A Novel Trust Management Model in the Social Internet of Things

Gholamhossein Ekbatanifard[1], Omid Yousefi[2]
1- Computer Engineering Department, Lahijan Branch, Islamic Azad University, Lahijan, Iran.(ekbatanifard@liau.ac.ir)
2- Mehrastan University.

Received (2018-11-17) Accepted (2019-04-01)

Abstract: *The Internet of Things (IoT) and social networking integration, create a new concept named Social Internet of Things (SIoT) according to which the things are able to autonomously establish social relationships with regard to the owners. Things in SIoT operate according to a service-oriented architecture. There may be misbehaving owners and consequently misbehaving devices that can perform harmful attacks based on their social relationships with other things for their own gain at the expense of other IoT devices. This motivates us to work on the issue of how to estimate the trust of a service provider to avoid malicious service providers and select the best service provider. In this paper, a novel trust management model is proposed based on four properties. The model deals with attacks (especially on-off attacks) and considers service levels for services provided by each node. A method to provide different levels of services via SIoT devices, and a new trust assessment scheme are the contributions of this paper. We evaluated the proposed scheme with extensive simulations and the results show that the proposed model can effectively select the best service provider and cope with most trust related attacks.*

Keywords: *Trust, Trust management, Social Internet of Things, Security.*

How to cite this article:
Gholamhossein Ekbatanifard, Omid Yousefi . A Novel Trust Management Model in the Social Internet of Things J. ADV COMP ENG TECHNOL, 5(2) Spring : 57-70

I. INTRODUCTION

The term Internet of Things (IoT) was coined by Kevin Ashton in 1999. At that time the internet was the hottest trend and technology got a new foundation. Now we are living in the era of Internet-of-Things, where billions of computing devices surround us, operating and interacting within to provide some of the most significant computing services[1].

A Social Internet of Things (SIoT) system can be viewed as a mix of P2P and social networks, where things autonomously establish social relationships according to the owners' social networks and can provide services to each other. The internet of social things combines cyberspace and physical word through tags, RFIDs, sensors and things owned by people. Things in a real word are traceable by connecting to the tags and information can be obtained from environmental conditions via these tags. Smart objects such as mobile phones and other electronic devices, which have computing and resource sharing capability, are able to provide millions of services to connect things in many places. Emerging SIoT phenomenon attracted many applications towards it, such as electronic health (e-health) [2, 3], smart homes and communities [4, 5].

Such future of SIoT applications will act based on a service-oriented architecture (SOA)

[6], where every device is a service consumer and could be a service provider when it is necessary to provide a service or resources to share and interact with other consumers of services through APIs. Our motivation for providing a trust management system for SIoT systems is noticeable: There are misbehaving owners and consequently malicious devices that may perform discriminatory attacks based on their social relationships to ruin the reputations of other IoT devices, which provide similar services, in order to monopoly the list of specific services.

SIoT brings similar capabilities of humans to things, where things act like social behavior of humans [7]. The types of relationships have been taken from some sociologists and anthropologists studies, such as [8] and [9]. [10] analyzes the implementation of such model of behavior in IoT. Some of the other parameters are also studied in [11]. The owners of SIoT can control their devices and the relationships between them. The owners can let things to provide services upon requests of other things. Obtaining permission from owners may only be done in the first time and then be used in other interactions. The owners may also show selfish and malicious behavior against a service requester.

In this paper, we aim to design and evaluate a novel trust management model for SIoT based on SOA. Our proposed trust management model runs autonomously by SOA based IoT devices so that it requires minimal human supervision. The basic idea of the proposed trust management model is inspired by [12] Our contributions in this paper are: 1- proposing a new method to provide levels of services via the SIoT devices 2- proposing a new method for trust assessment based on four properties that can deal with existing trust related attacks including on-off attacks.

The rest of the paper is organized as follows: in Section II we review the related works, we present the Concepts in Section III, and in Section IV we present the proposed model and the evaluation of the model is presented in Section V. Finally, we summarize the paper and outline the future works in Section VI.

II. RELATED WORK

There are various related works about trust management protocol in p2p service provider systems [13-16]. These p2p systems has a common aspect with IoT systems so that nodes, themselves provide services, hence, node's trust evaluation is crucial, but trust protocols for P2P service provider systems do not consider social aspect of SIoT devices. Hence, they are not implementable in a SIoT systems that includes heterogeneous objects with different owners, relationships, and interests. On the other hand, trust management protocols in social networks [11, 17, 18] evaluate entities due to times, duration and modality of the relationships between two entities. These are inattentive to P2P service providing which SIoT devices themselves are responsible for service providing. There is little work on Trust management in SIoT systems especially for confronting malicious attacks [19-21]. A few works in SIoT trust management is conducted so far [12, 22-25].

[25] Proposed a trust management model based on fuzzy reputation for IoT systems. Their trust management model considers a very specific IoT environment populated with wireless sensors only, so they only considered the quality of service trust metrics like packet forwarding, delivery ratio, and energy consumption for measuring trust of sensor devices. On the contrary, our work considers both QoS and social properties, which give rise to social relationships of owners of IoT devices in the social IoT environment. [23] proposed a context-aware and multiservice approach for trust management in IoT systems against malicious attacks. However, it requires the presence of centralized trusted servers to collect and disseminate trust data, which is not viable in IoT environments. Relative to this work, our trust protocol is completely distributed without requiring any centralized entity.

[24] Proposed a trust management protocol considering both social trust and QoS metrics, and using both direct observations and indirect recommendations to update trust in IoT systems. However, the issue of adaptively adjusting trust evaluation in response to dynamically changing conditions, to cope with misbehaving nodes and maximize the performance of IoT applications

running on top of the trust management, was not addressed.

[22] Considered the social relationships of owners of IoT devices for trust management in social IoT systems. They proposed two models for trustworthiness management. Namely, a subjective model deriving from social networks, with each node computing the trustworthiness of its friends on the basis of its own experience and on the opinion of friendly recommenders, and an objective model deriving from P2P communication networks with each node storing and retrieving trust information towards its peers in a distributed hash table structure, so that any node can make use of the same information. Their objective model requires pre-trusted nodes to be in place for maintaining the hash table, which is questionable in IoT environments. Their subjective model taking into consideration the social relationships between owners of IoT devices.

[12] Considered that each IoT device evaluates other IoT devices using both direct service experiences and indirect recommendations. Adaptive IoT trust, a distributed IoT trust management protocol, is the end product. Adaptive trust management is achieved by determining to combine direct trust (from direct experiences) and indirect trust (from recommendations) dynamically. It seems that they did not consider some trust composition properties to evaluate a nodes trust which can lead to more accurate trust evaluation for service providers. For example, all transactions have only one importance level.

All of the above-mentioned works did not present any solutions to deal with the on-off attack and some other like [23-25] didn't work on opportunistic service attacks. By comparing their models, it seems that [12] and [22] are better models. In [12], some trust composition properties, which can lead to more accurate trust evaluation for service providers, are not involved. For instance, transaction's importance property is not considered and all transactions have just one importance level. As an example, for transaction's importance property, a thing might achieve a high trust value because of providing an environment weather temperature service and take this trust value for a more important transaction like a financial transaction, which in fact may cause problems. A malicious node raises its trust value by providing a good service and then takes part in a subversive transaction. To complete it, required social properties and quality of services can be added to reach more accurate trust value in different environmental conditions. In addition, both proposed models in [12] and [22] are vulnerable to dealing with on-off attacks. In contrast, our proposed model has improved previous models in many parts including presenting a new trust evaluation approach based on four properties and also considered levels for services provided by nodes. Other contributed innovations are to prevent opportunistic service and on-off attacks by the proposed trust predictability model.

III. THE CONCEPTS

The threat in our management model is presented in this section. In an IoT system, each IoT device can be a service provider (SP) or a service requester (SR) or both. Every IoT device wants to be chosen to provide service for profit when it is an SP and wants to find the best SPs for best available service when it is an SR.

A malicious SP acts for its own benefit and would like to be selected to serve a service even if the service providing is inferior. In the context of IoT, we are concerned with trust-related attacks that can disrupt the trust system. Bad-mouthing and ballot-stuffing attacks are the most common forms of reputation attacks. Self-promoting and opportunistic service attacks are the most common forms of attacks based on self-interest [12]. On-off attacks are often used by malicious nodes to evade detection. The service feedback value provided by a malicious node is low and the intimacy value of such node is low too and has little cooperation among nodes. Five types of trust related attacks that a malicious node can perform are introduced as follows:

Self-promotion attacks (SPA): A malicious node can promote its importance (by providing good recommendations for itself) so as to be selected as an SP, but then can provide bad or malfunctioned services. We address this attack by Service feedback property (see section IV.3).

Bad-mouthing attacks (BMA): A malicious node can ruin the trust of a well-behaved node by providing bad recommendations against it, in order to decrease the chance of that node being selected for service. This is a form of collusion recommendation attack, i.e., a malicious node can collaborate with other malicious nodes to ruin the trust of a good node. We address this attack by evaluating of nodes' trust toward a recommender. The recommender's trust will be increased only by providing good services (see section IV.3).

Ballot-stuffing attacks (BSA): A malicious node can boost the trust of a malicious node by providing good recommendations, so as to increase the chance of that malicious node being selected as an SP. As mentioned before, in BMA, we address BSA attack by evaluating of nodes' trust toward the recommender (see section IV.3).

Opportunistic service attacks (OSA): A malicious node can provide good service to gain high reputation opportunistically, especially when its reputation is dropping because of providing bad services. With a good reputation, it can effectively collude with other bad nodes to perform bad-mouthing and ballot-stuffing attacks. We address this attack by a trust predictability method to predict attacks (see section IV.3).

On-off attacks (OOA): Instead of always performing good services, a malicious node can sometimes perform bad services. With on-off attacks, a malicious node performs a bad service on and off (randomly) so as to avoid being labeled as a low trust node and risk itself not being selected as an SP, as well as not being able to effectively perform bad-mouthing and ballot-stuffing attacks. We address this attack by a trust predictability method to predict attacks and cope with the OOA (see section IV.3).

IV. THE PROPOSED MODEL

SIoT brings similar capabilities of humans to things, where things act like humans social behavior [7]. We introduce a dynamic trust management system for trust value assessment of service providers in which each object can provide a level of services to other nodes. In the next section, we present the system model in our management model.

1. System model

The main purpose of this paper is presenting a new trust management model. As Fig. 1 shows, in our model, the set of SIoT nodes are shown by a set of vertices {1, 2, …, n}. Each node's communication with other node are shown with an edge between two vertices. Each node in the figure can provide one or more services. Each node can be in the role of a service requester and service provider. Each service has some service levels that are referred as S_i^L where i is service number and L is the level of service. For example, node 5's services and its service levels are shown as $\{S_1^1, S_1^2, S_2^1, S_2^2, S_3^1\}$.

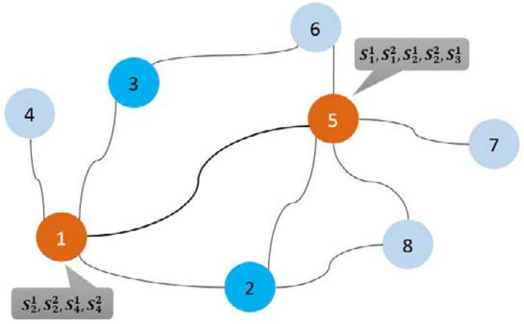

6
$S_1^1, S_1^2, S_2^1, S_2^2, S_3^1$
3
4
5
7
1
8
2
$S_2^1, S_2^2, S_4^1, S_4^2$

Fig. 1: displaying network nodes with their service levels in our model.

Consider a scenario in Fig.1 where node 1 provides services S_2 and S_4 and requests service S_1. Among all nodes which node 1 is communicating with, node 5 provides it. So, node 5 is a service provider. Now assume that node 5 is not the only node that provides the required service, and nodes 6 and 7 are able too, hence node 1 needs to assess the trust of these nodes to select the best service provider. In each step, the service requester node i evaluates trust value of its intended services, provided by node j, denoted by T_{ij}. This trust value is evaluated by a combination of direct and indirect trust evaluation properties. Likewise, in each step according to service receivers' trust history, if the node is identified as a good node, P^+ score is assigned to node *i* and if it is known as bad, P^- score will be assigned to its latest score. The total score of service requester *i* is kept by node *j* as P_{ij}. This score for the newly joined node is considered zero.

Similar to the most related research papers [12, 22], the trust value is scaled in the range of 0 to 1 in this paper, where a trust value closer to 0 means a low degree of trust and a higher degree of trust has a value closer to 1.

2. Trust Evaluation

SIoT consists of thousands of heterogeneous social objects that usually provide various services. Therefore, evaluating things' trust is essential in order to choose a reliable service provider. SIoT applications may utilize various evaluation methods. For instance, in an application, the number of packets that are received accurately might be considered as an evaluation parameter. In another application in a greater network including social relations between nodes, it is required to evaluate not only nodes' sociability but also social groups which these nodes belong to. As a result, according to the applications' requirement, trust evaluation properties might be diverse and different.

In our approach, some distinct properties of trust are used to calculate the total trust. It means that according to applications' requirements, one or more properties might be taken into account and their practical use is denoted in part 5.6. To assess the trust value, different properties are required to be evaluated. Although there are numerous social trust properties [11, 26], in our approach 4 more important and effective properties such as intimacy, service feedback, sociability and transaction importance are used. Intimacy determines how intimate two nodes are. It is calculated by observing the interactions between every two nodes. Service feedback presents a response or evaluation of a received service by a node. Sociability is a combination of social relations and the degree of friendship. The last but not the least property is transaction importance, which is for considering different levels of importance for each service transactions so that nodes' trust value increases more as long as they carrying out more important transactions. All these properties are calculated individually, but complete each other.

Our approach here is transaction-based; every time that a node requests a service, trust is evaluated. If the required trust is satisfied, service will be received and the receiver node will rate it accordingly.

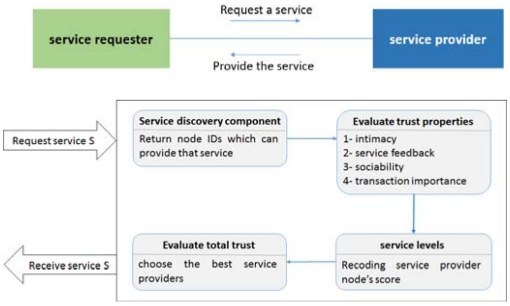

Request a service
service requester
service provider
Provide the service
Request service S
Service discovery component
Return node IDs which can provide that service
Evaluate trust properties
1- intimacy
2- service feedback
3- sociability
4- transaction importance
Receive service S
Evaluate total trust
choose the best service providers
service levels
Recoding service provider node's score

Fig. 2: Service provider and Service requester interaction.

As Fig. 2 shows, the main steps of evaluating trust in our approach are as follows:

Step 1: *Requesting service:* when a node in the network requests a service, a list of all service providers will be given to the node by service discovery component.

Step 2: *Evaluating service providers' trust properties:* Fig. 2 shows a service provider and a service receiver interaction for two nodes. The requester node evaluates trust properties of the provider node. Evaluation involves calculating current received services' trust (first-hand information) and previously stored information. In addition, other node's recommendations about the service provider node may be used.

Step 3: *Evaluating total trust:* total trust will be computed according to the application requirements and 4 properties which are calculated in the previous step. The calculated information about the service providers' trust will be stored to be used in the future.

Step 4: *Recording service providers' score:* for a newly joined node there is no information available at the beginning, so its score is considered zero. Hence, the minimum service level is assigned to it. If a node isn't a newcomer, the node score is recorded by the service provider node. Using this information, the node receives a proper level of service. It's a crucial point that a node with a higher score not only requests services one-way but also provides services properly as well.

These aforementioned steps are explained below in more detail:

2.1 Requesting a service

By joining a node to a network, if it requests a service, its request will be received by a service discovery component. A list of service providers that are able to provide the intended service is presented by the service discovery component. The list includes node *ids* which can provide that service. In addition, nodes which is in the network may ask for the list from this component as well. The returned list includes the *id* of nodes that are connected directly or indirectly to the current node.

2.2 Service providers' trust properties

As mentioned earlier, the approach is based on interactions. In each interaction, all trust properties must be studied individually. These properties are intimacy, service feedback, and sociability and transaction importance. Even though, there are numerous trust properties evaluation methods and properties that can be used, in our approach these four properties are taken into account and they are selected in order to satisfy the majority of SIoT applications. Assume that symbol X can be each of these four properties. To evaluate them we need the old trust value of X property and also a new evaluated one (direct or indirect trust value of X property). Equation 1 is introduced to evaluate trust value of node *i* to node *j* for trust property of X:

$$T_{ij}^{X} = (1-\alpha-\omega)\,TO_{ij}^{X} + \alpha\,TD_{ij}^{X} + \omega\,TR_{kj}^{X} \tag{1}$$

In equation 1 symbol T_{ij}^{X} indicates evaluated trust value of X property of node *i* about the service provided by node *j*. This is formed by three sections that are: TO, TD and TR which are defined as follows:

$\boldsymbol{TO_{ij}^{X}}$: Old trust of X property that is stored by node *i* toward node *j*.

$\boldsymbol{TD_{ij}^{X}}$: Direct trust of X property that is calculated by node *i* toward node *j* at present time.

$\boldsymbol{TR_{kj}^{X}}$: Indirect recommended trust of X property that is currently evaluated by node *k* (the middle node between *i* and *j*) toward node *j*.

Furthermore, parameters ω and α are used to weigh the importance of each part of the equation. In other words, α is used to weigh of the node *i*'s direct trust toward node *j*, ω is used to weigh node k's recommended trust value toward node *j* and the remaining weight (1 - α - ω) is used for old trust. In trust evaluation of X, while the requesting node is connected directly to the provider node, calculating indirect trust *TR* is not necessary and ω is considered zero in order to isolate. Likewise, while two nodes are not connected directly α is considered zero in order to isolate. α and ω are adjusted according to the application's requirement. Increasing α and ω leads to assigning a heavier weight to new evaluated trust and a light weight to old trust. The principle of evaluating trust properties are explained in details below:

Intimacy: Trust value of node *i*'s intimacy toward node *j*, refers to total interactions between *i* and *j* over total interactions between *i* and its neighbors.

$$TD_{ij}^{intimacy} = \frac{tr_{ij}}{tr_i} \tag{2}$$

In equation 2, $TD_{ij}^{intimacy}$ illustrates the evaluation of node *i* 's direct intimacy trust toward node *j*, tr_{ij} is the total transaction between *i* and *j*, and tr_i is the total number of transactions done by *i* and its neighbors. Intimacy value is a number ranging from 0 to 1, where 0 indicates low trust intimacy and 1 is high trust intimacy.

If the connection between nodes *i* and *j* is indirect, the evaluation of node *k's* trust value, which is a middle node between *i* and *J*, will be calculated by equation 3:

$$TR_{kj}^{intimacy} = \frac{tr_{kj}}{tr_k} \tag{3}$$

Service feedback: Service feedback, which is computed by node *i* about the service received by node *j*, is evaluated based on direct observations from node *i* toward node *j*. Node *i* calculates

$TD_{ij}^{feedback}$ by assessment of the received service by a value between 0 and 1. This value can be assessed due to direct evaluation of node *i* or by counting node *j's* suspected experiments using anomaly detection techniques such as a huge difference in reading sensors' value or interrupt, repetition, delay [27, 28]. If the number of suspected experiments is more than a specified threshold, then node *j* will be considered as a node with low service feedback $TD_{ij}^{feedback}$. If node *j* is not a direct node, a recommender node will be needed. While considering this node as *k*, its service feedback about *j* is shown by $TD_{kj}^{feedback}$.

Sociability: Sociability property of nodes is the combination of two properties, the degree of friendship (DF) of two nodes, and social groups (SG) that two nodes are the member of. Friendship degree is the number of mutual friends of i and j over the total number of their friends. If this ratio is greater, the relationship between these two nodes is stronger. Friendship degree is a number between 0 and 1. Equation 4 is used to calculate it.

Nodes' social groups play an essential role in evaluating sociability. The more mutual social groups of two nodes over their total number of social groups, the more mutual social interests and the closer relationship they have. Social groups here mean: being in the same location, being colleagues, being in the same city, having the same brands etc. equation 5 is used to evaluate this value.

$$DF_{ij} = \frac{friend(i) \cap friend(j)}{friend(i) \cup friend(j)} \qquad (4)$$

$$SG_{ij} = \frac{Group(i) \cap Group(j)}{Group(i) \cup Group(j)} \qquad (5)$$

In order to preserve privacy, node *i* and node *j* agree on a one-way hash function upon interacting while exchanging the list of groups or friends. Thus, the lists of groups or friends lists exchanged are encrypted with a one-way hash function, as a result, each node is only able to identify a list of common groups or friends, but cannot know the other party's list. Equation 4 and 5 show the evaluation of friendship degree and mutual social groups of two direct nodes. Here to evaluate sociability property of two nodes equation 6 is used:

$$TD_{ij}^{Sociability} = \lambda DF_{ij} + \lambda SG_{ij} \qquad (6)$$

In equation 6, λ is considered 0.5 to *DF* and *SG* values contribution is equaled. If there is no direct connection between *i* and *j*, indirect sociability trust of node *k* as a recommender about node *j* is computed by equation 7.

$$TR_{kj}^{Sociability} = \lambda DF_{kj} + \lambda SG_{kj} \qquad (7)$$

Transaction importance: Points that are awarded to a transaction or service, for instance, a temperature query may be less important than financial transactions. The importance of each transaction can be between 0 and 1. The more important they have, the closer to 1 their point is. The less important a transaction have, the point is closer to 0. The importance of each transaction considered constantly by service requester node. The importance of direct transaction for a node *i* which is connected directly to a second node *j* is shown by $TD_{ij}^{importance}$ importance and in the case of indirect communication, it is shown by $TR_{kj}^{importance}$ importance.

2.3 Total trust

T_{ij}^{X} denotes the total trust value of each of the 4 properties intimacy, service feedback, sociability, and transaction importance. How to evaluate the total trust of these four properties is one of this part's challenges that depends on the social application's requirements that using our trust management approach. Our goal is to adopt the necessary parameters and properties dynamically to reach maximum performance for applications in different situations. Evaluating total trust is expressed by a mathematical expression that will be explained in section V.6, in the form of an example of a SIoT application.

2.4 Service provider node's score

In addition to provide a number of services, things are able to supply some levels of service to a particular node. For example, suppose a node that wants to communicate with a printer, where the printer is a service provider and provides print service. It could be the first time that the printer node communicates with an unfamiliar node, so less printing service level will be provided for that node. For example, it only allows printing 5 sheets of paper in a day. If this node does not have any hostility behavior, the print service points increased and a new level of service released. For example, it allows printing 100 sheets of paper per day.

As mentioned before, this approach is based on transactions. In each transaction, parties involved in it do its evaluations. Receiver node calculates trust value in order to choose one of the service providers among all providers of that specific service then this information will be stored in the node. The service provider rates this node by a score based on requesting node's trust history and type of service. This score increases by P^+ and decreases by P^- constantly for each service. In the beginning, this score starts with 0 and then increases and decreases by P^+ and P^-. Deciding whether to decrease or increase this value is due to the node's old trust value. If the score of service requester increases based on providing good services, in fact, if the service receiver is a good service provider as well, then it will receive a higher service level. Each node stores a table like table 1. Each node at the end of its services receives a score based upon its old trust value. This value is saved in the node's memory as well as total trust.

In table 1, three types of services are provided. The first and second services have two levels 1 and 2 and the third service has only one level. A negative score is considered more than positive because, by this, the bad node is punished more and receives less point. Scoring also can be based on service level or just by the type of the service that is requested.

Table 1
scoring services in a sample node.

service	positive point (p^+)	Negative point (p^-)
S_1^1	0.0010	0.0030
S_1^2	0.0015	0.0020
S_2^1	0.0020	0.0060
S_2^2	0.0030	0.0090
S_3^1	0.0500	0.0600

3. Defending against trust related attacks

In self-promotion attacks, a malicious node can promote its importance by providing good recommendations to other nodes. In the proposed approach despite self-advertising, a node must provide good services not only to gain higher scores by other nodes but also to reach higher trust values as well. Service feedback property is introduced in order to evaluate a nodes' trust based upon its provided service.

To defeat bad-mouthing and ballot-stuffing attacks, nodes do not accept recommender node's unless there is no direct connection between them. In this method, parameter β is utilized to evaluate trust and is computed by equation 8.

$$\omega=\beta\ .TD_{ik} \qquad (8)$$

Computing parameter β by equation 8 leads to take node *i's* trust toward node k into action and defend bad-mouthing and ballot-stuffing attacks. Parameter ω is to weight direct trust between *i* and *k*. This is an essential point that total and recommender node's trust will be increased just by providing good services.

In defending against opportunistic service attacks, a malicious node can provide good service to gain high reputation opportunistically, especially when it senses its reputation is dropping because of providing bad service. To address this type of attack and also on-off attacks *trust predictability* method is proposed as follows.

By adding *trust predictability* as a new component to the proposed trust model, nodes can be controlled and be able to find out if a node will be bad or good in the future. Due to the calculated total trust in step four and a chosen threshold, nodes that their overall trust is less than the threshold will be marked as bad nodes. In each step, after computing trust value, it is determined whether a node is bad or good and the trust value will be stored in the node. To save these values, a window is utilized to keep a specific number of last transactions. A window with a certain size is defined. In each transaction, by comparing node's trust value with the trust threshold it is determined if a node is bad or good. Then the last value which might be good (G) or bad (B) is stored in a binary form 0 or 1. Fig. 3 shows a window which has stored the behavior of a node.

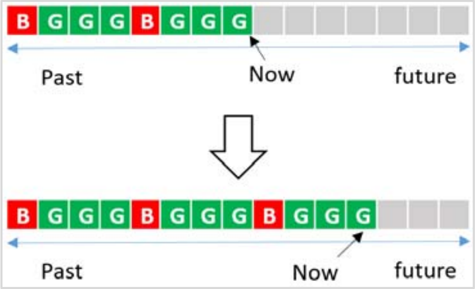


Fig. 3: The window to store nodes' behavior.

Equation 9 is used to calculate the amount of trust predictability. If PT_i is low, two assumptions are considered; one, this node's trust might be low and two, this node behaves unstable and do on-off attacks or opportunistic service attacks.

$$PT_i = \frac{G}{G + B} \qquad (9)$$

By means of trust predictability, it is possible to predict attacks. If PT_i is less than a specific threshold, the requesting node will insert the malicious nodes in a blacklist for a while and will not interact with them. This period of time is defined to allow a malicious node to amend. After this period, the window will establish again to assess the node.

V. EVALUATIONS

To evaluate the proposed approach, we run extensive simulations, where the simulator includes more than 4000 lines of code. Table 2 shows the simulation parameters. In the simulations, primary nodes' trust is supposed to be zero, and all transactions are event-based. The total number of groups and services are increased by incrementing the number of nodes. In the beginning, the simulation environment is considered as an environment with no hostility and conflicts and the simulator is capable of changing the simulation environment dynamically. Each node has a list of its neighbors and it is assigned at the beginning of the simulation. The list doesn't change during the simulation. Even though the model supports node mobility, but nodes are considered to be fixed during our simulations. Each object can be a member of six existing groups. The total number of groups is proportional to the number of nodes. The number of transactions is shown in the following figures. Simulations observations indicate that the best output is achieved by changing α and β dynamically.

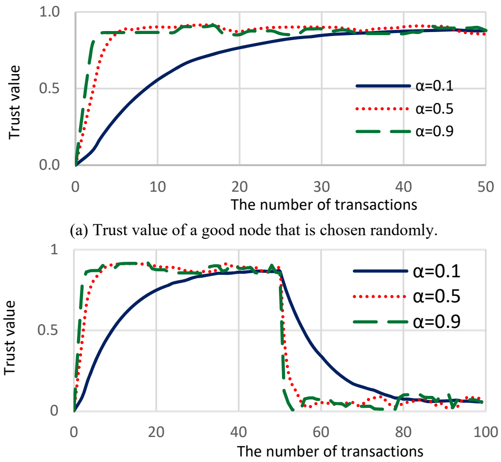
Table 2
simulation parameters.

Title	value
Number of things	50
Min object distance	100m
Min connections distance	260m
Network dimension	1000m × 1000m
Groups of each node	6
Services of each node	3
Objects, services, groups distribution strategy	randomly
α, β and threshold	Dynamic due to condition

1. The effect of α on trust evaluation

We first investigate the impact of the α parameter on trust evaluation. In equation 1, α is the weight of the direct trust to the old trust. In this section, service feedback property is used to evaluate the model while for the other three properties the same applies. To study the sensitivity of α, its value varies from 0.1 to 0.9 and we set β to 0 to isolate its effect. Fig. 4(a) shows the impact of service feedback on good nodes which do not perform malicious attacks. Two nodes among all are chosen randomly and their trust values are assessed within 50 transactions. It is notable that the first trust value is considered 0. It is observed that by selecting smaller α trust evaluation converges to 1 slowly. The more this value is, the faster trust value converges to 1.

In Fig. 4(b), trust value of a good node that changes into a bad node after a while is evaluated. In the beginning, a healthy interaction environment is determined and the node's behavior changes gradually into hostility. Hostility among things in the second 50 transactions is set at 100%. As it is obvious in the figure, in the first 50 transactions we are dealing with a good node and node's trust value is converged to 1. Trust convergence for α=0.1 is slower than α=0.9. After the 50 transactions, the environment turns into a hostile mode and as it can be seen, the trust value in all three graphs converges rapidly to 0. Convergence speed for α=0.1 is slower than α=0.9.



(a) Trust value of a good node that is chosen randomly.

(b) Trust value of a bad node that is chosen randomly.

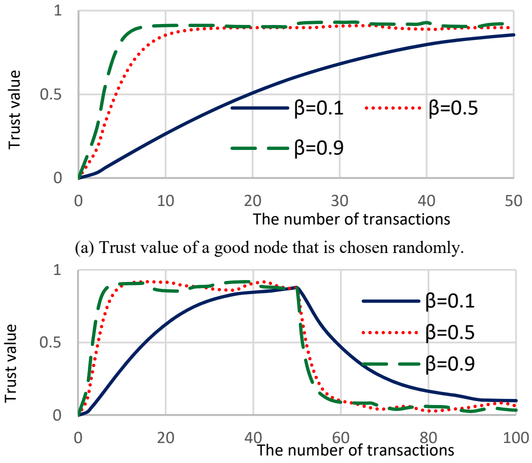
Fig. 4: evaluating α in service feedback trust: (a) A good node that changes into a bad node after a while, it converges after the 23rd transaction. (b) It changes into a malicious node after the 50th transaction, attack ratio is set to 100% and trust value converges to 0.

2. The effect of β on trust evaluation

Next, we investigate the effect of β on trust evaluation. β is the weight given to the recommendations of middle node for old trust. In order to analyze β's sensitivity, various values are set as β and then results are assessed. To isolate α is considered zero.

Fig. 5(a), illustrates the evaluation of β for two randomly chosen good nodes in 50 transactions. The more β is, the faster trust convergence occurs. Comparing Fig. 4(a) and 5(a) shows that convergence occurs faster by changing α. For instance, in Fig. 4(a), α=0.1 convergence occurs in the 23rd transaction, however, in Fig. 5(a) for β=0.1 convergence happens in the 40th transaction. The reason is that in direct states, the first node calculates the second node's trust value directly. But in Fig. 5(a), middle nodes' trust value is necessary to evaluate trust so the computation and convergence depend on two nodes and both must be good to attain higher trust value.

Fig. 5(b), illustrates the impact of parameter β on trust evaluation for two randomly chosen nodes. The second node is a good node that changes into a bad node after a while. In the beginning, a healthy interaction environment is determined and the node's behavior changes gradually into hostility. Hostility among things in the second 50 transactions is set at 100%. As it is obvious in the figure, in the first 50 transactions we are dealing with a good node and node's trust value is converged to 1. In the second 50 transactions, the environment turns into a hostile mode and nodes are bad, as it can be seen trust value converges rapidly to 0. With the larger value of the parameter β, faster convergence occurs. In the first 50 transactions by selecting a larger value for β, trust converges faster to 1. After the 50 transactions, the environment turns into a hostile mode and trust value in all three graphs converges rapidly to 0.



(a) Trust value of a good node that is chosen randomly.

(b) Trust value of a bad node that is chosen randomly.

Fig. 5: evaluating β in service feedback trust: (a) a good node that changes into a bad node after a while, it converges since the 40th transaction. (b) After the 50th transaction, it changes into a malicious node and hostility among things is set to 100% and trust value converges to 0.

3. Trust evaluation in changing hostility conditions

In section V.1 and V.2, by changing nodes condition to 100% hostility, trust converges to lower values. Trust values of nodes in different hostility conditions are discussed here considering SPA, BMA and BSA attacks. In the simulation, the hostility of two nodes is adjustable in percentage. To evaluate this part, hostility is tested in 3 states

10%, 40% and 70% for 100 transactions. Trust evaluation parameters α and β are considered 0.3 and 0.5, respectively. 0% Hostility is also taken into account to compare. In Fig. 6, trust evaluation in any of the mentioned states is shown for service feedback property. Other three properties are similar to service feedback. As it is illustrated in Fig. 6 increasing hostility among nodes leads to a reduction in trust value and an increment in trust volatility. The reason is that trust is varying continuously according to α and β. More hostility among nodes causes higher trust volatility. Due to bad and malicious services by some nodes, it is observed that trust is converging to a constant value.

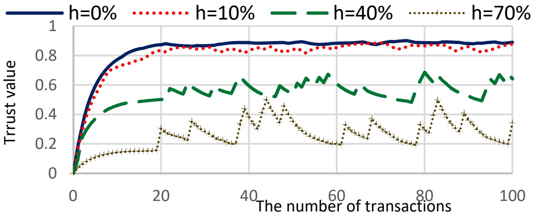
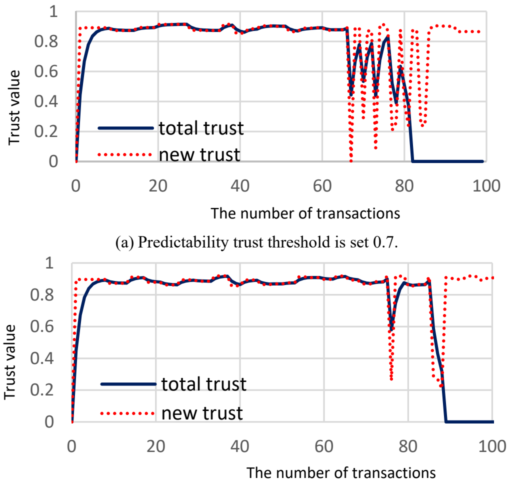


Fig. 6: the effect of the hostile environment in evaluating service feedback trust among two nodes: nodes are chosen randomly and hostility (h) is considered dynamic.

4. Evaluating trust model toward OSA and OOA attacks

The model performance regarding OSA and OOA attacks is studied in this section. Evaluations are under following circumstances: α=0.5, β=0.5 and the second node's hostility at first is zero and after a while it is adjusted between 0 and 100 to simulate the mentioned attacks. Size of the window which stores history of goodness or badness of a node is set 15. Trust threshold for node's badness is considered 0.5.

In Fig. 7(a), two graphs are shown. The first one is the new trust graph which for each received service, service feedback trust are calculated and it has shown just for evaluation. The second graph, the total trust includes the new and the old trust that are influenced by applying α and β on them. It is observed that up to 65th transaction, the second node performs properly and provides sufficient service with average service feedback trust of 0.9. From 65th transaction, the second node that has a high trust value by the first node, a decrease in service feedback trust, in other words an attack with low service feedback trust has occurred. Here trust predictability (PT) threshold is set 0.7 for inserting the node into a blacklist. It is seen that from 66th transaction to 80th, a chance is given to bad node to amend its behavior several times but the node continues its malicious behavior. The trust evaluation value are decreased and PT is less than the specified threshold (0.7) so this node will be inserted into the blacklist and its trust value is set to zero. Since then there is no interaction between the first and second node.



(a) Predictability trust threshold is set 0.7.

(b) Predictability Trust threshold is set 0.8.

Fig. 7: dealing with OSA and OOA attacks in service feedback trust between two nodes that are chosen randomly. The second node does OSA and OOA attacks. (a) Predictability threshold is set 0.7. (b) Predictability threshold is set 0.8.

A stricter strategy can be done by taking a higher threshold for PT. For example, in Fig. 7(b) threshold is changed to 0.8. As it is shown, in this case, the second node has been inserted into the blacklist sooner and less chance is given to the node.

5. Rating services and service levels

As discussed before, for each node three services are considered in the simulation. To assess the functionality of this part, 3 service levels are assigned to each service. The required threshold for releasing service levels by the second node is shown in table 3. Negative points, *p*-, and positive points, p^+, for each service of provider nodes are considered as table 4. Numbers in simulations are randomly generated.

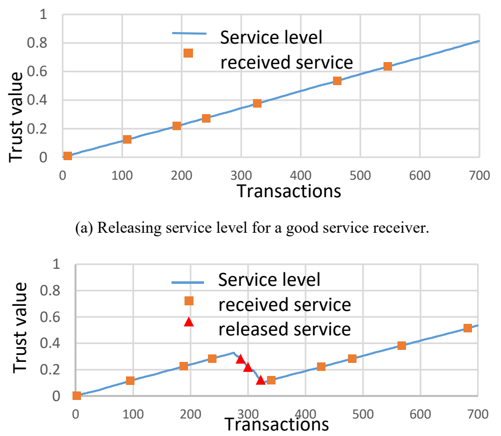
Table 3
score threshold for releasing service levels.

Service	Level 1	Level 2	Level 3
S_1	0.00	0.12	0.38
S_2	0.00	0.28	0.51
S_3	0.00	0.22	0.63

Table 4
scoring services in a service provider node.

Service	positive point (p^+)	Negative point (p^-)
S_1	0.0010	0.0025
S_2	0.0011	0.0030
S_3	0.0015	0.0060

The simulation is run for 700 transactions. Releasing service levels for a service receiver due to its received point is presented in Fig. 8(a). At first, points are zero by default. By this point, three services 1, 2 and 3 are provided with the least service levels. By interacting, the provider node rates the receiver node higher gradually and next service levels are released for it. S_3^3 has the highest threshold and service receiver reached this service level by the 530th transaction. A receiver node must be a good service provider as well to be able to receive good services because its score will be stored for rating. In this figure, the receiver node is assumed as a good node that never provides low-level services or attacks.



(a) Releasing service level for a good service receiver.

(b) Releasing service level for a good node that suddenly provides bad service.

Fig. 8: releasing services: (a) service receiver is good and all service levels are free. (b) service receiver behaves badly for some transactions and loses and achieves some service levels.

By switching on-off attack detection and trust predictability off, it is assumed that a service provider had provided good services for a while and after that provides bad services. Fig. 8(b) illustrates the release of service levels for a service provider which provides bad service for a while. From transaction 275th to 366th, bad services are provided; then again good services are supplied. In this study, 700 transactions have been evaluated. As it is indicated in the figure, till 275th transaction, six service levels are released and after that bad services are provided and three service levels that are obtained earlier are withdrawn. But after the 366th transaction, good services are supplied again and services are released. It is essential that trust increment diagram slope is less than trust decrement diagram slope and it is because of the assigned factor to the positive points (p^+) toward the negative points (p^-). In other words, each node achieves points hardly and loses it easily. Another point is that, till 700th transaction, the node is not able to release a service level S_3^3 .

6. A sample of a SIoT application

There are lots of applications about SIoT [10, 29, 30]. In this section, we present a real application and analyze how to evaluate the total trust value.

Imagine a person has been recently employed in a big company. This company consists of different departments. He works there for a limited time. He has got a laptop for work. In the company which he works for, there are many smart devices like printers connected to the internet, smart card readers, faxes and other devices that he needs to connect his laptop to them. Since he is new in the company and none of these devices are known to his laptop he needs to start a connection between them to receive services from them. Services provided by these devices are from different service levels and these services will be released gently.

In this example, 30 smart things are assumed in the company. Groups defined for this company are colleagues group, a group of people in the same location (for example, two things in a room), a group of people with the same brand. Each object provides 2 services with 3 levels. To select a node among all nodes, $\max(T_{ij})$ is used that *i* refers to the current node or laptop and j refers to direct or indirect nodes and T_{ij} is computed by equation 10.

$$T_{ij} = 0.3333 \times T_{ij}^{feedback} \times \left(T_{ij}^{intimac} + T_{ij}^{sociability} + T_{ij}^{importance}\right) \quad (10)$$

This equation is the multiplication of service feedback value and 0.3333, and summation of three trust properties. The result is a number between 0 and 1. Fig. 9 shows the model operation with 50 transactions. We need a node which has the highest trust value according to equation 10. It is obvious that the node with the highest total trust will always be selected.

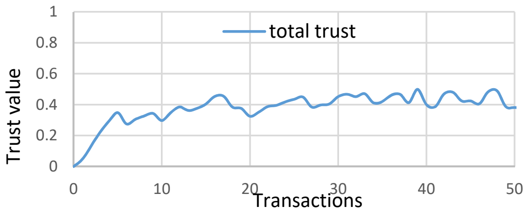


Fig. 9: computing total trust in an application.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper, a trust management approach in SIoT is presented that computes service provider's trust based on 4 properties including service feedback, intimacy, sociability, and transaction importance. The model is proposed to face with on-off and OSA attacks considering the level of services. In the simulations, the total trust value is computable by four trust properties. There are multiple levels for services. If a service provider presents a good service, then its service point will raise in the future and will receive higher service levels. To deal with on-off attacks, trust predictability model is proposed that keeps track of nodes behavior in a window to predict their behavior in the future. If a node is recognized as a malicious node, it will be inserted into a blacklist. Moreover, an example for trust evaluation is presented where a combination of trust properties has been taken into account to compute the total trust.

As the future works: the model can be implemented and analyzed for other applications presented in section V.6. This model and the network can be implemented in a real environment and compared with the simulations results. Other parameters such as consumed energy and memory usage can be considered in the computations and can find solutions to reduce energy consumption and memory usage as well.

REFERENCES

1. K. Routh and T. Pal, "A survey on technological, business and societal aspects of Internet of Things by Q3, 2017," in 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), 2018, pp. 1-4.

2. N. Bui and M. Zorzi, "Health care applications: a solution based on the internet of things," presented at the Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies, Barcelona, Spain, 2011.

3. A. J. Jara, M. A. Zamora, and A. F. Skarmeta, "An internet of things-based personal device for diabetes therapy management in ambient assisted living (AAL)," Personal Ubiquitous Comput., vol. 15, no. 4, pp. 431-440, 2011.

4. L. Xu, L. Rongxing, L. Xiaohui, S. Xuemin, C. Jiming, and L. Xiaodong, "Smart community: an internet of things application," Communications Magazine, IEEE, vol. 49, no. 11, pp. 68-75, 2011.

5. Y. Kung, S. Liou, G. Qiu, B. Zu, Z. Wang, and G. Jong, "Home monitoring system based internet of things," in 2018 IEEE International Conference on Applied System Invention (ICASI), 2018, pp. 325-327.

6. D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services," IEEE Transactions on Services Computing, vol. 3, no. 3, pp. 223-235, 2010.

7. L. Atzori, A. Iera, and G. Morabito, "SIoT: Giving a Social Structure to the Internet of Things," IEEE Communications Letters, vol. 15, no. 11, pp. 1193-1195, 2011.

8. A. P. Fiske, "The four elementary forms of sociality: framework for a unified theory of social relations," Psychological review, vol. 99, no. 4, p. 689, 1992.

9. L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl5, and H.-W. Gellersen, "Smart-Its Friends: A

Technique for Users to Easily Establish Connections between Smart Artefacts," in Ubicomp 2001: Ubiquitous Computing: International Conference Atlanta Georgia, USA, September 30–October 2, 2001 Proceedings, G. D. Abowd, B. Brumitt, and S. Shafer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 116-122.

10. 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, 11/14/ 2012.

11. 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, 6// 2014.

12. I. 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, no. 6, pp. 684-696, 2016.

13. R. Zhou, K. Hwang, and M. Cai, "GossipTrust for Fast Reputation Aggregation in Peer-to-Peer Networks," IEEE Trans. on Knowl. and Data Eng., vol. 20, no. 9, pp. 1282-1295, 2008.

14. A. A. Selcuk, E. Uzun, and M. R. Pariente, "A reputation-based trust management system for P2P networks," in Cluster Computing and the Grid, 2004. CCGrid 2004. IEEE International Symposium on, 2004, pp. 251-258.

15. C. Yang, H. Shuang, C. Gongliang, and L. Jianhua, "An improved trust and authorization mechanism for P2P file sharing system," in 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), 2017, pp. 1080-1084.

16. X. Meng and D. Liu, "GeTrust: A Guarantee-Based Trust Model in Chord-Based P2P Networks," IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 1, pp. 54-68, 2018.

17. J. Hendler and J. Golbeck, "Metcalfe's law, Web 2.0, and the Semantic Web," Web Semant., vol. 6, no. 1, pp. 14-20, 2008.

18. H. Mayadunna and L. Rupasinghe, "A Trust Evaluation Model for Online Social Networks," in 2018 National Information Technology Conference (NITC), 2018, pp. 1-6.

19. k. Ashton, "That 'Internet of Things' Thing , available at http://www.rfidjournal.com/articles/view?4986," RFID journal, 2009.

20. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol. 29, no. 7, pp. 1645-1660, 9// 2013.

21. B. Zhuming, X. Li Da, and W. Chengen, "Internet of Things for Enterprise Systems of Modern Manufacturing," Industrial Informatics, IEEE Transactions on, vol. 10, no. 2, pp. 1537-1546, 2014.

22. M. Nitti, R. Girau, and L. Atzori, "Trustworthiness Management in the Social Internet of Things," Knowledge and Data Engineering, IEEE Transactions on, vol. 26, no. 5, pp. 1253-1266, 2014.

23. Y. Ben 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, vol. 39, Part B, pp. 351-365, 11// 2013.

24. F. Bao and I.-R. Chen, "Dynamic trust management for internet of things applications," presented at the Proceedings of the 2012 international workshop on Self-aware internet of things, San Jose, California, USA, 2012.

25. D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang, "TRM-IoT: A trust management model based on fuzzy reputation for internet of things," Comput. Sci. Inf. Syst., vol. 8, no. 4, pp. 1207-1228, 2011.

26. W. Sherchan, S. Nepal, and C. Paris, "A survey of trust in social networks," ACM Comput. Surv., vol. 45, no. 4, pp. 1-33, 2013.

27. M. S. Islam, R. H. Khan, and D. M. Bappy, "A hierarchical intrusion detection system in wireless sensor networks," Computer Science Netw.Security, vol. 10, no. 8, pp. 21–26, 2010.

28. A. P. R. d. Silva, M. H. T. Martins, B. P. S. Rocha, A. A. F. Loureiro, L. B. Ruiz, and H. C. Wong, "Decentralized intrusion detection in wireless sensor networks," presented at the Proceedings of the 1st ACM international workshop on Quality of service &amp; security in wireless and mobile networks, Montreal, Quebec, Canada, 2005.

29. L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," Computer Networks, vol. 54, no. 15, pp. 2787-2805, 10/28/ 2010.

30. E. Borgia, "The Internet of Things vision: Key features, applications and open issues," Computer Communications, vol. 54, pp. 1-31, 12/1/ 2014.