Dynamic Trust Management for Internet of Things Applications

Fenye Bao and Ing-Ray Chen
Department of Computer Science, Vriginia Tech
{baofenye, irchen}@vt.edu

ABSTRACT

We propose a *dynamic trust management* protocol for Internet of Things (IoT) systems to deal with misbehaving nodes whose status or behavior may change dynamically. We consider an IoT system being deployed in a smart community where each node autonomously performs trust evaluation. We provide a formal treatment of the convergence, accuracy, and resilience properties of our dynamic trust management protocol and validate these desirable properties through simulation. We demonstrate the effectiveness of our dynamic trust management protocol with a trust-based service composition application in IoT environments. Our results indicate that trust-based service composition significantly outperforms non-trust-based service composition and approaches the maximum achievable performance based on ground truth status. Furthermore, our dynamic trust management protocol is capable of adaptively adjusting the best trust parameter setting in response to dynamically changing environments to maximize application performance.

Keywords

Dynamic trust management, Internet of things, social networking, performance analysis, adaptive control, security.

1. INTRODUCTION

The emerging paradigm of the Internet of Things (IoT) builds upon the ubiquitous connectivity of smart objects with wide applicability [1, 6, 9]. One important characteristic of IoT is that most smart objects are human-carried or human-related heterogeneous devices. Therefore, the social relationships among the device users must be taken into consideration during the design phase of IoT applications. Atzori *et al*. [2] introduced the notion of Social Internet of Things (SIoT) and analyzed various types of social relationships, like parental object relationship, co-location and co-work relationship, ownership, etc., among objects. Further, devices in IoT very often expose to public areas and communicate through wireless [13]. Hence, IoT objects are vulnerable to malicious attacks [11]. In this paper, we propose a *dynamic trust management* protocol for IoT systems considering both malicious and socially uncooperative nodes, with the goal to enhance the security and increase the performance of IoT applications. Our aim is to design and validate a dynamic trust management protocol that can dynamically adjust trust design parameter settings in response to changing environment conditions to maximize application performance.

Self-IoT'12, September 17, 2012, San Jose, California, USA.

Security management in IoT environments is of fundamental importance [4, 10, 11, 14]. However, to the best of our knowledge there is little work on trust management in IoT environments for security enhancement, especially for dealing with misbehaving nodes who are legit members of an IoT community. Chen *et al*. [5] proposed a trust management model based on fuzzy reputation for IoT. However, their trust management model considers a specific IoT environment consisting of only wireless sensors with QoS trust metrics like packet forwarding/delivery ratio and energy consumption. Very recently Bao and Chen [3] proposed a trust management protocol considering both social trust and QoS trust metrics and using both direct observations and indirect recommendations to update trust. However they only considered static environments (e.g., a static population of malicious nodes) and thus the results are not easily applicable to IoT environments in which environment conditions are evolving, e.g., increasing misbehaving node population/activity, behavior change, rapid membership change, and rapid interaction pattern change.

This work extends from [3]. The new contributions in this paper are as follows. First, we develop a dynamic trust management protocol for IoT systems to deal with misbehaving nodes whose status or behavior may evolve or change dynamically. Second, we provide a formal treatment of the convergence, accuracy, and resilience properties of our dynamic trust management protocol and validate these desirable properties through simulation. The trust protocol developed in this paper is adaptive to changing environment conditions. Lastly, using service composition as an IoT application, we demonstrate that our dynamic trust management protocol is capable of adaptively adjusting the best trust parameter setting in response to dynamically changing environments to maximize application performance.

The rest of this paper is organized as follows. Section 2 describes our IoT system model. In Section 3, we present the detail of our dynamic trust management protocol and formally analyze the convergence, accuracy, and resiliency properties of the protocol. Section 4 provides simulation results for validating these desirable properties. Further, using a trust-based service composition application running on top of our dynamic trust management protocol, we demonstrate the utility of dynamic trust management. Finally, Section 5 concludes the paper and outlines future work.

2. SYSTEM MODEL

Our IoT system model follows that of [3]. Figure 1 illustrates the system model for an IoT system with socially interacting entities. We consider an IoT environment with no centralized trusted authority. Each node is able to autonomously and independently interact with others, perform computation, and store information. Every device (node) has an owner and an owner could have many

devices. Each owner has a list of friends, representing its social relationships. A device is carried or operated by its owner in certain communities or working environments. Nodes belonging to a similar set of communities likely have similar interests or similar capabilities. We differentiate uncooperative nodes from malicious nodes. An uncooperative node acts for its own interest. So it may stop providing service to a service requester if it does not have a strong social tie (e.g., friendship) with the service requester. A malicious node aims to break the basic functionality of the IoT. In addition, it can perform the following trust-related attacks:

1. Self-promoting attacks: it can promote its importance (by providing good recommendations for itself) so as to be selected as the service provider, but then stop providing service or provide malfunction service.
2. Bad-mouthing attacks: it can ruin the reputation of well-behaved nodes (by providing bad recommendations against good nodes) so as to decrease the chance of good nodes being selected as service providers.
3. Good-mouthing attacks; it can boost the reputation of bad nodes (by providing good recommendations for them) so as to increase the chance of bad nodes being selected as service providers.

A node's trust value is assessed based on direct observations and indirect information like recommendations. The trust of one node toward another node is updated upon encounter and interaction events. Each node will execute the trust protocol independently and will perform its direct trust assessment toward an encountered node based on specific detection mechanisms designed for assessing a trust property. Later we will discuss these specific detection mechanisms employed in our protocol.

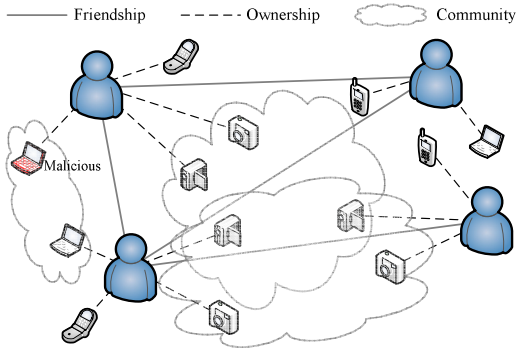

Figure 1: System Model for an IoT System.

3. DYNAMIC TRUST MANAGEMENT

3.1 Protocol Description

Our trust management protocol for IoT is distributed. Each node maintains its own trust assessment towards other nodes. For scalability, a node may just keep its trust evaluation towards a limited set of nodes which it is most interested in. 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 our trust management protocol, a node maintains multiple trust properties in *honesty*, *cooperativeness*, and *community-interest*. The *honesty* trust property represents whether or not a node is honest. The *cooperativeness* trust property represents whether or not the trustee is socially cooperative [8] with the trustor. The *community-interest* trust represents whether or not the trustor and trustee are in the same social communities/groups (e.g. co-location or co-work relationship [2]) or have the similar capabilities (parental object relationship [2]). The trust assessment of node i towards node j at time t is denoted by $T_{ij}^{X}(t)$ where X = *honesty*, *cooperativeness*, or *community-interest*. The trust value $T_{ij}^{X}(t)$ is a real number in the range of [0, 1] where 1 indicates complete trust, 0.5 ignorance, and 0 distrust. When node i encounters or directly interacts with another node k at time t, node i will update its trust assessment $T_{ij}^{X}(t)$ as follows:

$$T_{ij}^{X}(t) = \begin{cases} (1-\alpha)T_{ij}^{X}(t-\Delta t) + \alpha D_{ij}^{X}(t) & if\ j == k \\ (1-\gamma)T_{ij}^{X}(t-\Delta t) + \gamma R_{kj}^{X}(t) & if\ j\ != k \end{cases} \quad (1)$$

Here, Δt is the elapsed time since the last trust update. If the trustee node j is node k itself, node i will use its new trust assessment toward node j based on direct observations ($D_{ij}^{X}(t)$) and its old trust toward node j based on past experiences to update $T_{ij}^{X}(t)$. A parameter α ($0 \leq \alpha \leq 1$) is used here to weigh these two trust values and to consider trust decay over time, i.e., the decay of the old trust value and the contribution of the new trust value. A larger α means that trust evaluation will rely more on direct observations. Here $D_{ij}^{X}(t)$ indicates node i's trust value toward node j based on direct observations accumulated over the time period $[0, t]$. Below we describe how each trust component value $D_{ij}^{X}(t)$ can be obtained based on direct observations for the case in which node i and node j interacting or encountering each other within radio range.

3.1.1 Honesty

$D_{ij}^{honesty}(t)$: This trust property refers to the belief of node i that node j is honest based on node i's direct observations toward node j. Node i estimates $D_{ij}^{honesty}(t)$ by keeping a count of suspicious dishonest experiences of node j which node i has observed during $[0, t]$ using a set of anomaly detection rules such as a high discrepancy in recommendation has been experienced, as well as interval, retransmission, repetition, and delay rules as in [7, 12]. If the count exceeds a system-defined threshold, node j is considered totally dishonest at time t, i.e., $D_{ij}^{honesty}(t) = 0$. Otherwise, $D_{ij}^{honesty}(t)$ is computed by 1 minus the ratio of the count to the threshold. Our hypothesis is that a compromised node must be dishonest. We consider non-zero false positive probability (P_{fp}) and false negative probability (P_{fn}) for such detection mechanism. The system-defined threshold is a design parameter which leads the to the calculation of P_{fp} and P_{fn}.

3.1.2 Cooperativeness

$D_{ij}^{cooperativeness}(t)$: This trust property provides the degree of cooperativeness of node j as evaluated by node i based on direct observations over $[0, t]$. We use the social friendship [8] relationship among device owners to characterize the cooperativeness. Our hypothesis is that friends are likely to be cooperative toward each other. The cooperativeness trust of node i towards node j is computed as the ratio of the number of common friends over the total number of nodes i's and j's friends, i.e., $\frac{|friends(i) \cap friends(j)|}{|friends(i) \cup friends(j)|}$, where $friends(i)$ denotes the set of node i's friends. A node is included in its own friend list (i.e., $i \in friends(i)$) to deal with the case where two nodes are the only friends to each other. When node i and node j encounter and

directly interact with each other, they can exchange their friend lists. Node *i* can validate a friend in node *j*'s list if it is their common friend. Therefore, the direct observation of cooperativeness will be close to actual status.

3.1.3 Community-Interest

$D_{ij}^{community-interest}(t)$: This trust property provides the degree of the common interest or similar capability of node *j* as evaluated by node *i* based on direct observations over $[0,t]$. The community-interest trust of node *i* towards node *j* is computed as the ratio of the number of common community/group interests over the total number of nodes *i*'s and *j*'s community/group interests, i.e., $\frac{|community(i) \cap community(j)|}{|community(i) \cup community(j)|}$, where $community(i)$ denotes the set of node *i*'s communities/groups. When node *i* and node *j* encounter and directly interact with each other, they can exchange their service and device profiles. Node *i* can validate whether node *j* and itself are in a particular community/group. Therefore, the direct observation of community-interest will be close to actual status.

3.1.4 Recommendation

On the other hand, in Equation 1, if node *j* is not node *k*, then node *i* will not have direct observation on node *j* and will use its past experience $T_{ij}^X(t-\Delta t)$ and recommendations from node *k* ($R_{kj}^X(t)$ where *k* is the recommender) to update $T_{ij}^X(t)$. The parameter γ is used here to weigh recommendations vs. past experiences and to consider trust decay over time as follows:

$$\gamma = \frac{\beta D_{ik}^X(t)}{1+\beta D_{ik}^X(t)} \tag{2}$$

Here we introduce another parameter $\beta \geq 0$ to specify the impact of "indirect recommendations" on $T_{ij}^X(t)$ such that the weight assigned to indirect recommendations is normalized to $\beta T_{ik}^X(t)$ relative to 1 assigned to past experiences. Essentially, the contribution of recommended trust increases proportionally as either $D_{ik}^X(t)$ or β increases. Instead of having a fixed weight ratio $D_{ik}^X(t)$ to 1 for the special case in which $\beta = 1$, we allow the weight ratio to be adjusted by adjusting the value of β and test its effect on protocol resiliency against slandering attacks such as good-mouthing and bad-mouthing attacks. Here, $D_{ik}^X(t)$ is node *i*'s trust toward node *k* as a recommender (for node *i* to judge if node *k* provides correct information). The recommendation $R_{kj}^X(t)$ provided by node *k* to node *i* about node *j* depends on if node *k* is a good node. If node *k* is a good node, $R_{kj}^X(t)$ is simply equal to $D_{kj}^X(t)$. If node *k* is a bad node, it can provide $R_{kj}^X(t) = 0$ when node *j* is a good node by means of bad-mouthing attacks, and can provide $R_{kj}^X(t) = 1$ when node *j* is a bad node by means of good-mouthing attacks. In our analysis we assume this worst-case attack behavior to test our protocol resiliency.

3.2 Convergence, Accuracy and Resiliency Analysis

We analyze the convergence, accuracy, and resiliency properties to trust attacks of our trust management. For ease of disposition, we omit the superscript *X = honesty, cooperativeness, or community-interest* in $T_{ij}^X(t)$, $D_{ij}^X(t)$, and $R_{ij}^X(t)$, since the analysis is generic and applicable to all three trust properties.

Suppose there are N_T nodes in the network. Let the ground truth status of node *j* at time *t* be denoted by $G_j(t) \in [0,1]$. The direct observation on node *j* observed by node *i* at time *t* using the imperfect detection mechanism is modeled by a random variable $D_{ij}(t) \sim N(G_j(t), \sigma)$ bounded in the interval $[0,1]$, where σ is the standard deviation. At time *t*, node *i* updates its trust towards node *j*. Suppose that at time *t*, the probability that node *i* encounters or directly interacts with node *j* is p_0, and the probabilities that node *i* encounters or directly interacts with the other $N_T - 2$ nodes are $p_1,\ p_2,\ \ldots,\ p_{N_T-2}$, (such that $p_0 + p_1 + p_2 + \cdots + p_{N_T-2} = 1$). The expected trust evaluation of node *i* towards node *j* at time *t* therefore can be computed as:

$$\begin{aligned}
T_{ij}(t) &= p_0\left[(1-\alpha)T_{ij}(t-\Delta t) + \alpha D_{ij}(t)\right] \\
&\quad + \sum_{m=1}^{N_T-2} p_m\left[(1-\gamma_m)T_{ij}(t-\Delta t) + \gamma_m R_{k_m j}(t)\right] \\
&= \left[p_0(1-\alpha) + \sum_{m=1}^{N_T-2} p_m(1-\gamma_m)\right] T_{ij}(t-\Delta t) \\
&\quad + p_0\alpha D_{ij}(t) + \sum_{m=1}^{N_T-2} p_m\gamma_m R_{k_m j}(t) \\
&\triangleq q_0 T_{ij}(t-\Delta t) + \sum_{m=1}^{N_T-1} q_m R_{k_m j}(t) \\
&= q_0^n T_{ij}(0) + \sum_{l=0}^{n}\sum_{m=1}^{N_T-1} q_0^l q_m R_{k_m j}(t-l\Delta t)
\end{aligned} \tag{3}$$

where we have simplified the expression by using $q_0 \triangleq p_0(1-\alpha) + \sum_{m=1}^{N_T-2} p_m(1-\gamma_m)$; $q_m \triangleq p_m\gamma_m$; and $q_{N_T-1} \triangleq p_0\alpha$. Also $R_{k_{N_T-1}j}(t) \triangleq R_{ij}(t) = D_{ij}(t)$ considering node *i* as its own recommender. Note that we use the notation γ_m as the γ value used by node *i* towards node k_m based on Equation 2.

Lemma 1: *The trust evaluation in our protocol converges to the ground truth status as long as* $0 < \alpha \leq 1$ *or* $\beta > 0$.

Proof: In Equation 3, if $0 < \alpha \leq 1$ *or* $\beta > 0$, then $0 \leq q_0 < 1$. q_0^n will monotonically decrease to 0 as n or time t increases. Therefore, the expected value of $T_{ij}(t)$ (i.e. $E[T_{ij}(t)]$) will monotonically converge to $E\left[\sum_{l=0}^{n}\sum_{m=1}^{N_T-1} q_0^l q_m R_{k_m j}(t-l\Delta t)\right] = G_j(t)$ as n or time t increases until the ground truth status changes. In a dynamic environment, the convergence procedure repeats every time when the ground truth status changes. Specifically, if the ground truth status changes at time t, the trust evaluation at time t (i.e. $T_{ij}(t)$) will be set with a new initial value and the trust evaluation will converge towards the new ground truth status until the ground truth status changes again.

Lemma 2: *The trust convergence speed of our protocol increases as* α *or* β *increases*.

Proof: In Equation 3, as α or β increases, the value of q_0 decreases and the absolute value of slope of q_0^n (as a function of n or time t) increases. Using the conclusion of Lemma 1, we know that the convergence time is shorter as α or β increases.

Lemma 3: *The variance of the trust evaluation after converging in our protocol increases as* α *or* β *increases*.

Proof: In Equation 3, the variance of trust evaluation $T_{ij}(t)$ is $Var[T_{ij}(t)] = \sum_{l=0}^{n}\sum_{m=1}^{N_T-1}\left(q_0^l q_m\right)^2\sigma^2$. If n is sufficient large (i.e. $E[T_{ij}(t)]$ converges to $G_{ij}(t)$), $Var[T_{ij}(t)]$ increases as α or β increases.

Lemma 4: *If the percentage of malicious nodes is λ and the false negative and false positive probabilities of the detection mechanism are P_{fn} and P_{fp}, then the mean square error (MSE) of the trust evaluation in our protocol is less than $\frac{\lambda}{1-\lambda}\frac{P_{fn}}{1-P_{fp}}$ after trust converges. The MSE decreases as α increases or β decreases.*

Proof: We note that in Equation 3, after trust convergence, we have $E[T_{ij}(t)] = E[\sum_{l=0}^{n}\sum_{m=1}^{N_T-1} q_0^l q_m R_{k_{mj}}(t-l\Delta t)] \triangleq C\{\alpha G_j(t) + \frac{\beta}{1+\beta}[\lambda P_{fn} E[R_{kj}(t')] + (1-\lambda)(1-P_{fp})G_j(t)]\}$ where $t' \le t$ and C is constant to $R_{kj}(t')$ and $G_j(t)$. Thus, no matter what recommendations that malicious nodes provide, the MSE of trust evaluation against the ground truth is less than $\frac{\lambda P_{fn}}{(1-\lambda)(1-P_{fp})+\alpha\frac{1+\beta}{\beta}} \le \frac{\lambda}{1-\lambda}\frac{P_{fn}}{1-P_{fp}}$ and the MSE decreases as α increases or β decreases.

Note: In the proof for Lemma 4, we assume that probability of encountering or directly interacting with malicious nodes is the same as the probability of encountering or directly interacting with good nodes. Also we assume that the expected value of the direct observation is the ground truth. In reality, this may not be true (e.g., the ground truth trust value is 0 or 1). In this case, the trust evaluation converges to the expected value of the direct observation instead of the ground truth trust value. Lemmas 1-4 will still hold.

4. SIMULATION RESULTS

In this section, we give simulation results obtained as a result of executing our dynamic trust management protocol by IoT devices and demonstrate the effectiveness of our protocol with a service composition application in IoT environments.

Table 1: Default Parameter Values Used.

Param	Value	Param	Value	Param	Value
N_T	50	N_H	20	N_G	10
N_M	5	a	[0, 1]	β	[0, 1]
λ	[0, 90%]	P_{fp},P_{fn}	5%	T_S	100 hours

Table 1 lists the default parameter values. We consider an IoT environment with N_T = 50 heterogeneous smart objects/devices. These devices are randomly distributed to N_H = 20 owners. The social cooperativeness relationship among the devices is characterized by the friendship relationship (matrix) [8] among device owners, i.e., if the owners of devices i and j are friends, then there is a 1 in the ij position. Devices are used by their owners in one or more social communities or groups. A device can belong to up to N_G = 10 communities or groups. We consider a random waypoint mobility model where nodes move randomly and encounter or directly interact with each other when they are within the radio range. The total simulation time is 100 hours. We consider the hostility environment where the percentage of dishonest nodes $\lambda \in [0, 90\%]$ is randomly selected out of all devices. A normal or good node follows the execution of our trust management protocol, while a dishonest node acts maliciously by providing false trust recommendations (good-mouthing, bad-mouthing, and self-promoting attacks) to disrupt trust management. The initial trust value of all devices is set to ignorance with a trust level of 0.5.

Our design is for dynamic trust management. We first test our protocol with static environments, i.e., the ground truth status of a node does not change over time and then we test the protocol with dynamic environments, i.e., the ground truth status of a node changes over time.

4.1 Effect of α on Trust Evaluation

We first investigate the effect of design parameter α on trust evaluation. We vary the value of α by selecting five different values (0.1, 0.3, 0.5, 0.7, and 0.9) and fix the value of β to 0 to isolate its effect.

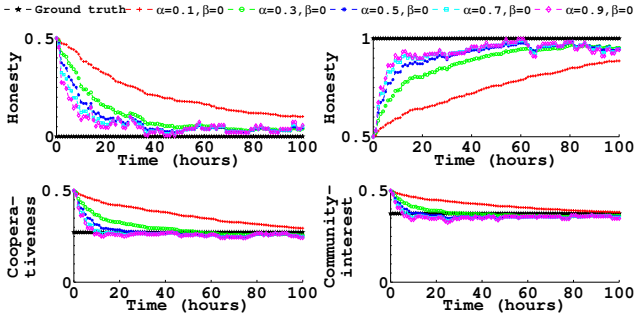

Figure 2: Effect of α on Trust Evaluation in Static Environment.

Figure 2 shows the effect of α on trust evaluation in static environments where the ground truth status does not change as time t increases. Figure 2 shows the trust evaluation for *dishonesty* (*ground truth trust* = 0), *honesty* (*ground truth trust* = 1), *cooperativeness*, and *community-interest* respectively. We can see that for all four cases, the trust evaluation approaches the ground truth status as time increases (Lemma 1). Further, we observe that as the value of α increases the trust evaluation converges to the ground truth faster (Lemma 2), but the trust fluctuation becomes higher (Lemma 3). The reason is that new direct observation can better reflect actual node status than past trust information. Using more new direct observation (higher α) in trust evaluation can help trust converge to the actual node status quickly.

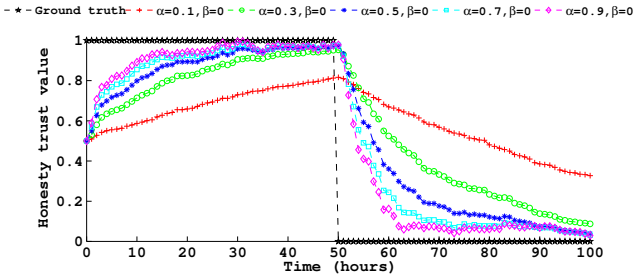

Figure 3: Effect of α on (Honesty) Trust Evaluation in Dynamic Environments.

To demonstrate the performance of our protocol in the dynamic environment, we consider a changing environment where a node initially is good, and then is compromised. Figure 3 shows the results of trust evaluation for *honesty* in this setting. We can see that after a status change, the trust evaluation converges towards the new ground truth status. In addition, as the value of α increases the trust evaluation converges to the new ground truth faster, albeit with a higher fluctuation. The results correlate well Lemmas 1-3 in dynamic environments.

4.2 Effect of β on Trust Evaluation

Next, we investigate the effect of design parameter β on trust evaluation. We fixed the value of α to 0.5 to isolate its effect and

vary the value of β by selecting five different values (0, 0.1, 0.2, 0.5, and 1).

Figure 4 shows the effect of β on trust evaluation in static environments for *dishonesty* (*ground truth trust* = 0), *honesty* (*ground truth trust* = 1), *cooperativeness*, and *community-interest* respectively. Again, we can see that for all four cases, our trust evaluation approaches ground truth status as time increases (Lemma 1). We also observe that as β increases, the trust evaluation converges to the ground truth faster (Lemma 2), but the trust fluctuation becomes higher (Lemma 3). The reason is that using more recommendations (higher β) helps trust convergence through effective trust propagation. However, one can see that the effect of β is insignificant compared to the effect of α as long as $\beta > 0$. The reason is that very often in an IoT environment with a large number of nodes, the chance of the trustor encountering a recommender is higher than the chance of the trustor directly interacting with a trustee. As long as $\beta > 0$, our protocol is able to effectively aggregate trust using recommendations from a large number of recommenders, thus making the effect of further increasing the value of β insignificant.

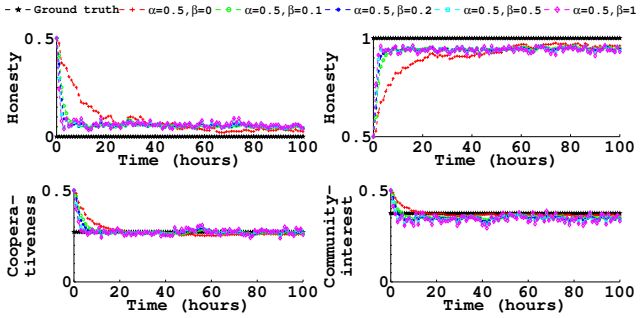

Figure 4: Effect of β on Trust Evaluation in Static Environment.

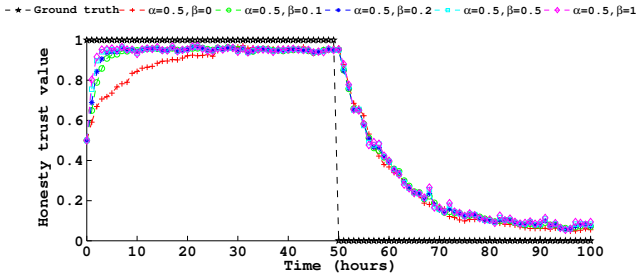

Figure 5: Effect of β on (Honesty) Trust Evaluation in Dynamic Environment.

Figure 5 shows the effect of β on (*honesty*) trust evaluation in dynamic environments. Again, we see that after the ground truth status changes, our trust protocol quickly converges towards the new ground truth status. Initially using recommendations ($\beta > 0$) in trust evaluation helps trust convergence. However, using recommendations does not contribute much to the trust convergence speed if the ground trust status changes dynamically. The reason behind this is that an honest recommender will provide obsolete and inaccurate trust recommendation, if it has not interacted with the trustee since the trustee's status changes. As the trustor will not exclude these inaccurate recommendations from good recommenders, it hinders trust convergence. One solution to this is to select a recommender only if this recommender has recently interacted with the trustee and use the statistical method to exclude recommendation outliers. We will explore this solution as one future research direction.

4.3 Protocol Resiliency to Trust Attacks

We further validate resiliency of our trust protocol to trust attacks. We choose the design parameters $(\alpha, \beta) = (0.5,0.5)$ and consider five different hostile environments with the percentage of malicious nodes being λ = 10%, 30%, 50%, 70%, and 90%. The malicious nodes are randomly selected and perform good-mouthing and bad-mouthing attacks.

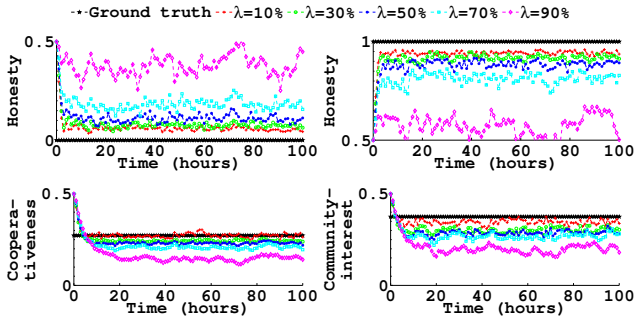

Figure 6: Effect of Hostility on Trust Evaluation.

Figure 6 shows trust evaluation results in static environments for *dishonesty* (*ground truth trust* = 0), *honesty* (*ground truth trust* = 1), *cooperativeness*, and *community-interest* respectively in the five hostile environments. One can see that the trust evaluation quickly converges and it is remarkably close to the ground truth status when $\lambda \leq 50\%$, demonstrating resiliency to trust attacks. As λ increases, the MSE of trust evaluation increases because of more false recommendations from malicious nodes. When $\lambda = 70\%$ and $\lambda = 90\%$, the MSE reaches 12% and 40%, as predicted by Lemma 4.

4.4 Dynamic Trust Management Application: Trust-Based Service Composition

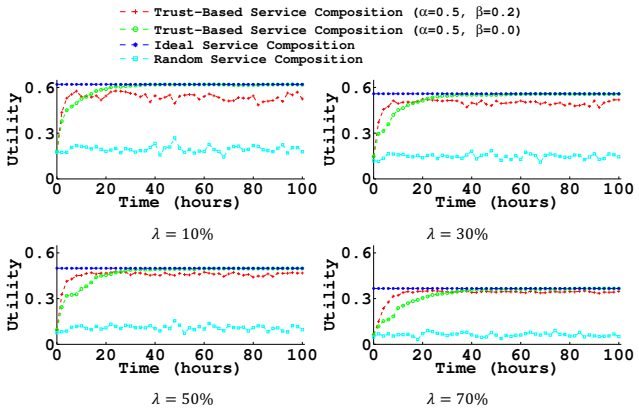

Figure 7: Performance Comparison on Service Composition.

Finally, to demonstrate the effectiveness of our dynamic trust management protocol, we consider a trust-based service composition application in IoT environments. In this application scenario, a node requests services (or information) from N_M service providers. The objective is to select the most trustworthy service providers such that the *utility* score representing the goodness of the service composition is maximized. Trust formation using the three trust components is application-specific. We consider the trust formation design that if a selected service provider is malicious, the returning utility score is zero; otherwise, the returning utility score equals to the smaller one of the *cooperativeness* trust value and *community-interest* trust value the

node has towards the service provider. In *trust-based service composition*, a node estimates the possible returning utility of each service provider based on its own knowledge and selects N_M service providers with the highest combined returning utility. The "actual" returning utility score is then computed based on actual status of the service providers selected. We compare the performance of our *trust-based service composition* with two baseline approaches, *ideal service composition* which returns the maximum achievable utility score derived from global knowledge, and *random service composition* in which a node randomly selects N_M service providers without regard to trust.

Figure 7 gives the results of the performance comparison in terms of utility score of our trust-based service composition against the two baseline service comparison methods. We consider two versions of our trust-based service composition by selecting two different sets of design parameters: $(\alpha, \beta) = (0.5, 0.2)$ and $(\alpha, \beta) = (0.5, 0)$. We see that as the percentage of malicious nodes increases, the utility score obtained by each protocol decreases because of fewer good service providers. We also observe that our trust-based service composition significantly outperforms random service composition and approaches the maximum achievable performance by ideal service composition. In addition, we see that there is a crossover point on the utility curves of two trust-based service composition methods. Before the crossover point, trust-based service composition under the setting of $(\alpha, \beta) = (0.5, 0.2)$ performs better, while after trust-based service composition under the setting of $(\alpha, \beta) = (0.5, 0)$ performs better. The reason is that while using recommendations helps trust quickly converge, it also introduces trust bias because of bad-mouthing and good-mouthing attacks. We observe that the crossover time point increases as the percentage of malicious nodes increases. Specifically, the crossover point is at t = 12 hours for λ = 10%, t = 18 hours for λ = 30%, t = 26 hours for λ = 50%, and t = 32 hours for λ = 70%. Thus, in a dynamic IoT environment in which the hostility (in terms of the percentage of malicious nodes) changes over time, dynamic trust management is achieved by choosing the best design parameter settings (α, β) to maximize the service composition application performance.

5. CONCLUSION

In this paper, we developed and analyzed a dynamic trust management protocol for IoT. The proposed protocol takes social relationships into account and advocates the use of three trust properties, *honesty*, *cooperativeness*, and *community-interest* to evaluate trust. The protocol is distributed and each node only updates trust towards others of its interest upon encounter or interaction events. The trust assessment is updated by both direct observations and indirect recommendations, with parameters α and β being the design parameters to control trust propagation for these two sources of information.

We formally analyzed the effect α and β on the convergence, accuracy, and resiliency properties of our trust management protocol, and validated our analysis using simulation. The results demonstrate that (1) the trust evaluation of our protocol converges to the ground truth status in dynamic IoT environments, (2) one can tradeoff trust convergence speed for low trust fluctuation, and (3) our protocol is resilient to misbehaving attacks. We demonstrated the effectiveness of our trust management protocol by a service composition application in IoT environments. The results showed that trust-based service composition outperforms random service composition and approaches the maximum achievable performance from ground truth. The best trust-based trust service composition can be dynamically achieved by choosing the best design parameter settings in response to increasing hostility over time.

In this paper we only considered increasing hostility over time as an instance of changing environment conditions. In the future, we plan to test our dynamic trust protocol's resiliency toward a multitude of changing environment conditions to which an IoT application (e.g., service composition) can automatically and autonomously adjust the best trust parameter settings dynamically to maximize application performance.

6. REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," *Computer Networks,* vol. 54, no. 15, Oct. 2010, pp. 2787-2805.

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

[3] F. Bao, and I.-R. Chen, "Trust Management for the Internet of Things and Its Application to Service Composition," in *IEEE WoWMoM 2012 Workshop on the Internet of Things: Smart Objects and Services*, San Francisco, CA, USA, June 2012.

[4] C. Chen, and S. Helal, "A Device-Centric Approach to a Safer Internet of Things," in *the 2011 International Workshop on Networking and Object Memories for the Internet of Things*, Beijing, China, Sep. 2011, pp. 1-6.

[5] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang, "TRM-IoT: A Trust Management Model Based on Fuzzy Reputation for Internet of Things," *Computer Science and Information Systems,* vol. 8, no. 4, Oct. 2011, pp. 1207-1228.

[6] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton, and T. Razafindralambo, "A Survey on Facilities for Experimental Internet of Things Research," *IEEE Communications Magazine,* vol. 49, no. 11, Nov. 2011, pp. 58-67.

[7] M. S. Islam, R. H. Khan, and D. M. Bappy, "A Hierarchical Intrusion Detection System in Wireless Sensor Networks," *Computer Science and Network Security,* vol. 10, no. 8, August 2010, pp. 21-26.

[8] Q. Li, S. Zhu, and G. Cao, "Routing in Socially Selfish Delay Tolerant Networks," in *IEEE Conference on Computer Communications*, San Diego, CA, March 2010, pp. 1-9.

[9] X. Li, R. Lu, X. Liang, X. Shen, J. Chen, and X. Lin, "Smart Community: An Internet of Things Application," *IEEE Communications Magazine,* vol. 49, no. 11, Nov. 2011, pp. 68-75.

[10] W. Ren, "QoS-aware and compromise-resilient key management scheme for heterogeneous wireless Internet of Things," *International Journal of Network Management,* vol. 21, no. 4, July 2011, pp. 284-299.

[11] R. Roman, P. Najera, and J. Lopez, "Securing the Internet of Things," *Computer,* vol. 44, no. 9, Sep. 2011, pp. 51-58.

[12] A. d. Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, and H. Wong, "Decentralized Intrusion Detection in Wireless Sensor Networks," in *ACM International Workshop on Quality of Service and Security in Wireless and Mobile Networks*, Montreal, Quebec, Canada, Oct. 2005, pp. 16-23.

[13] S. Tozlu, M. Senel, W. Mao, and A. Keshavarzian, "Wi-Fi Enabled Sensors for Internet of Things: A Practical Approach," *IEEE Communications Magazine,* vol. 50, no. 6, June 2012, pp. 134-143.

[14] L. Zhou, and H.-C. Chao, "Multimedia Traffic Security Architecture for the Internet of Things," *IEEE Network,* vol. 25, no. 3, May-June 2011, pp. 35-40.