previous examples show that behavior of a malfunctioning node can be similar to behavior of an on-off attacking node. A malfunctioning node is often qualified as malicious node. Yet, the frequency of misbehavior is different between a malfunctioning node and a malicious node. A malfunctioning node's behavior is random and temporary. The behavior of a malicious node is persistent and intentional. Hence, a truly malicious node can be detected through computing the frequency of misbehavior. In [16], the authors discuss the method of computing the frequency of misbehavior, where a time window mechanism is used. But evaluation results [16] show the proposed scheme is sensitive to the changes from bad behaviors to good behaviors. Moreover, the authors consider malfunctioning nodes as malicious nodes, but in fact there are essential differences between malfunctioning nodes and malicious nodes. Therefore, in the paper, we revise the computing method, and measure the frequency of misbehavior based on the duration of on and off behaviors rather than the number of on and off periods. The simulation results in section 6 show that it is more accurate to determine the node's status with the duration. The frequency of misbehavior is measured as follows:

$$f_{t_n} = \frac{Z_{t_n}}{Z_{t_n} + H_{t_n}} \tag{8}$$

where Z_{t_n} and H_{t_n} are the duration of on and off behaviors during t_n. Then, for time window t_n, we can determine the state of node with equation (8):

$$Q(f_{t_n}) = \begin{cases} 1 \ \ malicious\ node\ or\ malfunctioning\ node \\ (0;\psi) \ \ benevolent\ node \\ (\psi;1) \ \ malicious\ or\ on-off\ \ attacking\ node \end{cases} \tag{9}$$

Combining ψ_{t_n} and f_{t_n} obtains bad factor B_{t_n}, which is presented as:

$$B_{t_n} = \begin{cases} 1-\psi_{t_n} \ if \psi_{t_n} > f_{t_n} \\ \omega \times (1-f_{t_n}) + (1-\omega) \times (1-\psi_{t_n}) \ \ otherwise \end{cases} \tag{10}$$

t_n is time window. ψ_{t_n} is the weight of misbehavior. It is obtained based on the rate of misbehavior in each time unit [17]. ω is the weight given to the frequency and weight of misbehavior. Then, we set up time window t_n is the sum time of collecting feedbacks about node *j*, and B_{t_n} is incorporated into the Equation (7), we get:

$$R_j^m = \sum_{k \in S} \left(\frac{\beta_k}{\sum_{k \in s} \beta_k} E_{kj}^m B_{t_n} \right) = \frac{\sum_{k \in s} \beta_k E_{kj}^m B_{t_n}}{\sum_{k \in s} \beta_k} \tag{11}$$

Finally, the adaptive trust estimation scheme is enhanced based on a bad behavior factor,

which can help to identify truly malicious behavior of nodes and prevent possible On-Off attacks to a multiservice IoT.

5. Experimental Analysis

In this section, in order to evaluate the effectiveness of the proposed trust management, in identifying malicious nodes and evaluating collaboration service, a series of test scenarios are developed. We will study the effect of β_k and misbehavior detection in computing trustworthiness of a node. Comparisons were done with TMP [11], DTMS [5] and RTES [16]. Experiments were run using the ns-3 simulator [22]. ns-3 is a discrete-event network simulator for Internet systems, targeted primarily for research and educational use. ns-3 is free software, licensed under the GNU GPLv2 license, and is publicly available for research, development, and use.It is easier to simulate interaction patterns and behaviors than other simulation tools. The simulation configuration in **Table 2** was used.

Table 2. Default simulations parameters in the experiments

Parameter type	Value
Number of sensor nodes	200
Type of services	[1, 6]
Number of time units in time window	L=3
Trust threshold	0.6
Threshold for rate of misbehavior	ψ =0.1
The rate of malicious nodes	0%-90%
The rate of benevolent nodes	0%-50%
Energy level	[0-100]
Number of communities	[1, 10]
Trust and misbehavior frequency and weight estimation period	Δ
Simulation time	100 Δ

In our experimental setup, each node dynamically changes its status.

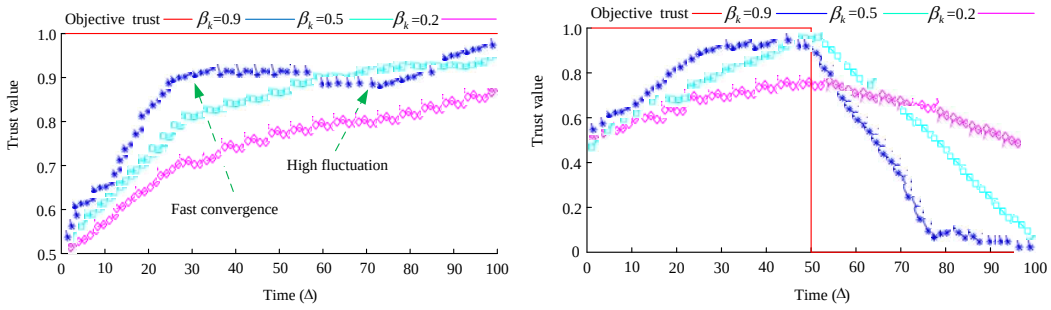
In order to evaluate trust establishment under node misbehavior, we first need to define the threat model. In the model, the behavior of node includes behavior of a benevolent node, a malicious node and a malfunctioning node. Hence, in this section, we define general and basic notions about benevolent, malicious and malfunctioning nodes, and we model node behavior. The percentage of malicious nodes [10%-90%] is randomly selected out of all sensor nodes. We consider an IoT environment with 200 heterogeneous sensors with all of them providing honest services. These sensors are randomly distributed to different community based on their community interest. Our system allows dynamic community lists. The community list kept by each sensor is simulated initially and remains the same throughout the simulation. Sensor nodes are in one or more communities. A sensor can belong to up to 10 communities. This is also simulated and remains fixed throughout the simulation. We assume that the interaction frequency is 6 times every 1 hour. The total simulation time is 100 hours. The average interaction-contact is 2 hours. Every node can provide up to 6 different services. The initial trust value of all sensor nodes is set to a trust level of 0.5. Node capabilities are quantified based on energy consumption.

We consider a situation where the trust level of a dishonest node is evaluated with respect to a resource demanding service. We can see that this node, being considered under a global trust value, manages to hide its misbehaviour when performing this service. It maintains an overall high trust level (red graph) since it compensates received bad scores with good scores obtained for its good behaviours in simpler services.

5.1. Effect of β_k on trust evaluation

We first investigate the effect of design parameter β_k on trust evaluation. β_k is the weighting factor and to consider trust decay over time in Equation 11.

In order to analyze the effect of β_k, we select 10 very recently feedback and vary β_k by selecting different values (0.2, 0.5, and 0.9). The percentage of malicious nodes P_m is 35%. **Fig. 4(a)** shows the effect of β_k on trust evaluation toward a "benevolent" node whose objective trust status keeps steady value as time increases. The objective trust status for this good node is constant at 1. We can see that as the value of β_k increases, the trust value converges to objective trust faster, but there exists more high trust fluctuation. Here we observe that the trust convergence time is 30 to 70 Δ because the average inter-arrival interaction time is set to 20 Δ. **Fig. 4(b)** shows the results of trust evaluation toward a "malicious" node randomly selected. The state of node changes from benign to malicious after 50 Δ. We can see that once the state varies, the trust evaluation converges towards the new objective trust status. In addition, as the value of β_k increases, the trust evaluation converges to the new objective trust status faster, and there exists more high trust fluctuation. This result shows that our trust evaluation scheme is adaptive to the time changes, and exactly reflects the actual trust state of node.



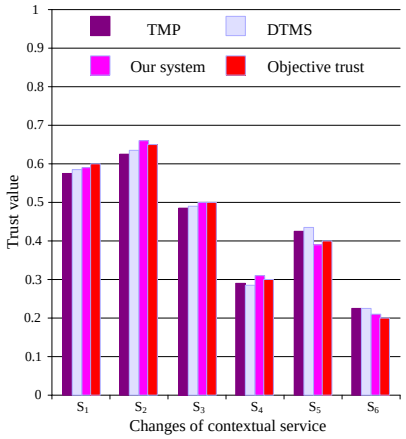
(a) Trust of a benevolent node randomly picked (b) Trust of a malicious node randomly picked

Fig. 4. Effect of β_k on trust evaluation

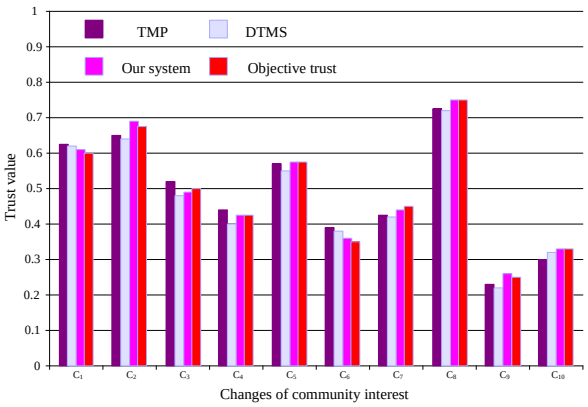
5.2. Effect of m on trust evaluation

Fig. 3 shows that the trust evaluation quickly converges and it is remarkably close to the objective trust status. We further investigate the effect of m on trust evaluation, where m = service, capability and community interest. We vary service and community interest by selecting 6 different services ($service_1$, $service_2$…, $service_6$), and 10 different communities ($community_1$, $community_2$…, $community_{10}$). The changes of capability are in the range of [0-100]. Comparisons were done with TMP [11] and DTMS [5].

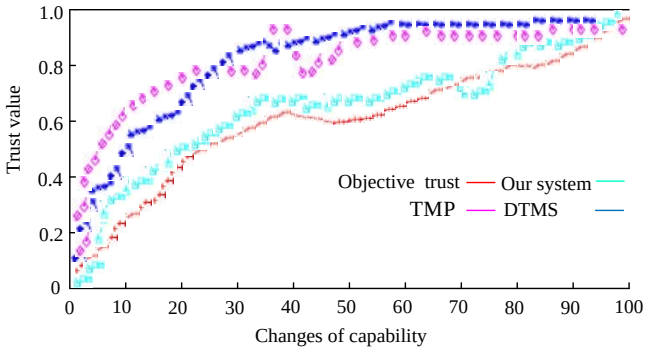
Fig. 5(a), **5(b),** and **5(c)** show trust evaluation results of a benevolent node randomly picked toward another benevolent node also randomly picked for service, community, and capability, respectively. We further validate resiliency of our adaptive trust evaluation scheme toward changes of contexts in IoT environments with different simulation circumstances. Our trust management system outperforms all other trust mechanisms with Equation (3) and it is remarkably close to the objective trust status (marked with red color) with acceptable mean absolute error less than 10%. Our adaptive trust evaluation scheme can dynamically evaluate the same node by considering the changes of service, community interest and capability.



(a) m =service



(b) m =community interest



(c) m =capability

Fig. 5. Effect of m on trust evaluation

5.3. On-off attack detection

In this section, we evaluate our trust scheme under on-off attacks. Comparisons were done with DTMS [5] and RTES [16]. **Table 3** shows the parameters of on-off attacks. In order to make the simulation more realistic, we add a number of malfunctioning nodes to the environment. These nodes behave well for the 10 first Δ. Then become stuck malfunctioning for the second 10Δ and reverts to normal for the followed time Δ. We dynamically add the number of malfunctioning node and malicious node being 5 and 20 at every 10Δ. The number of malicious node is increased from 20 to 200. At the beginning, the number of all nodes is 200.

A malicious node changes its behavior alternatively: in the experiment, we intentionally add the frequency of malicious behaviors, but at the time, the number of malicious behaviors is decreased. Moreover, we randomly distribute on and off behaviors over time. **Fig. 6** shows the detected percentage ρ_m of malicious nodes. ρ_m is measured as follows:

$$\rho_m = \frac{n_m}{n_{all}} \tag{12}$$

where n_m and n_{all} are the number of detected malicious nodes and all nodes in the network. An important observation from **Fig. 6** is that, even when malicious nodes add to 140, our system and RTES [16] remarkably close to the objective detection result. However, compared with RTES [16], the acceptable mean absolute error of our detection result (marked with red color) is less than 10%. Because RTES is sensitive to the changes from bad behaviors to good behaviors, and more malfunctioning nodes are often qualified as malicious nodes.

Table 3. Parameters to simulate an on-off attack

Parameter type	Value
Probability of an on period	0.6
Probability of an off period	0.4
Number of good behavior at on period	10 to18
Number of good behavior at off period	10 to 18
Number of bad behavior at on period	12 to 16
Number of bad behavior at off period	12 to 16

Specifically, detection results of other mechanisms in the attack prove it is necessary to include the bad behavior factor $B_{t_n}^{mn}$ in trust estimation. As **Fig. 6** illustrates, the proposed mechanism with Equation (8) and (11) outperforms other two trust mechanisms. We assume that the malicious nodes will not regain trust during 100 Δ.

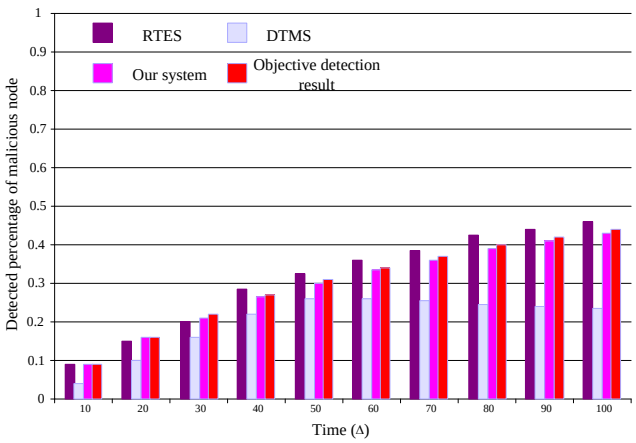
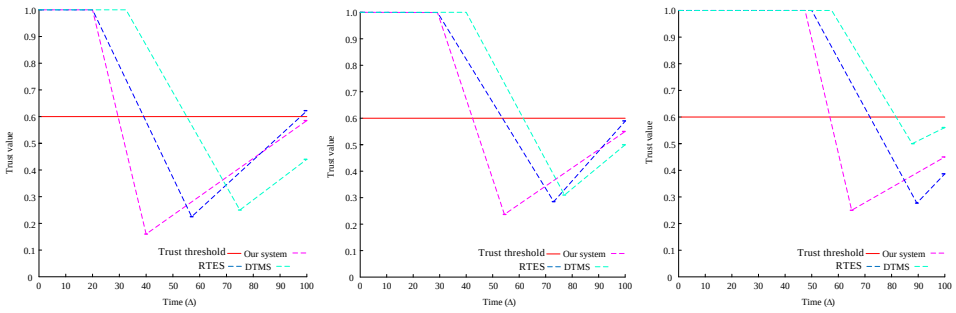


Fig. 6. On-off attack detection

5.4. Comparison of reactions against On-off attack

In order to prove the effectiveness of our system, we evaluate its conduct towards the On-off attacks. Comparisons were done with RTES [16] and DTMS [5].

In the experiment, the different of contexts (service, capability and community interest) are not considered. The percentage of malicious nodes P_m is 10%, 40% and 70%, respectively. The malicious nodes behave well for the 10 first Δ. Then it provides bad services for the second 10Δ and reverts to normal. From **Fig. 7**, one can see DTMS takes more time to detect the bad behavior of the node, and therefore hides the malicious transition for longer. Our system and DTMS detect earlier the node misbehavior. Compared with other two schemes, our system can efficiently recognize this bad behavior and starts to decrease slightly its trust level. Once the node is recognized, it stops bad behaviors and regains trust. The reason is that our system adds a bad behavior factor in trust estimation, so that dishonest nodes require them to perform many good actions to recover their trust values.



(a) P_m =10% (b) P_m =40% (c) P_m =70%

Fig. 7. Trust level evolution of the in presence of on-off attack

6. Conclusion and Future

Traffic classification was carried out in two phases. In the first off-line phase, we started with no assumptions about traffic classes and used the unsupervised SOM and K-means clustering algorithms to find the structure in the traffic data. The data exploration procedure found three clusters corresponding to three QoS classes: transactional, interactive, and bulk data transfer. There are a large number of smart sensor nodes in the Internet of Things (IoT), and these sensor nodes in IoT often are resource-constrained nodes, so they have a greater need to collaborate with one another for providing advanced service and applications. In the context of the IoT, it is difficult to evaluate the node's trustworthiness in the same trust model when a node provides different services. In addition, guaranteeing the availability of the trust management service is another significant challenge because heterogeneous sensor nodes in the IoT are vulnerable to attacks, and distributed in different communities. Until now, huge amount of work about trust mainly focused on defining and evaluating the trust relationships among nodes and proposing the trusted frameworks and algorithms; rather than the development of a robust model for ensuring the availability of trust management service.

With these issues in mind, this paper aims at developing a robust trust management system that evaluates the trust between two nodes and ensures the availability of trust management service in IoT environments. The proposed system is a partly decentralized trust management

framework that is able to measure the trustworthiness of nodes based on feedbacks collected from participants in a specific context and ensure the availability of trust management services. Depending on the system, a requesting node can select the best partners to provide collaborative service. Our system exploits techniques to ensure the availability of trust management service. We have studied the performance of the proposed trust management system in a simulated environment. In future, we will solve the recovering problem of lost data during a down time of power nodes, and predict the availability of each power node. We also will analyze the vulnerability of the system to other threats. Performance optimization of the trust management system is another focus of our future research work.

Acknowledgments

The work in this paper has been supported by National Natural Science Foundation of China (Program No.71501156 and No.61702414) and China Postdoctoral Science Foundation (Program No.2014M560796).

References

[1] W. Abdelghani, C.A. Zayani, and I. Amous,F. Sèdes, "Trust management in social Internet of Things: A Survey, " in *Proc. of 15th IFIP Conference on e-Business, e-Services and e-Society*, pp. 1-12, 2016. Article (CrossRef Link)

[2] N.B. Truong, G.M. Lee, "A reputation and knowledge based trust service platform for trustworthy social Internet of Things," in *Proc. of 19th International Conference on Innovations in Clouds, Internet and Networks*, pp. 104-111, 2016. Article (CrossRef Link)

[3] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The Social Internet of Things (SIoT) - When social networks meet the Internet of Things: Concept, architecture and network characterization," *Computer Networks*, vol. 56, no. 16, pp. 3594-3608, November, 2012. Article (CrossRef Link)

[4] Ray, B.R., Abawajy, J., Chowdhury, M., "Scalable RFID security framework and protocol supporting Internet of Things," *Computer Networks*, vol. 67, no. 10, pp. 89–103, 2014. Article (CrossRef Link)

[5] C. V. L. Mendoza, J.H..Kleinschmidt, "Mitigating on-off attacks in the Internet of Things using a distributed trust management scheme," *International Journal of Distributed Sensor Networks*, vol. 11, no. 11, pp. 1-8, 2015. Article (CrossRef Link)

[6] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for Internet of Things," *Journal of Network and Computer Applications*, vol. 42, pp. 120-134, 2014. Article (CrossRef Link)

[7] Guo, J., Chen, R., "A classification of trust computation models for service-oriented internet of things systems," in *Proc. of 2015 IEEE International Conference on Services Computing (SCC)*, pp. 324-331, 2015. Article (CrossRef Link)

[8] Xiao, H., Sidhu, N., Christianson, B, "Guarantor and reputation based trust model for social internet of things," in *Proc. of International Wireless Communications and Mobile Computing Conference*, pp. 600-605, 2015. Article (CrossRef Link)

[9] M. D. Alshehri, F. K. Hussain, "A comparative analysis of scalable and context-aware trust management approaches for Internet of Things," in *Proc. of 22nd International Conference on Neural Information Processing*, pp. 596-605, 2015. Article (CrossRef Link)

[10] F. Bao and I.-R. Chen, "Dynamic trust management for internet of things applications," in *Proc. of 1st International Workshop on Self-Aware Internet of Things*, pp. 1-6, 2012. Article (CrossRef Link)

[11] F. Bao and I.-R. Chen, "Trust management for the internet of things and its application to service composition," in *Proc. of 13th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp.1-6, 2012.Article (CrossRef Link)

[12] N. Labraoui, M. Gueroui, and L. Sekhri, "On-off attacks mitigation against trust systems in wireless sensor networks," in *Proc. of 5th IFIP International Conference on Computer Science and Its Applications*, vol. 456, pp. 406-415, 2015. Article (CrossRef Link)
[13] Y. Chae, L. C. DiPippo, and Y. L. Sun, "Trust management for defending on-off attacks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 1178-1191, 2015. Article (CrossRef Link)
[14] Y. B. Saied, A. Olivereau, D.Zeghlache and M. Laurent, "Trust management system design for the Internet of Things: A context-aware and multiservice Approach," *Computers & Security*, Volume 39, Part B, pp.351-365, 2013. Article (CrossRef Link)
[15] I.R. Chen, F. Bao, and J.Guo, "Trust-based service management for social Internet of Things Systems," *IEEE Transactions on Dependable and Secure Computing*, vol.13, pp. 684-696, 2016. Article (CrossRef Link)
[16] F. Ishmanov, S. W. Kim and S. Y. Nam, "A robust trust establishment scheme for Wireless Sensor Networks," *Sensors*,vol.15, pp. 7040-7061, 2015. Article (CrossRef Link)
[17] M. Jo, V. Odelu, A.K. Das and K.K.R. Choo, "Expressive CP-ABE scheme for mobile devices in IoT satisfying Constant-size Keys and Ciphertexts," *IEEE Access*, Vol.5, No.1, pp.3273-3283, 2017. Article (CrossRef Link)
[18] S. Roy, S. Chatterjee, A.K. Das, S. Chattopadhyay and S. Kumari, "Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing Internet of Things," *IEEE Internet of Things Journal*, 2017. Article (CrossRef Link)
[19] M. Wazid, A.K. Das, M.K. Khan, A.D. Al-Ghaiheb, N. Kumar, "Secure authentication scheme for medicine anti-counterfeiting system in IoT environment," *IEEE Internet of Things Journal*, Vol. 4, No. 5, pp. 1634-1646, 2017. Article (CrossRef Link)
[20] S. Challa, M. Wazid, A.K. Das, N. Kumar, A.G. Reddy, "Secure signature-based authenticated key establishment scheme for future IoT applications," *IEEE Access*, Vol. 5, pp. 3028-3043, 2017. Article (CrossRef Link)
[21] S. Kumari, M. Karuppiah, A.K. Das, X. Li, F. Wu, "A secure authentication scheme based on elliptic curve cryptography for IoT andcloud servers," *The Journal of Supercomputing*, 2017. Article (CrossRef Link)
[22] ns3. Available from: Article (CrossRef Link)

Xu Wu received her Ph.D. degree in Computer Science, from the Beijing University of Technology, in 2010. She was out of post-doctoral stations of the MOEKLINNS Lab, Department of Computer Science and Technology of Xian Jiaotong University in 2016. She is an associate professor of Guangxi University. She is working as a visiting scholar in School of Engineering and Technology, Indiana University–Purdue University Indianapolis, Indianapolis, USA, when working on this paper. Her research interests include trusted computing, pervasive computing, mobile computing, and software engineering. She has published more than 30 technical papers and books/chapters in the above areas. Her research is supported supported by National Natural Science Foundation of China (Program No. 71501156 and No. 61702414) and China Postdoctoral Science Foundation (Program No.2014M560796) and Shaanxi Postdoctoral Science Foundation and special funding for key discipline construction of general institutions of higher learning from Shanxi province.