

A Survey of Trust Computation Models for Service Management in Internet of Things Systems

Jia Guo[†], Ing-Ray Chen[†], and Jeffrey J.P. Tsai*

[†]Virginia Tech
Department of Computer Science
{jiaguo, irchen}@vt.edu

*Asia University
Department of Bioinformatics and Biomedical Engineering
jjptsai@gmail.com

Abstract

In this paper we survey trust computation models for Internet of things (IoT) systems for the purpose of service management, i.e., whether or not to select an IoT device as a service provider. Future IoT systems will connect the physical world into cyberspace everywhere and everything via billions of smart objects, and are expected to have a high economic impact. To date there is little work on trust computation in IoT environments for service management, especially for dealing with misbehaving owners of IoT devices that provide services to other IoT devices in the system. Our approach is to classify existing trust computation models for service management in IoT systems based on five essential design dimensions for a trust computation model: trust composition, trust propagation, trust aggregation, trust update, and trust formation. We summarize pros and cons of each dimension's options, and highlight the effectiveness of defense mechanisms against malicious attacks. We also summarize the most, least, and little visited trust computation techniques in the literature and provide insight on the effectiveness of trust computation techniques as applying to IoT systems. Finally, we identify gaps in IoT trust computation research and suggest future research directions.

Keywords: Internet of things; service-oriented computing; service management; trust; classification.

1 INTRODUCTION

It is envisioned that a future Internet of Things (IoT) system will connect a great amount of smart objects in the physical world, including radio frequency identification (RFID) tags, sensors, actuators, PDAs, and smartphones, as well as virtual objects in cyberspace such as data and virtual desktops on the cloud [26] [52] [57] [59]. The emerging paradigm of IoT has attracted a wide variety of applications running on top of it, including e-health [33], smart-home, smart-city, and smart-community [65].

A service-oriented IoT system can be viewed as a peer-to-peer (P2P) owner-centric community with devices (owned by humans) requesting and providing services on behalf of the owners, and with devices establishing social relationships autonomously with other devices based on social rules set by their owners, as well as

interacting with each other opportunistically as they come into contact. To best satisfy the service requester and maximize application performance, it is crucial to evaluate the trustworthiness of service providers in IoT environments. The motivation of providing a trust management system for IoT systems is clear: there are misbehaving owners and consequently misbehaving devices that may perform discriminatory attacks based on their social relationships with others for their own gain at the expense of other IoT devices which provide similar services. Further, misbehaving nodes with close social ties may collude and monopoly a class of services. Since trust provisioning in this environment inherently is fully integrated with service provisioning, the notion of trust-based service management is of paramount importance [34] [45].

The open issue to solve is to devise an effective and efficient trust computation method for an IoT device acting as a service requester to dynamically assess the service trustworthiness of another IoT device acting as a service provider, taking into consideration of the service history (from either self-observations or recommendations) of the target IoT device, and its own social relationships with that target IoT device. An IoT system can be viewed as a mix of P2P MANETs, social networks, and service computing systems where “things” autonomously establish social relationships according to the owners’ social network, and seek trusted “things” that can provide services needed when they come into contact with each other opportunistically in both the physical world and cyberspace [8]. Existing trust protocols for P2P MANETs [20] [35] [37] [53] [61] do not consider social relationships of IoT device owners. Existing trust protocols for social networks [1] [23] [58] consider social relationships. However, they do not consider service experiences and node mobility. Existing trust management protocols for service computing systems [29] [44] consider service experiences. However they do not consider node mobility and social relationships. Lastly unlike existing trust protocols for P2P MANETs, social networks, and service computing systems, a feasible trust protocol for an IoT system must deal with scalability and heterogeneity because potentially an IoT system will have a huge number (e.g., millions) of IoT devices some of which may be resource-constrained (e.g., smart phones, sensors, etc. that are intermittently connected to the Internet with no energy replenishment most of the time) and some of which may be resource-rich (e.g., a server that is connected to the power source and the Internet all the time). Hence, unlike its counterparts for P2P MANETs, social networks, and service computing systems, trust computation for IoT remains an open issue as it needs to consider scalability, node mobility, social relationship, and service experiences altogether.

To date there is limited work on trust computation in IoT environments for security enhancement [66], especially for dealing with misbehaving owners of IoT devices that provide services to other IoT devices in the system. It is noteworthy that unlike Yan et al. [66] which surveyed architectures and technologies (including trust evaluation, trust framework, privacy preservation computation, data, user, application, and communication trust) to enable trustworthy IoT, we survey only trust computation models for the purpose of service management (i.e., whether to select an IoT device as a service provider) in service-oriented IoT systems. Our approach is to classify

existing trust computation models based on five design dimensions considered essential for a trust computation model: trust composition, trust propagation, trust aggregation, trust update, and trust formation [20]. We summarize advantages and drawbacks of each dimension's options, and highlight the effectiveness of defense mechanisms against malicious attacks. We also summarize the most and least studied trust computation techniques in the literature and provide insight on the effectiveness of trust computation techniques as applying to service-oriented IoT systems. Finally, we identify gaps in IoT trust computation research and suggest future research areas.

The rest of the paper is organized as follows: Section 2 develops a classification tree for organizing existing trust computation techniques for IoT systems and explains the dimensions used. Section 3 develops a threat model and presents defense mechanisms developed in the literature against malicious attacks. Section 4 classifies existing IoT trust computation techniques for service management following the classification tree developed. In Section 5, we summarize the most and least studied IoT trust computation techniques in the literature. We provide insight on the effectiveness of trust computation techniques as applying to IoT systems and identify research gaps that are worthy of further research efforts. Section 6 presents our conclusion and suggests future research directions.

2 CLASSIFICATION TREE

In this section, we develop a classification tree for classifying trust computation techniques. The intent is to identify research gaps in IoT trust computation research. In this section we only classify existing IoT trust computation models. The detailed descriptions of existing IoT trust computation models for IoT systems will be given further in Section 4.

Figure 1 shows the classification tree based on five design dimensions: trust composition, trust propagation,

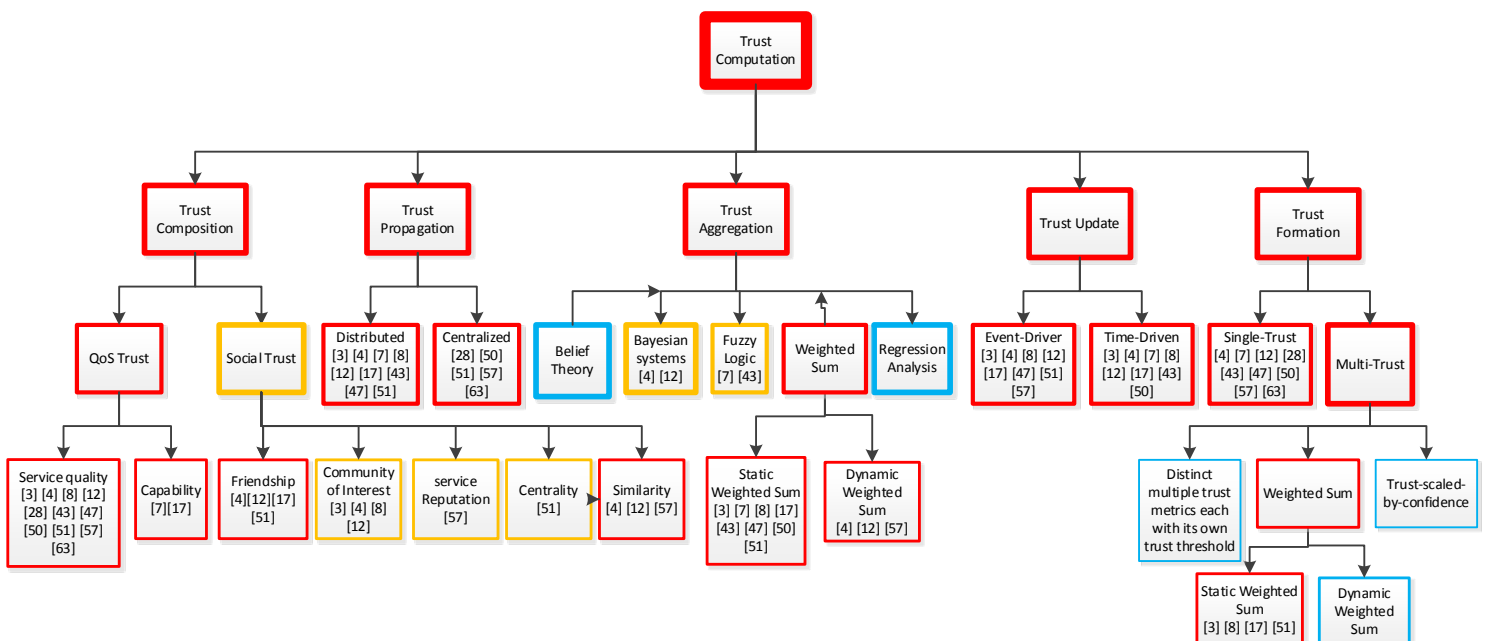


Figure 1: Classification Tree.

trust aggregation, trust update, and trust formation. These 5 dimensions are considered essential for a trust computation model [20]. It is color coded with red indicating the most visited, yellow for least visited, and blue for little visited. In each color class, we use thick vs. thin line format to differentiate the amount of exposure. Also at the bottom leaf level, we put in references for works that have used the approach. Below we discuss each classification design dimension in detail.

2.1 Trust Composition

Trust composition refers to what components to consider in trust computation. Trust components include quality of service (QoS) trust and social trust.

2.1.1 QoS Trust

QoS trust refers to the belief that an IoT device is able to provide quality service in response to a service request. QoS trust in general refers to performance and is measured by competence, cooperativeness, reliability, task completion capability, etc. Nitti et al. [51] used transaction performance to measure QoS trust. Chen et al. [7] used end-to-end packet forwarding ratio, energy consumption, and packet delivery ratio to measure QoS trust.

2.1.2 Social Trust

Social trust derives from social relationship between owners of IoT devices and is measured by intimacy, honesty, privacy, centrality [24], and connectivity. Chen et al. [12] made use of friendship, social contact, and community of interest (CoI) to rate a rater. Chen et al. [13] measured social trust by connectivity, intimacy, honesty and unselfishness. Social trust is especially prevalent in social IoT systems where IoT devices must be evaluated not only based on QoS trust, i.e., a device's capability to execute a service request, but also based on social trust, i.e., a device's commitment and good will to perform a service request. Moreover, when taking in a recommendation, an IoT device may trust its socially connected devices (of their owners) over unrelated devices.

2.2 Trust Propagation

Trust propagation refers to how to propagate trust evidence to peers. In general, there are two trust propagation schemes – that is, distributed and centralized.

2.2.1 Distributed

Distributed trust propagation refers to IoT devices autonomously propagating trust observations to other IoT devices they encounter or interact with without the use of a centralized entity. This is particularly the case in which it is difficult to setup or access a centralized entity in IoT environments mimicking a mobile ad hoc network (MANET) and/or a wireless sensor network (WSN). Chen et al. [12] proposed a distributed trust propagation scheme for social IoT systems. In Chen et al. [7], each node in the network maintains a data forwarding information table by overhearing activities of its neighboring nodes.

2.2.2 Centralized

Centralized trust propagation requires the presence of a centralized entity, either a physical cloud or a virtual trust service implemented by participating IoT devices. Nitti et al. [51] proposed a distributed hash table structure

to store node trust feedbacks and answer queries for node trust. Saied et al. [57] proposed a centralized trust manager keeping trust information of IoT entities and selecting capable IoT devices for answering a service request.

2.3 Trust Aggregation

Trust aggregation refers to aggregating trust evidence collected through either self- observations or feedbacks from peers. Major trust aggregation techniques investigated in the literature Josang et al. [32] include weighted sum, belief theory, Bayesian inference (with belief discounting), fuzzy logic, and regression analysis.

2.3.1 Weighted Sum

Weighted sum is a popular technique to aggregate evidence. Many reputation systems [56] [60] [68] aggregate ratings or feedbacks using weighted sum such that raters with a higher reputation or transaction relevance have a higher weight. Martinez-Zulia and Skarmeta [46] used *credibility* (derived from QoS and social trust) as the weight associated with the recommendation or feedback provided by a rater for indirect trust aggregation. Chen et al. [12] also used similarity (derived from social trust) as the weight for indirect trust aggregation. Weighted sum can also be used to aggregate direct trust (through self-observations) with indirect trust (through feedbacks or recommendations) for the same trust property (e.g., service quality). There is a further classification of whether the weights assigned to direct trust and indirect trust can be dynamically adjusted or just static at design time.

2.3.2 Belief Theory

Belief theory, also known as evidence theory or Dempster–Shafer theory (DST), is a general framework for reasoning with uncertainty, with connections to other frameworks such as probability, possibility and imprecise probability theories.

Dempster–Shafer theory is based on two ideas: obtaining degrees of belief for one question from subjective probabilities for a related question, and Dempster's rule [22] for combining such degrees of belief when they are based on independent items of evidence. In essence, the degree of belief in a proposition depends primarily upon the number of answers (to the related questions) containing the proposition, and the subjective probability of each answer. Yu and Singh [67] adopted Dempster-Shafer Theory as the underlying trust computational model to compute trust of agents in autonomous systems.

2.3.3 Subjective Logic

Subjective logic operates on subjective beliefs about the world, and uses opinions to denote the representation of a subjective belief [31]. The basic idea is to model trust by belief, disbelief and uncertainty. Josang et al. [32] described a node's opinion in another node by (b, d, u, a) where b , d , and u represent belief, disbelief, and uncertainty, respectively, with $b+d+u=1$, and a is the base rate probability in the absence of evidence. The average trust is therefore the probability expectation value computed as $b+au$. Subjective logic operators such as the discount and consensus operators are used to combine opinions (either self-observations or

recommendations).

2.3.4 Certain Logic

Ries et al. developed Certain Logic [55] for the evaluation of propositional logic terms under uncertainty that is compliant with the standard probabilistic approach and subjective logic. Based on Certain Logic, Ries [54] proposed a P2P trust model called CertainTrust that allows agents to choose trustworthy partners for risk engagement. The key feature of CertainTrust is that it is capable of expressing the certainty of a trust opinion, depending on the context of use. A trust opinion o_A about the truth of a proposition A is given as $o_A = (t, c, f)$ where the parameters are called average rating $t \in [0, 1]$, certainty $c \in [0, 1]$, and initial expectation value $f \in [0, 1]$. The average rating t indicates the degree to which previous observations support the truth of the proposition. It is associated to the relative frequency of observations supporting the truth of the proposition. The certainty c indicates the degree to which the average rating is assumed to be representative for the future. It is associated to the number of past observations or collected evidence units. The higher the certainty of an opinion is, the higher is the influence of the average rating on the expectation value in relation to the initial expectation. When the level of certainty ($c = 1$) is maximum, the average rating is assumed to be representative for the future outcomes. The initial expectation f expresses the assumption about the truth of a proposition in absence of evidence. The expectation value of an opinion $E(t, c, f) \in [0, 1]$ is calculated by $E(t, c, f) = t * c + (1 - c) * f$, representing the expectation about the truth of the proposition that has taken into account the initial expectation, the average rating, and the certainty.

2.3.5 Bayesian Inference with Belief Discounting

Bayesian inference treats trust as a random variable following a probability distribution with its model parameters being updated upon new observations. It is a popular trust computational model because of its simplicity and sound statistical basis. Josang et al. [33] proposed a Beta reputation system based on Bayesian inference with the trust value modeled as a random variable in the range of $[0, 1]$ following Beta distribution; the amounts of positive and negative experiences are mapped to the (α, β) parameters in Beta distribution so that the average trust is computed as $\frac{\alpha}{\alpha + \beta}$. Ganeriwal et al. [25] applied Bayesian inference for a reputation system in a WSN, taking binary positive and negative ratings as input, and computing sensor node reputation scores. Belief discounting is applied to defend against bad-mouthing attacks (saying a good node as a bad node) and ballot-stuffing attacks (saying a bad node as a good node). Specifically let node i be the trustor, node j be the trustee, and node k be a recommender. Also let $(\alpha_{i,j}, \beta_{i,j})$ be the trustor's (α, β) toward the trustee, $(\alpha_{k,j}, \beta_{k,j})$ be the recommender's (α, β) toward the trustee and $(\alpha_{i,k}, \beta_{i,k})$ be the trustor's (α, β) toward the recommender. Based on belief discounting (see the detail in Ganeriwal et al. [25] and Josang et al. [33]), node i will compute its new $(\alpha_{i,j}^{\text{new}}, \beta_{i,j}^{\text{new}})$ as follows:

$$\alpha_{i,j}^{\text{new}} = \alpha_{i,j} + \frac{2\alpha_{i,k}\alpha_{k,j}}{[(\beta_{i,k} + 2)(\alpha_{k,j} + \beta_{k,j} + 2)] + 2\alpha_{i,k}} \quad (1)$$

$$\beta_{i,j}^{\text{new}} = \beta_{i,j} + \frac{2\alpha_{i,k}\beta_{k,j}}{[(\beta_{i,k} + 2)(\alpha_{k,j} + \beta_{k,j} + 2)] + 2\alpha_{i,k}} \quad (2)$$

The basic idea is that if node i does not trust k , it will discount the recommendation provided by node k , so $\alpha_{i,j}^{\text{new}} \sim \alpha_{i,j}$ and $\beta_{i,j}^{\text{new}} \sim \beta_{i,j}$ as if the recommendation from k does not have any effect. This can be derived from (11) and (12). First of all, if node i does not trust node k then $\alpha_{i,k} \ll \beta_{i,k}$. In case node k is performing a bad-mouthing attack on node j , then $\alpha_{k,j} \ll \beta_{k,j}$. Applying these two conditions to (11) and (12), one can easily verify $\alpha_{i,j}^{\text{new}} \sim \alpha_{i,j}$ and $\beta_{i,j}^{\text{new}} \sim \beta_{i,j}$. In case node k is performing a ballot-stuffing attack on node j , then $\alpha_{k,j} \gg \beta_{k,j}$ and again one can easily verify $\alpha_{i,j}^{\text{new}} \sim \alpha_{i,j}$ and $\beta_{i,j}^{\text{new}} \sim \beta_{i,j}$. After trust aggregation, the trustor's (or node i 's) trust toward the trustee (or node j) is then computed as $\frac{\alpha_{i,j}^{\text{new}}}{\alpha_{i,j}^{\text{new}} + \beta_{i,j}^{\text{new}}}$.

2.3.6 Fuzzy Logic

Fuzzy logic is a form of many-value logic; it deals with reasoning that is approximate rather than fixed and exact. Compared to traditional binary sets, fuzzy logic variables may have a truth value that ranges in degree between 0 and 1. Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false. Furthermore, when linguistic variables are used, these degrees may be managed by specific membership functions. Reputation or trust is represented as a fuzzy measure with membership functions describing the degrees of trust, e.g., a trust value in the range $(-1.25, 1.25)$ denotes very low trust, $(0, 2.5)$ low trust, $(1.25, 3.75)$ medium trust, $(2.5, 5)$ high trust, $(3.75, 6.25)$ high trust, and so on. Hence, a node with a trust value of 0.25 is 75% very low trust (a membership function) and 25% low trust (another membership function). Fuzzy logic provides rules for reasoning with fuzzy measures. Chen et al. [7] used a fuzzy membership function taking into consideration of the number of positive and negative experiences together with uncertainty to compute trust. More detail is discussed in Section 4.4.

2.3.7 Regression Analysis

Regression analysis is a statistical process for estimating the relationships among variables. It can be applied to estimate the relationships between trust and a set of variables characterizing the behavior of a node. Wang et al. [64] applied logit regression to learn the relation between cumulative evidence gathered by node i toward node j and the corresponding environmental context variables including energy-sensitivity, capability-limitation, and cost-awareness. This behavior pattern is learned dynamically and can be used to predict node j 's trust behavior, i.e., whether node j , from the perspective of node i , can provide good service when service is called for, given an environment setting characterized by a set of context variable values as input.

2.4 Trust Update

Trust update concerns when trust is updated. In general, there are two schemes - event-driven scheme and

time-driven scheme.

2.4.1 Event-driven

In the event-driven scheme, all trust data in a node get updated after a transaction or event is made. This can be when a service is rendered and therefore a feedback regarding service quality is sent to the trust manager in the cloud, or recorded in each node cache for trust aggregation. A recommendation can also be sent upon request in encounter-based environments [14] where nodes run into each other and request for recommendations for other nodes.

2.4.2 Time-driven

In the time-driven scheme, evidence (self-observations or recommendations) is collected periodically and trust is updated by applying a trust aggregation technique. In case no evidence is collected, trust decay over time is frequently applied because one should trust recent information more than past information. The exponential decay function with a parameter adjusting the rate of trust decay over time can be used depending on the specific application needs [10].

2.5 Trust Formation

Trust formation refers to how to form the overall trust out of multiple trust properties. In the literature, trust formation is considered from the aspect of single-trust or multi-trust.

2.5.1 Single-trust

Single-trust refers to the fact that only one trust property is considered in a trust protocol. For example, service quality is deemed the single most important metric in service-oriented IoT systems [12]. Therefore, an IoT device is being evaluated on its ability to produce quality service when called for. In a social IoT system, the service quality may be affected by the relationship between the service requester and the service provider, so inherently trust in service quality in a social IoT system is pair-wise. In other words, a node is more concerned with one-to-one trust toward another node based on their relationship, rather than the general reputation of the node derived from the public belief.

2.5.2 Multi-trust

Multi-trust implements the common belief that trust is multidimensional, so multiple trust properties should be considered for trust formation. Chen et al. [13] considered multiple trust properties including intimacy, honesty, unselfishness, and competence to assess the overall trust of a MANET node. There are multiple ways to do trust formation:

- One can just use individual trust properties without combining them together but define a minimum threshold for each trust property depending on the application requirements. For example, honesty is important, so a high threshold is used but competence may not be very critical so a low threshold is set.
- One can use *weighted sum* to combine individual trust properties together into an overall trust metric. The weight assigned can reflect the application requirements. For example, honesty is important so a high weight is

used. Furthermore, the weight assignment may be dynamically adjusted to reflect environmental situation awareness. For example in a hostile environment where attacks are likely, the weight associated with honesty can be set high so as to effectively defend malicious attacks (such as bad-mouthing and ballot-stuffing attacks). On the other hand in a friendly environment such as in a club setting, competence is more important than honesty, so a higher weight can be placed for competence instead. Chen et al. [12] proposed to readjust the weights of direct and indirect trust to maximize the average user satisfaction experiences during the most recent time period. Saied et al. [57] proposed to change the weight associated with positive recommendations dynamically to derive the overall trust value. Liu et al. [41] [42] proposed to use a penalty coefficient weight based on the authentication history to update trust.

One can use a *trust-scaled-by-confidence* technique for trust formation. The idea is to scale the most important trust property with less important trust properties which serve as confidence. Wang et al. [65] considered competence and integrity as two trust properties for rating a node with competence being the more important trust metric. It considered two scaling schemes: (a) competence trust drops to zero if integrity trust falls below a threshold; (b) competence trust scales up (to 1 maximum) or down (to 0 minimum), depending on whether integrity trust is higher or lower than the threshold.

3 THREAT MODEL

Trust as a soft security measure is particularly applicable to service-oriented IoT systems because IoT devices owned by human beings inherently can be malicious for their own gain. Malicious users can also collude to dominate the service provider market. In this section, we enumerate possible threats to IoT systems. Our intent is to survey how existing IoT trust management protocols in the literature deal with malicious attacks and identify gaps for future research.

It is noteworthy that the threat model covers threats that will disrupt the trust system. While trust can be used to enhance security of IoT systems [40], the threat model discussed here does not cover threats to security in general.

In an IoT system, every IoT device can be a service provider (SP) or a service requester (SR) itself. Therefore every IoT device wants to be selected to provide service for profit when it is a SP and wants to find the best SPs for best service available when it is a SR. A malicious SP node acts for its own benefit and would like to be selected for service even if the service provides 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 [30]. 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. Thus, a malicious IoT device (because its owner is malicious) can perform the following trust-related attacks:

1. *Self-promotion attacks* (SPA): a malicious node it can promote its importance (by providing good recommendations for itself) so as to be selected as a SP, but then can provide bad or malfunctioned service.
2. *Bad-mouthing attacks* (BMA): a malicious node can ruin the trust of a well-behaved node (by providing bad recommendations against it) so as 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.
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 a SP. This is another form of collusion recommendation attacks, i.e., it can collaborate with other malicious nodes to boost the trust of each other.
4. *Opportunistic service attacks* (OSA): 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. With good reputation, it can effectively collude with other bad node to perform bad-mouthing and ballot-stuffing attacks.
5. *On-off attacks* (OOA): instead of always performing best service, a malicious node can perform bad service. With on-off attacks, a malicious node performs bad service on and off (or randomly) so as to avoid being labeled as a low trust node and risk itself not being selected as a SP, as well as not being able to effectively perform bad-mouthing and ballot-stuffing attacks. One can view on-off attacks as random attacks.

A collaborative attack means that the malicious nodes in the system boost their allies and focus on particular victims in the system to victimize. We note that bad-mouthing and ballot-stuffing attacks are a form of collaborative attacks to the trust system to ruin the reputation of (and thus to victimize) good nodes and to boost the reputation of malicious nodes.

4 SURVEY AND CLASSIFICATION OF EXISTING IOT TRUST COMPUTATION MODELS

IoT trust computation is still in its infancy with limited work reported in the literature to date, possibly due to limited experiences with IoT platforms and experimentations. We only found [3] [4] [7] [8] [12] [17] [28] [43] [47] [50] [51] [57] and [63] in the literature to date. In particular, [28] [63] do not consider all five design dimensions in trust composition, trust propagation, trust aggregation, trust update, and trust formation. In this section, we follow the classification tree to classify existing IoT trust computation models based on the techniques used in trust composition / trust propagation / trust aggregation / trust update / trust formation. Based on the classification tree, we identify 8 classes as summarized in Table 1. It is noteworthy that the classification is based on the underlying trust techniques adopted in the five design dimensions. So works will fall into the same class only if they use identical trust techniques in all five design dimensions. A deviation of trust techniques used in just one design dimension will put the works in separate classes. By this way, we can compare

trust computation models class by class and identify the most effective class for trust computation for service management (whether to select a device as a service provider) in service-oriented IoT systems. For works that fall under the same class, we also survey defense mechanisms used (if any) to defend against malicious attacks discussed in Section 3.

We discuss these 8 classes identified in the 8 subsections below, with the subsection title reflecting the classification based on the techniques used in trust composition / trust propagation / trust aggregation / trust update / trust formation. A missing dimension is denoted by “-” in the class name (as in the last entry of Table 1).

Table 1: Eight Classes of IoT Trust Computation Models based on the Techniques used in Trust Composition / Trust Propagation / Trust Aggregation / Trust Update / Trust Formation.

Classification	Work	SPA	BMA	BSA	OSA	OOA
QoS + Social / Distributed / Bayesian inference + Dynamic weighted sum / Event + Time-driven / Single-trust (Section 4.1)	2013 Bao, et al. [4], and 2015 Chen, et al. [12]	Direct service quality trust assessment and feedback propagation	Social similarity to rate a recommender	Social similarity to rate a recommender	Adaptive filtering to adjust the weights of direct and indirect service quality trust dynamically	NA
QoS + Social / Distributed + Centralized / Static weighted sum / Event-driven / Multi-trust with static weighted sum (Section 4.2)	2014 Nitti, et al. [51]	Direct service quality trust assessment and feedback propagation	Credibility to rate a recommender	Credibility to rate a recommender	Long-term and short-term direct service quality trust assessment	NA
QoS / Centralized / dynamic weighted sum / Event-driven / Single-trust (Section 4.3)	2014 Saied, et al. [57]	Direct service quality trust assessment and feedback propagation	Recommender trust to rate a recommender	Recommender trust to rate a recommender	NA	NA
QoS / Distributed / Fuzzy logic + Static weighted sum / Time-driven / Single-trust with static weighted sum (Section 4.4)	2011 Chen, et al. [7]	Direct service quality trust assessment and feedback propagation	NA	NA	NA	NA
	2013 Mahalle, et al. [43]	NA	NA	NA	NA	NA
QoS + Social / Distributed / Static weighted sum / Event + Time-driven / Multi-trust with static weighted sum (Section 4.5)	2012 Bao, et al. [3], and 2015 Chen, et al. [8]	Honesty trust assessment and feedback propagation	Honesty trust assessment and feedback propagation	Honesty trust assessment and feedback propagation	NA	NA
	2015 Chen, et al. [17]	Direct service quality trust assessment and feedback propagation	Recommender trust to rate a recommender	Recommender trust to rate a recommender	NA	NA
QoS / Distributed / Static weighted sum / Event-driven / Single-trust (Section 4.6)	2015 Mendoza, et al. [47]	NA	NA	NA	NA	Reward and punishment scheme
QoS / Centralized / Static weighted sum / Time-driven / Single-trust with static weighted sum (Section 4.7)	2015 Namal, et al. [50]	NA	NA	NA	NA	NA
QoS / Centralized / - / - / Single-trust (Section 4.8)	2013 Gu, et al. [28] and Wang, et al. [63]	NA	NA	NA	NA	NA

4.1 Class 1: QoS + Social / Distributed / Bayesian inference + Dynamic weighted sum / Event + Time-driven / Single-trust





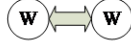



Trust Composition	 +  QoS Social
Trust Propagation	 Distributed
Trust Aggregation	 +  Bayesian inference Dynamic weighted sum
Trust Update	 +  Event-driven Time-driven
Trust Formation	 Single-trust

Figure 2: The General Approach Taken by Class 1.

Class 1 is characterized by the general approach of taking QoS and social for trust composition, distributed for trust propagation, Bayesian inference and dynamic weighted sum for trust aggregation, event and time-driven for trust update, and single-trust for trust aggregation, as illustrated in Figure 2

Among all works listed in Table 1, only Bao et al. [4] and Chen et al. [12] fall into this classification with Bao et al. [4] being the preliminary work of Chen et al. [12].

Bao et al. [4] and Chen et al. [12] used service quality (a QoS trust metric) to rate a SP, and social similarity (a social trust metric) to rate a recommender based on the concept of collaborative filtering to select feedbacks using similarity rating of friendship, social contact, and community of interest relationships as the filter.

In trust propagation, every node acts autonomously to collect evidence (through self-observations or recommendations) and also serves as a recommender upon request. Hence it is based on distributed trust propagation. A node first collects evidence of the service quality trust and social similarity trust of adjacent nodes. Then it collects recommendations from qualified adjacent nodes about other nodes in the system.

In trust aggregation, Bao et al. [4] and Chen et al. [12] used Bayesian inference to aggregate self-observations into direct trust. Social similarity-weighted sum is used to aggregate recommendations into indirect trust. A novel adaptive filtering technique is proposed to adjust weights associated with direct trust and indirect trust dynamically to minimize trust bias and maximize application performance.

In trust update, both event-driven and time-driven are considered. The direct trust is updated upon each service interaction while the indirect trust is updated periodically using peer recommendations collected during

the period. The work also considers and analyzes the effect of trust decay on trust convergence rate.

In trust formation, only a single service quality trust is considered, so it falls into the single-trust category. However, Bao et al. [4] and Chen et al. [12] used several social similarity metrics, i.e., friendship, social contact, and community-of-interest, and apply the weighted sum technique to combine these social similarity metrics into one to rate a recommender. The best weighting scheme to combine the three metrics into one is identified for a service-oriented IoT application.

Entry 1 of Table 1 summarizes the defense mechanisms used by Bao et al. [4] and Chen et al. [12] to defend against malicious attacks. SPA is detected in the protocol design and the feedback is propagated through trust propagation. BMA and BSA are tolerated by using social similarity to rate a recommender. OSA is resolved by adaptive filtering which dynamically adjust the weights associated with direct and indirect trust to capture the opportunistic service attack behavior. However, OOA is not considered.

4.2 Class 2: QoS + Social / Distributed + Centralized / Static weighted sum / Event-driven / Multi-trust with static weighted sum

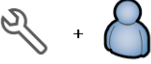
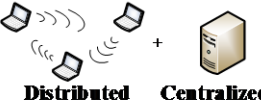



Trust Composition	 QoS Social
Trust Propagation	 Distributed Centralized
Trust Aggregation	 Static weighted sum
Trust Update	 Event-driven Time-driven
Trust Formation	 Single-trust

Figure 3: The General Approach Taken by Class 2.

Class 2 is characterized by the general approach of taking QoS and social for trust composition, centralized for trust propagation, static weighted sum for trust aggregation, event-driven for trust update, and multi-trust with static weighted sum for trust aggregation, as illustrated in Figure 3

The work by Nitti et al. [51] falls into this classification. In trust composition, Nitti et al. [51] considered both QoS trust and social trust. QoS trust properties considered include transaction service quality and computational capability. Social trust properties considered include centrality, relationship factor (such as ownership, co-

location, co-work, social and co-brand), and credibility. In particular, credibility is used to rate a recommender who provides indirect evidence and is computed by a weighted sum of the direct trust toward the recommender and the relative centrality of the recommender.

In trust propagation, the approach taken by Nitti et al. [51] is rather unique in that it considers both distributed and centralized models. In the distributed model, each node computes its own *subjective trustworthiness* toward another node. Transaction service quality trust is assessed by individual nodes after a transaction is completed, and feedbacks are propagated as indirect evidence from one node to another upon request. In the centralized model, transaction service quality feedbacks are propagated to a centralized entity making use of a dynamic hash (DHT) table structure on the network to maintain the *objective trustworthiness* status (global reputation) of a node. A node can query the DHT to receive the trust value of other nodes in the network, and the DHT returns the objective trustworthiness scores after searching the database.

In trust aggregation, Nitti et al. [51] applied *static weighted sum* to compute centrality trust, direct service quality trust, and indirect service quality trust separately. In particular, for direct service quality trust assessment, transaction relevance (along with relationship factor and computational capability to a lesser extent) is used as the weight. For indirect service quality trust assessment, credibility is used as the weight. The difference between subjective trustworthiness and objective trustworthiness is how evidence is collected. For subjective trustworthiness, a node uses the relative centrality for centrality trust assessment, self-observations for direct service quality trust assessment, and feedbacks provided to its peers for indirect service quality trust assessment of another node. For objective trustworthiness, the network centrality is used for centrality trust assessment, and all feedbacks are used for both direct and indirect service quality trust assessment.

In trust update, the event-driven scheme is taken. For the distributed model, a service receiver rates the service quality of a transaction at the end of the transaction, stores the rating in its local storage, and provides a feedback to its peers upon request. For the centralized model, the DHT collects the feedback and updates its trust database after the end of each transaction.

In trust formation, Nitti et al. [51] considered a multi-trust scheme by applying *static weighted sum* to combine centrality trust (which is a social trust property) with service quality trust (which is a QoS trust property obtained from direct service quality trust and indirect service quality trust) into an overall trust value.

For defending against attacks, Nitti et al. [51] used direct service quality trust assessment and feedback propagation for SPA, credibility rating for BMA and BSA, and differentiating long-term and short-term direct service quality trust assessment to change credibility to defend against OSA. However OOA is not considered.

4.3 Class 3: QoS / Centralized / dynamic weighted sum / Event-driven / Single-trust






Trust Composition	 QoS
Trust Propagation	 Centralized
Trust Aggregation	 Dynamic weighted sum
Trust Update	 Event-driven
Trust Formation	 Single-trust

Figure 4: The General Approach Taken by Class 3.

Class 3 is characterized by the general approach of taking QoS for trust composition, centralized for trust propagation, dynamic weighted sum for trust aggregation, event-driven for trust update, and single-trust for trust aggregation, as illustrated in Figure 4.

Saied et al. [57] proposed a reputation system for a service-oriented IoT system and falls into this classification. In trust composition, it considers service quality as the sole trust metric but uses context information [62] including service type and node capability (e.g., energy status) to associate a service quality rating. In trust propagation, it uses a centralized manager to store all reputation reports (with context information) sent by individual SRs, after service is rendered. Upon receiving a new service request, the centralized manager selects SP candidates based on the service context for servicing the request. It then uses only evaluation reports with similar service context for reputation assessment of each SP candidate. In trust aggregation, the service context similarity between a stored report and the target service is computed by a global contextual distance function. A higher weight is used if a higher service context similarity is found. The reputation score is then computed by *dynamic weighted sum* with the weight associated with a report corresponding to the *recommendation trust* of the recommender who supplies the report. The recommendation trust is updated dynamically based on if the recommender's report agrees or deviates from the majority of reports with a similar service context. Trust update is performed by the centralized manager via a learning process which presumably occurs whenever new reports are received, so it is based on event-driven. In trust formation, only service quality is considered.

Saied et al. [57] dealt with SPA by detection of service quality and feedback propagation. The centralized manager rates a recommender dynamically based on the degree to which the recommender’s report deviates from the majority reports. This can effectively defend against BMA and BSA, if the majority recommenders are not malicious. OSA and OOA are not considered.

4.4 Class 4: QoS / Distributed / Fuzzy logic + Static-weighted sum / Time-driven / Single-trust with static weighted sum



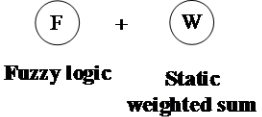


Trust Composition	 QoS
Trust Propagation	 Distributed
Trust Aggregation	 Fuzzy logic Static weighted sum
Trust Update	 Event-driven
Trust Formation	 Single-trust

Figure 5: The General Approach Taken by Class 4.

Class 4 is characterized by the general approach of taking QoS for trust composition, distributed for trust propagation, fuzzy logic and static-weighted sum for trust aggregation, time-driven for trust update, and single-trust with static weighted sum for trust aggregation, as illustrated in Figure 5.

Chen et al. [7] and Mahalle et al. [43] fall into this classification. In trust composition, Chen et al. [7] proposed a trust management model based on fuzzy logic and considered only QoS metrics including end-to-end packet forwarding ratio (EPFR), energy consumption (AEC), and packet delivery ratio (PDR). Mahalle et al. [43] proposed a fuzzy trust-based approach for access control [36] taking into consideration of three QoS factors, i.e., experience (EX), knowledge (KN), and recommendation (RC).

In trust propagation, trust is propagated in a distributed manner. Each node in the network maintains trust data of other nodes.

In trust aggregation, Chen et al. [7] proposed aggregating the overall trust of a node toward another node using a static weighted sum of direct trust based on direct interaction experiences, and indirect trust based on recommendations. The direct trust is computed by aggregating EPFR, AEC and PDR direct interaction evidence

using a static-weighted sum. If the aggregated trust value passes a threshold, the experience is a positive experience; otherwise, it is a negative experience. A fuzzy membership function taking into consideration of the number of positive and negative experiences together with uncertainty is used to compute the direct trust. The indirect trust on the other hand is computed by the product of the trustor's direct trust toward the recommender with the recommender's recommendation trust toward the trustee. The recommender's recommendation trust is computed in the same way as direct trust except that positive and negative recommendation experiences, together with uncertainty, are used in the fuzzy membership function definition. Since direct trust and recommendation trust are each defined by a fuzzy membership function, the overall trust can be aggregated by applying fuzzy logic "add" (for adding direct trust with indirect trust based on static weighted sum) and "multiply" (for multiplying direct trust with recommendation trust to obtain indirect trust) operators. In trust aggregation, Mahalle et al. [43] proposed that each trust component, EX, KN or RC, can be good, average or bad. EX is a normalized sum of all trust experiences of that node. KN is a weighted sum of direct knowledge and indirect knowledge. RC is also a weighted sum of recommendations from recommenders. Trust is computed through a fuzzy rule base.

In trust update, trust is updated periodically, so it is time-driven. In trust formation, only single-trust (service quality trust) is considered by either Chen et al. [7] or Mahalle et al. [43].

Regarding resiliency against attacks, Chen et al. [7] detected SPA and the feedback is propagated to the central manager through trust propagation. BMA, BSA, OSA and OOA were not considered. Mahalle et al. [43] did not consider SPA, BMA, BSA, OSA or OOA

4.5 Class 5: QoS + Social / Distributed / Static weighted sum / Event + Time-driven / Multi-trust








Trust Composition	 +  QoS Social
Trust Propagation	 Distributed
Trust Aggregation	 Static weighted sum
Trust Update	 +  Event-driven Time-driven
Trust Formation	 Multi-trust

Figure 6: The General Approach Taken by Class 5.

Class 5 is characterized by the general approach of taking QoS and social for trust composition, distributed for trust propagation, static weighted sum for trust aggregation, event and time-driven for trust update, and multi-trust for trust aggregation, as illustrated in Figure 6.

Bao et al. [3] and Chen et al. [8] fall into this classification. In trust composition, they considered separate trust properties, including QoS trust properties such as honesty and cooperativeness, and social trust such as community-interest.

In trust propagation, Bao et al. [3] and Chen et al. [8] followed the distributed scheme where each node maintains its own trust assessment towards other nodes and propagates its recommendation trust toward other nodes.

In trust aggregation, a trustor node aggregates its current trust toward the trustee node with new evidence based on weighted sum. The new evidence can be either direct evidence if the trustor node directly encounters and interacts with the trustee node or indirect evidence if it does not encounter the trustee node but receives a recommendation trust toward the trustee node from a recommender it encounters. In the latter case, the recommendation trust received is discounted by the trustor's direct trust toward the recommender. A novelty is that the weights associated with the past experience and the current evidence can be dynamically adjusted to tradeoff the trust convergence rate and trust fluctuation rate. Although the effect of weight parameters is analyzed, there is no discussion of how the weight parameters can be dynamically adjusted. Therefore they can be classified as static weighted sum at most.

In trust update, the trust management protocol is encounter-based as well as activity based, meaning that the trust value is updated upon an encounter event or an interaction activity. Two nodes encountering each other or involved in a direct interaction activity can directly observe each other and update their trust assessment. They also exchange their trust evaluation results toward other nodes as recommendations.

In trust formation, they considered multiple trust properties: honesty, cooperativeness and community-interest trust. However they did not discuss how to form the overall trust out of these separate trust properties.

Bao et al. [3] and Chen et al. [8] used honesty trust assessment and feedback propagation for defending against SPA, BMA and BSA. However, OSA and OOA are not considered.

Chen et al. [17] also falls into this classification. In trust composition, Chen et al. [17] considered both QoS trust (i.e., quality reputation and energy status) and social trust (i.e., social relationship using factors considered in Chen et al. [12] and Nitti et al. [51]). In trust propagation, Chen et al. [17] adopted a distributed scheme where each node maintains its own trust assessment towards other nodes. Evidence for each trust component is propagated separately. In trust aggregation, each trust component is assessed separately based on static weighted sum. In particular, quality reputation is a static weighted sum of direct trust and indirect trust. The indirect trust is also a feedback-trust-weighted sum of feedbacks received from all recommenders with each recommender's feedback trust being updated based on how the recommender's feedback deviates from the average. The social relationship trust and the energy trust are also each assessed by a static weight sum function. In trust update, it is event-driven with the trust value being computed based on transaction completion and timer events. In trust formation, it falls into the multi-trust category with the overall trust being formed from the three trust components, quality reputation, energy, and social relationship, based on static weighted sum.

4.6 Class 6: QoS / Distributed / Static-weighted sum / Event-driven / Single-trust






Trust Composition	 QoS
Trust Propagation	 Distributed
Trust Aggregation	 Static weighted sum
Trust Update	 Event-driven
Trust Formation	 Single-trust

Figure 7: The General Approach Taken by Class 6.

Class 6 is characterized by the general approach of taking only QoS for trust composition, distributed for trust propagation, static weighted sum for trust aggregation, event-driven for trust update, and single-trust for trust aggregation, as illustrated in Figure 7.

Mendoza et al. [47] falls into this classification. In trust composition, it considers direct QoS trust. When a node fulfills a service, it will be rewarded; otherwise, it will be punished. In trust propagation, trust is propagated in a distributed manner. Each node in the network maintains a data forwarding information table of other nodes. In trust aggregation, the overall trust of a node toward another node is just the direct trust derived from direct interaction experiences. In trust update, trust is updated after a service request event, so it is event-driven. In trust formation, only single-trust (i.e., service quality trust) is considered. No social trust is considered in this work.

OOA is mitigated by using a reward and punishment scheme. SPA, BMA, BSA and OSA are not considered.

4.7 Class 7: QoS / Centralized / Static-weighted sum / Time -driven / Single-trust with static weighted sum






Trust Composition	 QoS
Trust Propagation	 Centralized
Trust Aggregation	 Static weighted sum
Trust Update	 Time-driven
Trust Formation	 Single-trust

Figure 8: The General Approach Taken by Class 7.

Class 7 is characterized by the general approach of taking QoS for trust composition, centralized for trust propagation, static weighted sum for trust aggregation, time-driven for trust update, and single-trust with static weighted sum for trust aggregation, as illustrated in Figure 8.

Namal et al. [50] falls into this classification. In trust composition, it considers direct QoS trust metrics such as availability, reliability, irregularity, and capacity. In trust propagation, trust is propagated in a centralized manner. In trust aggregation, the overall trust of a node toward another node is aggregated using a static weighted sum of past experiences and new QoS sensor data on availability, reliability, and irregularity. In trust update, trust computation is performed periodically, so it is time-driven. In trust formation, only single-trust (service quality trust) is considered. No social trust is considered in this work.

4.8 Class 8: QoS / Centralized / - / - / Single-trust




Trust Composition	 QoS
Trust Propagation	 Centralized
Trust Aggregation	NA
Trust Update	NA
Trust Formation	 Single-trust

Figure 9: The General Approach Taken by Class 8.

Class 8 is characterized by the general approach of taking QoS for trust composition, centralized for trust propagation, and single-trust for trust formation, as illustrated in Figure 9. Trust aggregation and trust update are not considered.

Gu et al. [28] and Wang et al. [63] fall into this classification. In trust composition, Both considered QoS trust in three network layers, i.e., the sensor layer, core layer and application layer. The purpose of trust management for the sensor layer is selecting a subset of nodes based on their trust values to provide service. Trust management for the core layer considers historical trust, recommended experience, risk, ability of anti-attack and the service capability. It aims at acquiring a set of optimal routes in the network. The purpose of trust management for the application layer considers control attributes such as service efficiency, service risk, and service history. In trust formation, single-trust is considered based on the combination of sensor layer trust, core layer trust and application layer trust. Neither Gu et al. [28] nor Wang et al. [63] considered trust propagation or trust update. Also neither Gu et al. [28] nor Wang et al. [63] considered attack models.

5 RESEARCH GAPS

In this section, we identify research gaps in trust computation for IoT systems. We first summarize the most visited, least visited, and little visited trust computation methods, as summarized in Table 2. The rationale behind this is to identify the most popular trust composition, trust propagation, trust aggregation, trust update, and trust

formation techniques adopted by existing trust computation models, provide reasons for such popularity, and then identify research gaps that deserve more attention for IoT trust computation research.

Trust Composition: A modern IoT system is inherently socially oriented since IoT devices are owned by humans which are connected by social relationships. We see from Table 2 that most works indeed consider both QoS trust and social trust for trust composition. We take the view that a valid trust model for IoT must consider social trust metrics because social relationships between human operators control the way an IoT device would behave toward another IoT device. Among social trust properties, similarity and friendship are the most visited among all. The reason is that these two social properties arguably are the most important among all because friendship implies good service, and high social similarity implies high creditability of a recommendation. However, we argue that for certain IoT applications such as smart city air pollution detection and augmented map travel assistance [4], social metrics such as centrality and community of interest especially for rating recommenders, should be further explored to improve trust computation performance, including convergence, accuracy, and resiliency against malicious attacks.

Table 2: Most, Least and Little Visited IoT Trust Computation Models in the Literature.

Most visited	Trust composition	QoS trust [3][4][7][8][12][17][28][43][47][50][51][57][63]
		Social trust [3] [4] [8] [12] [17] [51] [57]
	Trust propagation	Distributed [3][4][7][8][12][17][43][47][51]
		Centralized [28][50][51][57][63]
	Trust aggregation	Static weighted sum [3] [7] [8] [17] [43] [47] [50] [51]
		Dynamic weighted sum [4][12][57]
	Trust update	Event-driven [3][4] [8][12][17][47][51][57]
		Time-driven [3][4][7][8][12][17][43][50]
	Trust formation	Single-trust [4] [7] [12] [28] [43] [47] [50] [57] [63]
		Multi-trust with static weighted sum [3] [8] [17] [51]
Least visited	Trust aggregation	Fuzzy logic [7][43]
		Bayesian inference[4][12]
Little visited	Trust aggregation	Belief theory
		Regression analysis
	Trust formation	Multi-trust with dynamic weighted sum
		Multi-trust each with its own minimum threshold
	Multi-trust with trust scaled by confidence	

Trust Propagation: Table 2 shows that existing work mostly considered distributed trust propagation for “subjective” trust computation without relying on a centralized entity. That is, each node propagates trust information to other nodes upon request, and each node also aggregates trust information (including self-observations) for trust assessment toward other nodes in the system. This is feasible for IoT systems where IoT

devices (e.g., smart phones, vehicles, etc.) are mobile with no access to a centralized entity such as a cloud. With the emergence of cloud services, however, centralize trust propagation for “objective” reputation computation (common belief of the public) makes more sense. The literature is limited in investigating cloud-based trust propagation methods. Only Gu et al. [28], Namal et al. [50], Nitti et al. [51], Saied et al. [57], and Wang et al. [63] discussed centralized trust propagation. Centralized trust propagation is implemented in Nitti et al. [51] by means of a distributed hash table structure, while Gu et al. [28], Namal et al. [50], Saied et al. [57], and Wang et al. [63] simply assumed the existence of a trusted central entity. No work thus far discusses the potential of integrating cloud computing with centralized trust propagation. The challenge lies in the cloud-based infrastructure design that facilitates trust information propagation from IoT devices to the cloud which aggregates trust feedbacks and answers user queries regarding the service trustworthiness of any particular IoT device in the system. This is a research gap that demands attention.

Trust Aggregation: Static weighted sum and Bayesian inference are the two most visited methods, dynamic weighted sum and fuzzy logic are least visited, while belief theory and regression analysis have not been investigated in the literature. There is no clear reason why fuzzy logic, belief theory, or regression analysis based trust aggregation cannot perform comparably or even better than weighted sum or Bayesian inference based trust aggregation. A comparative analysis is called for to compare these competitive trust aggregation methods for IoT systems. The comparative analysis should consider specific IoT applications and/or specific trust propagation models (distributed vs. centralized). For example, complex statistical analysis is required for regression analysis and hence it is particularly appealing when a centralized cloud is available for cloud-based IoT applications. This is a research gap that deserves attention.

Trust Update: Event-driven trust update is frequently performed when a service or transaction is completed, or a node encountering another node, while time-driven trust update occurs periodically to preserve energy. The hypothesis is that event-driven is more suitable under centralized trust propagation since the centralized entity (i.e., a cloud) is powerful, while time-driven is more suitable under distributed trust propagation since most, if not all, IoT devices are power limited. For the latter case, there is a tradeoff between trust accuracy and energy conservation. Hence, there exists an optimal trust update interval under which an IoT application performance is maximized depending on the trust accuracy and energy consumption requirements which collectively determine the success or failure of the IoT application. These hypotheses remain to be verified or refuted.

Trust Formation: there is a big gap from most visited methods in single-trust and multi-trust with static weighted sum, to little visited methods in multi-trust with dynamic weighted sum, multi-trust each with its own minimum threshold, or multi-trust with trust scaled by confidence. Multi-trust refers to the trust protocol that considers more than one trust properties, each being assessed separately and a trust formation method is applied to form the overall trust out of these multiple trust properties. IoT systems inherently are multi-trust based because both QoS trust (for service quality) and social trust (for social relationship) must be considered, taking

into considerations that device-to-device interaction behaviors derive from owner-to-owner relationships. We see from Table 2 that only Bao et al. [3], Chen et al. [8], Chen et al.[17], and Nitti et al. [51] considered multi-trust with static weighted sum. There is a pressing need to investigate more effective multi-trust formation methods and a comprehensive comparative analysis to identify the best trust formation technique to maximize IoT application performance.

6 CONCLUSION AND RESEARCH DIRECTIONS

In this paper we developed a classification tree based on five design dimensions considered essential for trust computation, namely, trust composition, trust propagation, trust aggregation, trust update, and trust formation, to classify existing IoT trust computation models. We discussed the pros and cons of existing IoT trust computation models in terms of the class they fall within. In particular for each class of IoT trust computation models, we discussed the defense mechanisms used and their effectiveness against malicious attacks aiming to disrupt the trust system.

Through the pros and cons analysis of IoT trust computation model classes, we identified the most effective trust computation techniques (and thus trust computation classes) as applying to IoT systems. We further identified research gaps in IoT trust computation research for supporting future IoT applications.

We offer eight directions for trust computation research in IoT systems as follows:

1. The first and foremost is to explore untouched trust aggregation techniques based on belief theory or regression analysis. Regression analysis especially is applicable when IoT nodes can access a centralized trust manager, say, located on a cloud because of the limited computing power of IoT devices for executing computationally expensive statistical analysis. Feedback data including service context information such as capability and energy of the SP, the traffic congestion condition of the network, and the service quality feedback itself can be propagated to the cloud for complex statistical analysis to better connect context information with service quality, and thus provide a more accuracy estimate of service quality trust of a SP in question.
2. The second research direction is to further explore innovative social trust metrics and the best way to combine them for IoT trust computation. The reason is that IoT systems are inherently social oriented. In the literature, using social similarity to rate a trustee or a recommender is emerging [12]. However, how to combine several social metrics such as friendship, social contact, and community-of-interest, into social similarity is still an open problem. Further, other than similarity, there are also many IoT social properties such as centrality, selfishness, cooperativeness and honesty that need to be further explored. Properly exploring social relationship between IoT devices can effectively defend bad-mouthing and ballot-stuffing attacks which happen commonly in social IoT systems.
3. The third research direction is to devise and validate a trust computation model that can defend against all

attacks. Existing works have considered ways to defend against self-promotion attacks (SPA), bad-mouthing attacks (BMA) and ballot-stuffing attacks (BSA) effectively but not opportunistic service attacks (OSA) and on-off attacks (OOA). Properly leveraging social trust may be an effective way to defend against bad-mouthing attacks and ballot-stuffing attacks, but this needs to be verified with real service-oriented social IoT systems. Also, there is a need to further model malicious behaviors (e.g., [18] [19] [48] [49]) which can happen in service-oriented social IoT systems, and devise as well as validate effective defense mechanisms against such malicious behaviors.

4. There is a big gap in the area of trust formation when there are several distinct trust metrics and one wants to combine several trust metrics into one overall trust metric. The literature is thin in this area [3] [8] [17] [51] considering only the use of static weighted sum for trust formation. The fourth research direction is to investigate the use of more effective trust formation methods. Potential methods to investigate further include (a) dynamic weighted sum, (b) setting a minimum threshold for each trust property without combining multiple properties into one, and (c) using one trust property as the main one which is scaled by other trust properties serving as confidence. In particular, dynamic weighted sum potentially can improve application performance when the weights associated multiple trust properties can be dynamically adjusted based on environment context information available at runtime. A potential technique to use for dynamic weighted sum is regression analysis [64] to link context information with trust accuracy and/or application performance so as to determine the best weight assignment. Another potential technique is adaptive filtering [12] to follow the principle that a node should have high trust toward IoT devices to which it has more positive user satisfaction experiences and, conversely, low trust toward those with more negative user satisfaction experiences.
5. The fifth research direction which is an open problem is to design a trust computation method that can scale. A modern IoT system comprises not just thousands but millions or even billions of heterogeneous IoT devices. Performance analysis of scalable trust propagation and trust storage methods is of paramount importance. For distributed trust computation, a push-based periodic trust propagation method will not scale. A more scalable pull-based trust propagation method is called for. A possible technique is encounter-based trust propagation [14] such that trust information is exchanged only when IoT devices encounter with each other, so there is no extra traffic created or energy spent for forwarding recommendation packets. Also many IoT devices will be tiny so it is impractical for each IoT device to store the trust values of all other IoT devices in the system just for trust-based decision making. Therefore a scalable storage scheme based on heuristic design principles is called for. One possibility is to store trust information for nodes with the highest trust values and nodes recently interacted or encountered, as these nodes are most likely to share common interests [4]. For centralized trust computation, a single cloud server will not scale. One research direction is to leverage tiered cloud architectures for scalable hierarchical trust management [5] [9] [11] [13]. A two-tier cloud architecture proposed by Rahimil et al. [52] consists of local and public clouds, with public clouds sitting at the high level

of the hierarchy and local clouds sitting at the low level. Utilizing this 2-tier cloud architecture, an IoT device roams from one local cloud server area into another local cloud server area, thus triggering a “service handoff” similar to a “mobility handoff” in hierarchical mobility management [15] [16] [27] [38]. A service handoff causes trust information to be transferred from one cloud server to another. The two-tier cloud architecture can be expanded into n -tier as necessary to scale to millions or even billions of IoT devices.

6. The sixth research direction is to integrate cloud service with trust management service, aka, trust as a service, for centralized trust management of an IoT community. The IoT community for example can be an e-health group paying particular attention to air pollution for the welfare of a group of users who may suffer from polluted air quality, an intelligent your-ride-on-demand IoT group (like Uber), or a smart city group consisting of visitors, merchants, restaurants, and entertainment business entities, etc. Trust as a service would be a perfect service provided by the cloud to members in each of these groups. Service reputation feedbacks along with service context information can be fed into the cloud for a complex yet complete statistical analysis. Users requesting a service or a composite service (i.e., several services bundled together via service composition and binding) can be assured of high-quality service as a result of “trust as a service” being applied to such a service-oriented IoT group.
7. The 7th research area is to expand and apply IoT trust management techniques to real-world IoT applications. See Atzori et al. [2] and Borgia et al. [6] for a list of emerging IoT applications. To date, only Chen et al. [8] tested the applicability on real-world IoT applications which require dynamic service composition and binding [21] [39]. It is of paramount importance to identify IoT application to which a trust computation technique can be feasibly applied for security enhancement, especially for those IoT applications that rely on cloud computing technology.
8. Lastly, there is a lack of a holistic design for scalable, adaptive and survivable trust computation for social IoT systems. Among the trust computational techniques to-date, only Bao et al. [4] and Chen et al. [12] partially addressed this issue in the context of distributed trust computation. Hence, the 8th research direction is to consider a more holistic design to manage “integrated” mobility, service and trust information of a large number of IoT devices in a scalable, secure, reliable, and efficient manner, possibly by integrating the design concepts currently existing in hierarchical trust management [5] [9] [11] [13], hierarchical mobility management [15] [16] [27] [38], and tiered cloud architectures [52]. While a node in a hierarchical mobility management architecture is a router responsible for keeping track of location information only (where and how to route), a node in a hierarchical cloud management architecture can be a cloud server responsible for keeping track of “integrated” information including location, trust, and service information. A lower-level cloud server (e.g., a cloudlet or a private cloud) keeps track of IoT devices in its directly covered service area. A higher level cloud server (e.g., a public cloud) in the architecture keeps track of status of all IoT devices covered by all local cloud servers below it. Should an IoT device roam from one cloud service area to another,

a “service handoff” ensues causing this IoT device’s location, trust and service information to be transferred between the two involving cloud servers. Such an IoT framework can track IoT devices not only in trust status, but also in service and mobility status dynamically to achieve the potential of anytime anywhere service-oriented IoT applications in the 21th century.

ACKNOWLEDGEMENT

This work is supported in part by the U.S. Army Research Office under contract number W911NF-12-1-0445.

REFERENCES

- [1] S. Adali et al., "Measuring Behavioral Trust in Social Networks," *IEEE International Conference on Intelligence and Security Informatics*, Vancouver, BC, Canada, May 2010.
- [2] 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, Nov. 2012, pp. 3594-3608.
- [3] F. Bao and I.R. Chen, "Dynamic Trust Management for the Internet of Things Applications," *International Workshop on Self-Aware Internet of Things*, San Jose, USA, Sept., 2012.
- [4] F. Bao, I.R. Chen and J. Guo, "Scalable, Adaptive and Survivable Trust Management for Community of Interest Based Internet of Things Systems," *11th International Symposium on Autonomous Decentralized System*, Mexico City, Mexico, March 2013
- [5] F. Bao, I. R. Chen, M. Chang, and J. H. Cho, "Hierarchical Trust Management for Wireless Sensor Networks and Its Applications to Trust-Based Routing and Intrusion Detection," *IEEE Trans. on Network and Service Management*, vol. 9, no. 2, 2012, pp. 161-183.
- [6] E. Borgia, "The Internet of Things vision: Key features, applications and open issues," *Computer Communications*, vol. 54, 2014, pp. 1-31.
- [7] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang "TRM-IoT: a trust management model based on fuzzy reputation for Internet of Things." *Computer Science and Information Systems*, vol. 8, no. 4, 2011, pp. 1207-1228.
- [8] 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*, 2016.
- [9] I. R. Chen, F. Bao, M. Chang, and J.H. Cho, "Trust-based intrusion detection in wireless sensor networks," *IEEE International Conference on Communications*, Kyoto, Japan, June 2011, pp. 1-6.
- [10] I.R. Chen, et al., "Dynamic Trust Management for Delay Tolerant Networks and its Application to Secure Routing," *IEEE Trans. Parallel and Distributed Systems*, vol. 25, no. 5, 2014, pp. 1200-1210.
- [11] I.R. Chen, and J. Guo, "Hierarchical Trust Management of Community of Interest Groups in Mobile Ad Hoc Networks," *Ad hoc Networks*, vol. 33, 2015, pp. 154-167.
- [12] I.R. Chen, J. Guo, and F. Bao, "Trust Management for SOA-based IoT and Its Application to Service Composition," *IEEE Transactions on Service Computing*, vol. 9, no. 3, 2016, pp. 482-495.
- [13] I.R. Chen and J. Guo, "Dynamic Hierarchical Trust Management of Mobile Groups and Its Application to Misbehaving Node Detection," *28th IEEE International Conference on. Advanced Information Networking and Applications*, Victoria, Canada, May 2014, pp. 1-6.
- [14] I.R. Chen, F. Bao, M. Chang, J.H. Cho, "Trust management for encounter-based routing in delay tolerant networks," *Global Telecommunications Conference*, Miami, USA, 2010, pp. 1-6.
- [15] I.R. Chen, T.M. Chen, and C. Lee, "Performance evaluation of forwarding strategies for location management in mobile networks," *The Computer Journal*, vol. 41, no. 4, 1998, pp. 243-253.
- [16] I.R. Chen, and N. Verma, "Simulation study of a class of autonomous host-centric mobility prediction algorithms for wireless cellular and ad hoc networks," *36th Annual Symposium on Simulation*, 2003, pp. 65-72.
- [17] Z. Chen, R. Ling, C.M. Huang and X. Zhu, "A scheme of access service recommendation for the Social Internet of Things," *International Journal of Communication Systems*, vol. 29, no. 4, 2016, pp. 694-706.
- [18] J.H. Cho, I.R. Chen, and P. Feng "Effect of Intrusion Detection on Reliability of Mission-Oriented Mobile Group Systems in Mobile Ad Hoc Networks," *IEEE Trans. on Reliability*, vol. 59, 2010, pp. 231-241.

- [19] J. H. Cho, A. Swami, and I. R. Chen, "Modeling and Analysis of Trust Management for Cognitive Mission-Driven Group Communication Systems in Mobile Ad Hoc Networks," *International Conference on Computational Science and Engineering*, vol. 2, 2009, pp. 641-650.
- [20] J. H. Cho, A. Swami, and I. R. Chen, "A survey on trust management for mobile ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 4, 2011, pp. 562-583.
- [21] K. Dar, A. Taherkordi, R. Rouvoy, and F. Eliassen, "Adaptable Service Composition for Very-Large-Scale Internet of Things Systems," *ACM Middleware*, Lisbon, Portugal, Dec. 2011
- [22] A.P. Dempster, "A generalization of Bayesian inference," *Journal of the Royal Statistical Society, Series B*, Vol. 30, pp. 205–247, 1968.
- [23] T. Dubois, J. Golbeck, and A. Srinivasan, "Predicting Trust and Distrust in Social Networks," *IEEE 3rd International Conference on Social Computing*, Boston, MA, USA, Oct. 2011, pp. 418-424..
- [24] L.C. Freeman, Centrality on social networks, *Social Networks*, vol. 1, 1979, pp. 215– 239.
- [25] S. Ganerwal, L.K. Balzano, and M.B. Srivastava, "Reputation-Based Framework for High Integrity Sensor Networks," *ACM Trans. Sensor Networks*, vol. 4, no. 3, 2008, pp. 1-37.
- [26] J. Granjal, E. Monteiro, and J. Silva, "Security for the internet of things: A survey of existing protocols and open research issues," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, 2015, 1294 - 1312.
- [27] B. Gu, and I. R. Chen, "Performance analysis of location-aware mobile service proxies for reducing network cost in personal communication systems," *Mobile Networks and Applications*, vol. 10, no. 4, 2005, pp. 453-463.
- [28] L. Gu, J. Wang, B.B. Sun, "Trust management mechanism for internet of things," *China Commun.* vol. 11, issue 2, pp. 148–156, 2014.
- [29] C.W. Hang and M.P. Singh, "Trustworthy Service Selection and Composition," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 6, no. 1, article 5, February 2011.
- [30] Hoffman, K., Zage, D., & Nita-Rotaru, C. (2009). "A survey of attack and defense techniques for reputation systems," *ACM Computing Surveys*, vol.42, issue 1, pp.1-31, 2009.
- [31] A. Jøsang, "A logic for uncertain probabilities," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, no. 3, June 2001, pp. 279– 311.
- [32] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vo. 43, no. 2, 2007, pp. 618-644.
- [33] A. Jøsang et al., "The Beta Reputation System," *Proc. 15th Bled Electronic Commerce Conf.*, 2002, pp. 1-14.
- [34] R. Lacuesta, G. Palacios-Navarro, C. Cetina, L. Penalver, J. Lloret, "Internet of things: where to be is to trust," *EURASIP Journal on Wireless Communications and Networking*, no. 1, 2012, pp. 203.
- [35] F. Li, and J. Wu, "Uncertainty Modeling and Reduction in MANETs," *IEEE Transactions on Mobile Computing*, vol. 9, no. 7, pp.1035-1048, 2010.
- [36] Li, M., Wang, H., & Ross, D. (2009). "Trust-Based Access Control for Privacy Protection in Collaborative Environment," *IEEE International Conference on E-Business Engineering*, pp. 425-430, 2009.
- [37] W. Li, and A. Joshi, "Coping with Node Misbehaviors in Ad Hoc Networks: A Multi-dimensional Trust Management Approach," *2010 Eleventh International Conference on Mobile Data Management*, Kansas City, USA, 2010, pp. 85-94.
- [38] Y. Li, and I.R. Chen, "Design and performance analysis of mobility management schemes based on pointer forwarding for wireless mesh networks," *IEEE Trans Mobile Computing*, vol. 10, no. 3, 2011, pp. 349-361.
- [39] L. Liu, X. Liu, and X. Li, "Cloud-Based Service Composition Architecture for Internet of Things," *International Workshop on Internet of Things*, Changsha, China, August 2012, pp. 559-564.
- [40] T. Liu, Y. Guan, Y. Yan, L. Liu, Q. Deng, "A WSN-oriented key agreement protocol in internet of things," *Applied Mechanics and Materials*, Vol. 401-403, pp. 1792-1795, 2013.
- [41] Y. Liu, Z. Chen, F. Xia, X. Lv, and F. Bu, "An integrated scheme based on service classification in pervasive mobile services," *International Journal of Communication Systems*. vol. 25 issue 9, pp. 1178–1188, 2012.
- [42] Y. Liu, Z. Chen, F. Xia, X. Lv, and F. Bu, "A trust model based on service classification in mobile services," *IEEE/ACM international conference on cyber, physical and social computing*, 2010, pp. 572–577.
- [43] P.N. Mahalle, P.A. Thakre, N.R. Prasad, and R. Prasad, "A fuzzy approach to trust based access control in internet of things," *3rd International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems, VITAE*, NJ, Atlantic City, 2013, pp. 1–5.
- [44] Zaki Malik and Athman Bouguettaya, "RATEWeb: Reputation Assessment for Trust Establishment among Web services," *The*

VLDB Journal, vol. 18, 2009, pp. 885-911.

- [45] D.W. Manchala, "Trust metrics, models and protocols for electronic commerce transactions," *18th IEEE International Conference on Distributed Computing Systems*, 1998.
- [46] P. Martinez-Julia, A.F. Skarmeta, "Beyond the separation of identifier and locator: building an identity-based overlay network architecture for the future internet," *Computer Networks*, vol. 57, issue 10, pp. 2280–2300, 2013.
- [47] C.V.L. Mendoza and J.H. Mendoza, "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, 2015, article 859731.
- [48] R. Mitchell and I.R. Chen, "Modeling and Analysis of Attacks and Counter Defense Mechanisms for Cyber Physical Systems," *IEEE Trans. on Reliability*, vol. 65, no. 1, March 2016, pp. 350-358.
- [49] R. Mitchell and I.R. Chen, "Effect of Intrusion Detection and Response on Reliability of Cyber Physical Systems," *IEEE Transactions on Reliability*, vol. 62, no. 1, March 2013, pp. 199-210.
- [50] S. Namal, J. Granjal, E. Monteiro, and J. Silva, "Autonomic trust management in cloud-based and highly dynamic IoT applications," *ITU Kaleidoscope: Trust in the Information Society (K-2015)*, Barcelona, Dec. 2015.
- [51] M. Nitti, R. Girau, and L. Atzori, "Trustworthiness Management in the Social Internet of Things," *IEEE Transactions on Knowledge and Data Management*, vol. 26, no. 5, 2014, pp. 1253-1266.
- [52] M.R. Rahimi¹, N. Venkatasubramanian¹, S. Mehrotra¹, and A.V. Vasilakos, "MAPCloud: Mobile Applications on an Elastic and Scalable 2-Tier Cloud Architecture," *IEEE/ACM Fifth International Conference on Utility and Cloud Computing*, 2012, pp. 83-90.
- [53] M. Raya, P. Papadimitratos, V.D. Gligory, and J.P. Hubaux, "On Data-Centric Trust Establishment in Ephemeral Ad Hoc Networks," *IEEE 27th Conference on Computer Communications*, April 2008.
- [54] S. Ries, "CertainTrust: A trust model for users and agents," *Proceedings of the 2007 ACM Symposium on Applied Computing*, 2007, pp. 1599-1604.
- [55] S. Ries, S.M. Habib, M. Mühlhäuser, and V. Varadharajan, "CertainLogic: A Logic for Modeling Trust and Uncertainty" *TRUST 2011*, Pittsburgh, PA, USA, June, 2011, pp. 254-261.
- [56] J. Sabater, and C. Sierra, "REGRET: a reputation model for gregarious societies," *4th Int. Workshop on Deception, Fraud and Trust in Agent Societies*, Montreal, Canada, 2001, pp. 61– 70.
- [57] Y.B. Saied, A. Olivereau, D. Zeghlache, and M. Laurent, "Trust management system design for the Internet of Things: A context-aware and multi-service approach," *Computers and Security*, vol. 39, 2013, pp. 351–365.
- [58] W. Sherchan, S. Nepal, and C. Paris, "A Survey of Trust in Social Networks," *ACM Computing Survey*, Vol. 45, No. 4, Article 47, August 2013.
- [59] S. Sicaria, A. Rizzardia, L.A. Griecob, and A. Coen-Portisini, "Security, privacy and trust in Internet of Things: The road ahead," *Computer Networks*, vol. 76, Jan. 2015, pp. 146-164.
- [60] G.D. Tormo, F.G. Marmol, G.M. Perez, "Dynamic and flexible selection of a reputation mechanism for heterogeneous environments," *Future Generation Computer Systems*, vol. 49, pp. 113-124, August 2015.
- [61] P.B. Velloso, et al., "Trust Management in Mobile Ad Hoc Networks Using a Scalable Maturity-Based Model," *IEEE Transactions on Network and Service Management*, vol. 7, no. 3, 2010, pp. 172-185.
- [62] K. Wan, and V. Alagar, "Integrating Context-Awareness and Trustworthiness in IoT Descriptions," *IEEE International Conference on Internet of Things*, Aug. 2013, Peking, China, pp. 1168-1174.
- [63] J. Wang, S. Bin, Y. Yu, X. Niu, "Distributed trust management mechanism for the internet of things," *Applied Mechanics and Materials*, Vol. 347-350, pp. 2463-2467, 2013.
- [64] Y. Wang, Y.C. Lu, I.R. Chen, J.H. Cho, and A. Swami, "LogitTrust: A Logit Regression-based Trust Model for Mobile Ad Hoc Networks," *6th ASE International Conference on Privacy, Security, Risk and Trust*, Boston, MA, Dec. 2014.
- [65] Y. Wang, I.R. Chen, J.H. Cho, A. Swami, and K. Chan "Trust-based Service Composition and Binding with Multiple Objective Optimization in Service-Oriented Mobile Ad Hoc Networks," *IEEE Transactions on Services Computing*, 2016, in press, DOI: 10.1109/TSC.2015.2491285.
- [66] 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, june 2014, pp. 120-134.
- [67] B. Yu, and M.P. Singh, "An evidential model of distributed reputation management," *1st ACM Int. Joint Conference on Autonomous Agents and Multiagent Systems*, July 2002.
- [68] H. Yu, Z. Shen, C. Miao, C. Leung, and D. Niyato, "A Survey of Trust and Reputation Management Systems in Wireless Communications," *Proceedings of the IEEE*, vol. 98, no. 10, pp. 1755-1772, 2010.