Trust as a Service for IoT Service Management in Smart Cities

Ing-Ray Chen, Jia Guo
Computer Science
Virginia Tech
irchen@vt.edu
jiaguo@vt.edu

Ding-Chau Wang
Information Management
Southern Taiwan University of Science and Technology
wangdc.stut@gmail.com

Jeffrey J.P. Tsai
Bioinformatics and Biomedical Engineering
Asia University
jjptsai@gmail.com

Hamid Al-Hamadi
Computer Science
Kuwait University
hamid@cs.ku.edu.kw

Ilsun You
Information Security Engineering
Soonchunhyang University
ilsunu@gmail.com

***Abstract*— We propose and analyze a cloud utility called *Trust as a Service* (TaaS) for service management of Internet of Things (IoT) devices in smart cities. The major challenge in IoT service management in smart cities is the selection of trustworthy IoT service providers. TaaS preserves the notion that trust is subjective despite the fact that trust computation is performed by the centralized cloud. We validate TaaS with two smart city IoT applications and compare its performance against contemporary trust-based service management protocols.**

Keywords—Internet of things, smart city, trust management, service management.

1. INTRODUCTION

A future smart city will consist of a huge number of autonomous Internet of Things (IoT) devices capable of providing services upon request [2]. The major challenge in IoT service management is the selection of trustworthy IoT service providers because not all IoT devices will be trustworthy and some IoT devices may behave maliciously to disrupt the IoT service management framework (e.g., in a terrorist attack scenario) or just for their own gain (e.g., to monopoly a particular type of service). In this paper, we propose and analyze a cloud utility called *Trust as a Service* (TaaS) for service management of IoT objects in smart cities. Under TaaS, an IoT device can query the "service trustworthiness" score in the range of 0 to 1 of a target IoT device for a particular type of service through the cloud utility. Based on the service trustworthiness score received, the IoT device can then determine if it wants to request or receive service from the target IoT service provider.

In our design, TaaS is made possible by a group of cloud servers (in a public cloud) whose size scales linearly with the number of IoT devices. TaaS is realized by following a novel "report-and-query" paradigm. Specifically, a user upon a service completion simply reports to its home cloud server of the user satisfaction result. To know if a target IoT device is trustworthy in providing a particular service, a user simply sends a query to its home cloud server even if the user has not had any service experience with the target IoT device. The server will return a trust value formed by considering self-observations from the user (if any exists) as well as recommendations from other users filtered by the user's own opinion if these recommenders are credible. Our design preserves the notion that trust is "subjective" despite the fact that trust computation is performed in the centralized cloud.

We validate TaaS by applying TaaS to two smart city IoT applications. We compare the performance of TaaS against two most-cited trust-based IoT service management protocols to-date, namely, Adaptive IoT Trust [8] and ObjectiveTrust [19], in selecting trustworthy service providers to maximize the application performance. The contribution of our work lies in design and validation of TaaS demonstrating its applicability as well as its superiority over contemporary trust-based IoT service management protocols when applying to smart city IoT applications.

The rest of the paper is organized as follows. Section 2 surveys related work and contrasts TaaS with existing trust-based IoT service management protocols. Section 3 discusses the system model. Section 4 provides a detailed description of TaaS design and implementation. In Section 5 we validate TaaS with two smart city IoT applications and conduct a comparative performance analysis of TaaS against Adaptive IoT Trust [8] and ObjectiveTrust [19]. Finally in Section 6 we conclude the paper and outline some future research areas.

2. RELATED WORK

Trust management protocols for IoT systems are still emerging. A comprehensive survey can be found in [12]. There are only a handful of IoT trust protocols designed and evaluated to-date [1, 3-8, 19, 20, 23]. Among the contemporary IoT trust management protocols, we select two very recent yet most cited protocols, namely, Adaptive IoT Trust [8] and ObjectiveTrust [19], as baseline IoT trust protocols against which TaaS is compared for a comparative performance analysis.

The reason we select Adaptive IoT Trust [8] is that it, like TaaS, also considers adaptive trust management to dynamically combine own experiences with recommendation based on the amount of own experiences in hand and uses social similarity as credibility for recommendation filtering. Also, it was shown in [8] that Adaptive IoT Trust outperforms existing distributed P2P trust protocols, including EigenTrust [15], PeerTrust [26], and ServiceTrust [25], so we are interested in knowing if TaaS, a cloud-based IoT trust protocol, can perform better than Adaptive IoT Trust, a proven distributed IoT trust protocol. The reason we select ObjectiveTrust [19] is that it is the only other centralized IoT trust protocol to-date that considers social standing and relationships for credibility rating and recommendation filtering.

Below we provide an overview of the two baseline trust-based IoT service management protocols and compare and contrast them with TaaS.

Adaptive IoT Trust [8] is a distributed IoT trust management protocol where each IoT device evaluates other IoT devices using both direct service experiences and indirect recommendations. Adaptive trust management is achieved by determining the best way to combine direct trust (from direct experiences) and indirect trust (from recommendations) dynamically to minimize convergence time and trust estimation bias in the presence of malicious nodes performing collusion attacks. Direct service experiences are collected based on own service experiences, while recommendations are collected at the time nodes encounter each other through social contacts. They used social similarity to rate recommenders. A common problem with a distributed IoT trust protocol such as Adaptive IoT Trust [8] is that a node may not encounter each other often to collect enough recommendations to make informed decisions. Also all trust data are stored by individual IoT devices, which can be a problem for resource-constrained IoT devices, especially when the number of IoT devices is high in a large-scale IoT system. Our approach based on TaaS does not have such constraints.

ObjectiveTrust [19] is a centralized IoT trust management system that assesses the trust score of a node through a weighted sum of the "centrality" score and the average opinion score (long term and short term) after applying the recommender's credibility score to filter untrustworthy recommendations. Specifically, ObjectiveTrust computes the centrality score (in the range of 0 to 1) of j based on if j is central in the network and if it is involved in many transactions. The credibility score of k (a recommender that provides opinions about i) is proportional to k's trust score because a trustworthy node does not lie, but is inversely proportional to the capability of k, the strong object relationship (including ownership, co-location, co-work, social, and parental) between i and k, and the number of transactions between i and k because high-capability and intimate nodes may collude. A common problem of a centralized IoT trust protocol such as ObjectiveTrust [19] is that it only computes the "objective trust" (common belief or reputation), not the "subjective trust" of an IoT device as TaaS and Adaptive IoT Trust do, so it does not preserve the notion that trust is subjective and is inherently one-to-one. This is especially problematic for IoT systems since IoT devices are owned by humans who have social relationships among themselves and the trust of one user toward another user is inherently one-to-one and subjective.

3. SYSTEM MODEL

3.1 TaaS Model

As illustrated in Figure 1, we consider a smart city environment populated with a large number of smart IoT devices which can be service providers (SPs) when they provide service or service requestors (SRs) when they request for service, with NC cloud servers (in a public cloud) being allocated to implement TaaS as a cloud utility to users participating in a particular IoT service community such as an ozone (O3) health group. Users in the O3 health group voluntarily sense and report the O3 levels of the locations they roam into. These cloud servers are assumed to be trusted and, among many service functions, provide TaaS for an IoT device to query the one-to-one "subjective" trust level of a target IoT device for the purpose of determining if the O3 level reported by the target IoT device is trustworthy.

We assume that each node (a user or an IoT device) has its unique identity. A user's unique id is at the cloud service level. An IoT device's unique id is at the device level. Each user maps to a "home" cloud server using its unique id based on distributed hash table techniques. In Figure 1, CS_2 is the home cloud server of user u_2 and CS_3 is the home cloud server of user u_3. For the case in which a user owns several IoT devices, all IoT devices also map to the owner's home cloud server as their home cloud server.

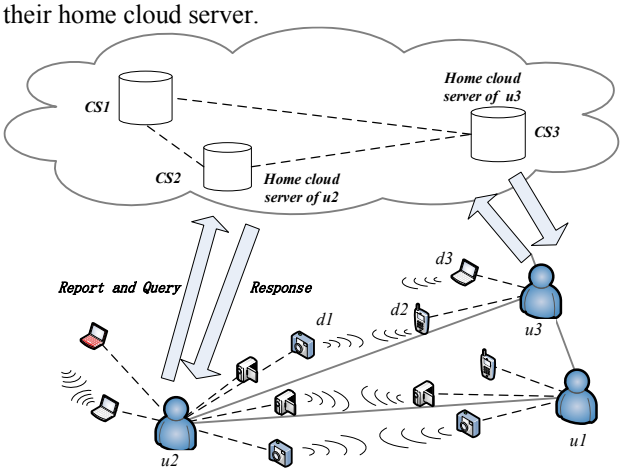

Figure 1: Information Flow in TaaS.

Whether a user has confidence on a recommender is based on social relationships among humans who are owners of IoT devices. Hence the trustor is a user and the trustee is an IoT device (owned by another user). The trust relationship is not between a user trustor and a user trustee because a user may own several IoT devices with vastly heterogeneous capabilities. Specifically, we use the social relationships between a trustor and a recommender (who provides a service trustworthiness score or service rating toward a trustee) as the trustor's "subjective" credibility toward the recommender. We consider three core social metrics for measuring social relationships which are multifaceted [24]: friendship (representing intimacy), social contact (representing closeness), and community of interest (representing knowledge and standard on the subject matter). The idea is that two users sharing similar social relationships are likely to have similar subjective views towards services provided by a trustee IoT device. Social relationships between owners are translated into social relationships between IoT devices as follows:

1. *Friendship*: Each owner has a list of friends (i.e., other owners), representing its social relationships. Specifically, each user u_x maintains a set of friends, denoted by $F_x = \{u_1, u_2, \dots\}$. This friendship list varies dynamically as an owner makes or denies other owners as friends. If the owners of two IoT devices are friends, then it is likely they will be cooperative with each other.

2. *Social Contact*: A device may be carried or operated by its owner in certain environments (e.g., at work, school, home, or social-event locations). Two devices have high social contact opportunities when their owners have similar mobility patterns or go to the same locations. Specifically, each user u_x maintains a set of locations that u_x frequently visited for social contact, denoted by a set $S_x = \{Loc_1, Loc_2, \dots\}$.
3. *Community of Interest* (CoI): Each owner has a list of communities of interest such as health, sport, travel, etc. Nodes belonging to a similar set of communities likely share similar interests or capabilities [1]. Specifically, each user u_x maintains a set of communities of interest that u_x is a member of, denoted by a set $C_x = \{CoI_1, CoI_2, \dots\}$.

3.2 Attack Model

A malicious node in general can perform communication protocol attacks to disrupt network operations. We assume such attack is handled by intrusion detection techniques [10, 17, 18] and is not addressed in this paper. 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 recommendation attacks. Bad-mouthing and ballot-stuffing attacks can be considered as a form of collaborative attacks to the trust system to ruin the trust of (and thus to victimize) good nodes and to boost the trust of malicious nodes. Self-promoting and opportunistic service attacks are the most common forms of attacks based on self-interest [9, 11, 27]. In this paper we consider a malicious IoT device (because its owner is malicious) capable of performing the following trust-related attacks (on top of on-off attacks) and our TaaS protocol design must maintain desirable accuracy, convergence, and resiliency properties against these attacks:

1. *Self-promoting attacks*: a service provider can promote its importance (by providing good recommendations for itself) for it to be selected as a service provider, but then can provide bad or malfunctioned service.
2. *Bad-mouthing attacks*: a recommender can ruin the reputation of a well-behaved IoT service provider (by providing bad recommendations against it) so as to decrease the chance of that good device being selected as a service provider.
3. *Ballot-stuffing attacks*: a recommender can boost the reputation of a misbehaving IoT service provider (by providing good recommendations) so as to increase the chance of that bad device being selected as a service provider.
4. *Discriminatory attacks* (or conflicting behavior attacks): a service provider can discriminatively attack non-friends or nodes without strong social ties (without many common friends) because of human nature or propensity towards friends in social IoT systems.
5. *Opportunistic service attacks*: a malicious node can provide good service to gain high reputation opportunistically especially when it senses its reputation is dropping because of providing bad service. With good reputation, it can effectively collude with other bad node to perform bad-mouthing and ballot-stuffing attacks.

4. TAAS PROTOCOL DESCRIPTION

4.1 Reporting

Whenever a service is rendered, a user (using its primary IoT device) reports whether it is satisfied with the service provided by an IoT device to the user's home cloud server via a service rating report. Let the current user satisfaction experience of user u_x toward device d_i be represented by a value, $f_{x,i}$ which can be a real number in the range of 0 to 1 indicating the user satisfaction level, or simply a binary value, with 1 indicating satisfied and 0 not satisfied. Here $f_{x,i}$ is the first piece of information sent from u_x to its home cloud server. For example in Figure 1, u_3 will send $f_{3,1}$ to CS_3, the home cloud server of u_3, whenever a service is rendered by d_1. A timestamp is also sent in the report to indicate the time at which this service rating happens. This allows cloud servers to know the event occurrence times of reports for regression analysis if necessary.

When user u_x encounters user u_y they exchange their (F_x, S_x, C_x) and (F_y, S_y, C_y) profiles so as to measure their mutual social similarity. For privacy and authentication, user u_x uses a cryptographic hash function in combination with a secret session key K to generate a hash-based message authentication code HMAC(K, x) for $x \in (F_x, S_x, C_x)$ and then transmits HMAC(K, x) along with HMAC(K, HMAC(K, x)) to u_y. When u_y receives the message, it can unilaterally generate HMAC(K, HMAC(K, x)) using HMAC(K, x) sent by u_x. If this matches with HMAC(K, HMAC(K, x)) sent by u_x, then u_y verifies the authentication of the message received. Then u_y can compare HMAC(K, x) with HMAC(K, y) for $y \in (F_y, S_y, C_y)$. If HMAC(K, x)=HMAC(K, y) then x=y and a common friend/device is identified. If HMAC(K, x)≠HMAC(K, y), it prevents the identities of uncommon friends/devices from being revealed to preserve privacy. With the *(F, S, X)* profile exchanged, user u_x applies cosine similarity as in [8] to compute the social similarity between u_x and u_y in friendship, social contact and CoI, denoted by $sim_i(u_x, u_y)$, $i \in \{f, s, c\}$, which is the second piece of information sent from u_x to its home cloud server. The above three social similarity measures (sim_f, sim_s, sim_c) are computed upon encountering of user u_x and user u_y, and are stored in the home cloud servers of user u_x and user u_y. For example, in Figure 1 after u_2 encounters u_3 they will each compute the three social similarity measures (sim_f, sim_s, sim_c) and store the results in the home cloud servers CS_2 and CS_3, respectively. When a home server receives these similarity scores, the home server applies a social relationship weighted sum formula $sim(u_x, u_y) = \sum_{i \in \{f,s,c\}} w_i \times sim_i(u_x, u_y)$ to compute the overall similarity score between u_x and u_y. The weights assigned to $sim_i(u_x, u_y)$, $i \in \{f, s, c\}$, depend on the application characteristics and the designer's belief of what similarity metric is more important than others in composing the overall similarity score between two users. We consider $w_f = w_s =$

$w_c = 1/3$ in this paper.

4.2 Querying and Replying

Whenever a user wants to know the trust value of an IoT device, it simply sends a query to its home cloud server. For example, in Figure 1, u_2 will send a query to its home cloud server CS_2 to know the "subjective" trust value of d_2 which belongs to u_3.

Let the "subjective" trust value of user u_x toward d_i be denoted by $t_{x,i}$. The home cloud server of u_x computes $t_{x,i}$ by combining $u_x's$ direct trust toward d_i ($t_{x,i}^d$) based on its own service rating reports, and $u_x's$ indirect trust toward d_i ($t_{x,i}^r$) based on other service rating reports submitted by other users, as follows:

$$t_{x,i} = \mu_{x,i} \cdot t_{x,i}^d + (1 - \mu_{x,i}) \cdot t_{x,i}^r \quad (1)$$

Here, $\mu_{x,i}$ is a weight parameter ($0 \le \mu \le 1$) to weigh the importance of direct trust relative to indirect trust. The selection of $\mu_{x,i}$ is critical to trust evaluation. As in [8], we apply adaptive filtering to adjust $\mu_{x,i}$ dynamically to effectively cope with malicious attacks and to improve trust evaluation performance.

To compute direct trust $t_{x,i}^d$, we adopt Bayesian framework [14] as the underlying model. The reason we choose Bayesian because it is well-established and because of its popularity in trust/reputation systems. In service computing, a service requester would rate a service provider after a service is rendered based on nonfunctional characteristics. The nonfunctional characteristics include user-observed service delay, service quality received, prices, etc. Then, we can consider the service rating $f_{x,i}$ as a Bernoulli trial with the probability of success parameter $\theta_{x,i}$ following a Beta distribution (a conjugate prior for the Bernoulli distribution), i.e., Beta($\alpha_{x,i}$, $\beta_{x,i}$). Then, the posterior $p(\theta_{x,i}|f_{x,i})$ has a Beta distribution as well. The model parameters $\alpha_{x,i}$ and $\beta_{x,i}$ are updated as follows:

$$\begin{aligned} \alpha_{x,i} &= \alpha_{x,i}^{(old)} + f_{x,i} \\ \beta_{x,i} &= \beta_{x,i}^{(old)} + 1 - f_{x,i} \end{aligned} \quad (2)$$

In Equation (2), $f_{x,i}$ contributes to positive service experience and $1 - f_{x,i}$ contributes to negative service experience. The direct trust $t_{x,i}^d$ of user u_x toward device d_i then can be computed as the expected value of $\theta_{x,i}$, i.e.,

$$t_{x,i}^d = E[\theta_{x,i}] = \frac{\alpha_{x,i}}{\alpha_{x,i} + \beta_{x,i}} \quad (3)$$

Specifically, the home cloud server of u_x updates $\alpha_{x,i}$ and $\beta_{x,i}$ whenever it receives $f_{x,i}$ (a service rating report) from user u_x based on Equation (2) and then computes $t_{x,i}^d$ based on Equation (3).

To compute indirect trust $t_{x,i}^r$, the home cloud server of u_x first locates social similarity records $sim(u_x, u_y)'s$ in its local storage. The home cloud server of u_x then selects top-R recommendations from R users with the highest similarity values with u_x and calculates the indirect trust ($t_{x,i}^r$) towards device d_i as follows:

$$t_{x,i}^r = \sum_{u_y \in U} \frac{sim(u_x, u_y)}{\sum_{u_z \in U} sim(u_x, u_z)} \cdot t_{y,i}^d \quad (4)$$

Here, U is a set of up to R users (R=5 in this paper) whose $sim(u_x, u_y)$ values are the highest, and $t_{y,i}^d$ is the rating or recommendation provided by user u_y toward device d_i, which is stored in the home cloud server of u_y but obtainable after the home cloud server of u_x communicates with the home cloud server of u_y.

In Equation (4), the feedback from u_y toward d_i $(\text{i.e.}, t_{y,i}^d)$ is weighted by the ratio of the similarity score toward the rater to the sum of the similarity scores toward all raters. Here we note that if u_y is malicious, then it can provide $t_{y,i}^d$=0 against a good IoT device to perform bad-mouthing attacks, and $t_{y,i}^d$=1 for a bad IoT device to perform ballot-stuffing attacks.

5. Applying TaaS to Smart City IoT Applications

In this section, we apply TaaS to two smart city IoT applications. We compare TaaS performanace against Adaptive IoT Trust [8] and ObjectiveTrust [19].

5.1 Smart City IoT Application 1: IoT Cloud Participatory Sensing of Air Quality

This IoT application is taken from [16] where IoT devices (e.g., smart phones carried by humans or smart cars driven by humans) can act as participants to collect air quality data and submit to a processing center located in the cloud for environmental data analysis. It is especially applicable to a health IoT group where the main concern is about a pollutant (O3 in our case study). Users in the group report their O3 sensing results upon receiving a query from a member who wishes to find out a location's O3 level at a particular time to decide if it should enter the location based on its susceptibility to the O3 level detected.

We use real traces of O3 levels and mobility traces of users in the O3 health group in the city of Houston and apply it to our participatory sensing case study. The original dataset in [22] covers the socio-demographically relevant activity sequences and the movements of each individual in 4.9 million synthetic individuals in the Houston metropolitan area. We extract a portion of this huge database to cover a smaller set of members in the O3 health group along with their mobility and activity data around a smaller area. In the case study, we assume a percentage of nodes, denoted by P_M in the range of [0, 30%], are malicious.

Every day this "good" member issues queries to its home cloud server before it enters a particular location to know the O3 level in the location it is about to step into. After collecting a number of O3 reports from other members, it then performs a trust-weighted computation to deduce the O3 reading (described later). If the O3 level is below a threshold, it would follow its route; otherwise, it will not enter the location or it will detour to avoid the location because the location has a high O3 level that can harm its owner's health. After the query-and-response event is completed, this "good" member will assess if an O3 sensing report submitted by another member is satisfactory and will submit the service experience to its home

server so as to facilitate the implementation of the TaaS cloud utility with TaaS as the underlying trust protocol.

A node (node i) in the O3 health group can query the ozone level in a particular location and at a particular time via a mobile IoT cloud application installed in its smartphone. The mobile application would send the query to all O3 health members that are in this particular location via the mobile cloud application. Upon receiving O3 sensing reports from other members, node i sends queries via TaaS to get the trustworthiness scores of these IoT devices who had reported sensing reports. To filter out untrustworthy O3 sensing reports, node i first accepts a sensing report (S_j) from j only if j is deemed trustworthy for O3 sensing service (i.e., i's trust score toward j, t_{ij}, is higher than a minimum trust threshold of 0.5). Then it computes a trust-weighted O3 level average as follows:

$$S = \sum_{j=1}^{N} (t_{ij} / \sum_{j=1}^{N} t_{ij}) \times S_j. \qquad (5)$$

where N is the number of trustworthy members providing O3 sensing reports in the particular location. If the average O3 level exceeds a maximum threshold defined by i's owner, node i will decide not to visit the location because the ozone level will cause harm to its owner's health.

Using the ns-3 simulator [21], we simulate the participatory sensing system. We use real traces of O3 levels and mobility traces of users in the O3 health group in the city of Houston [22]. The O3 level can be classified as good condition [0, 50] ug/m^3, medium condition [51, 168] ug/m^3 (unhealthy for sensitive groups), poor condition [169, 208] ug/m^3, and severe condition [209+] ug/m^3. The percentage of bad nodes is set at P_M in the range of [0, 30%]. A malicious node always reports O3 readings in poor condition range [169, 208] ug/m^3 regardless of location with the intention to break the system. Also a malicious node will perform bad-mouthing attacks (saying a good node's sensing result is not trustworthy in the user satisfaction report) and ballot-stuffing attacks (saying a bad node's sensing result is trustworthy) when it submits a service rating report recording its satisfaction experience s_i toward device d_i.

We compare TaaS with Adaptive IoT Trust [8] and ObjectiveTrust [19]. See Section 2 why we select these two protocols as the baseline protocols for performance comparison. We measure two performance metrics for performance evaluation:

1. the trust-weighted average O3 reading vs. ground truth (i.e., the actual O3 level at a specific location and a particular time);
2. the accuracy of selecting trustworthy participants.

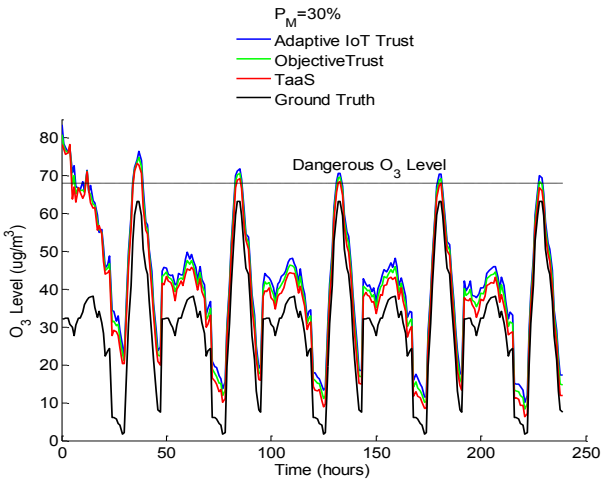

Figure 2: Trust-weighted O3 Readings vs. Query Time for TaaS against Adaptive IoT Trust and ObjectiveTrust.

Figure 2 shows the trust-weighted average O3 readings vs. the O3-level query time by a selected IoT device acting as a service requester (SR) asking for O3 readings at various locations it roams into. The percentage of bad nodes P_M is set at 30% representing a high attacker density scenario. In the experiment, the SR repeatedly queries the ozone level in the location that he will visit next over a 250 hour span. Each data point under a particular trust protocol is the average O3 level obtained from Equation 5. For example, at time t = 10 hours, the SR node sends queries via TaaS to get the trustworthiness scores of those IoT devices that have supplied O3 readings in the particular location. The SR node accepts results (S_j) from 557 trustworthy IoT devices (for which the trust score is higher than 0.5) for the O3 sensing service out of all 764 members in that particular location at that particular time and it then computes the average O3 level based on Equation 5.

The results indicate that TaaS (red line) can provide O3 readings very close to ground truth (black line) as time progresses. Further, TaaS outperforms Adaptive IoT Trust (orange line) and ObjectiveTrust (green line) in terms of accuracy (i.e., the difference between ground truth and the average O3 levels) and resiliency (against malicious attacks of 30% bad nodes). We attribute this to its ability to effectively and adaptively aggregate trust evidence from all nodes in the system through our effective and efficient localized report-and-query mechanism design. We draw a line "Dangerous O3 Level" for a user whose "dangerous O3 level" is 68 as diagnosed by his/her doctor as vulnerable to O3 exposure for long hours. We see that at time t=130, 180, or 235 (the last three peaks in the figure) only TaaS will correctly identify the fact that O3 level is below the dangerous level, while either Adaptive IoT Trust or ObjectiveTrust will raise a false alarm that the dangerous O3 level for this user is already reached.

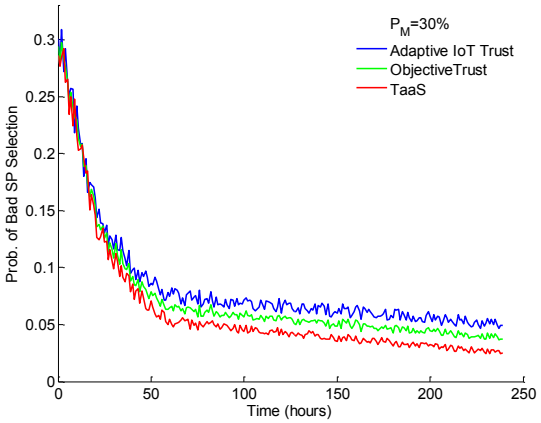

Figure 3: Percentage of Bad IoT Devices Selected to Provide O3 Sensing Service vs. Query Time for TaaS against Adaptive IoT Trust and ObjectiveTrust.

Figure 3 shows the percentage of bad nodes selected to provide sensing results to the SR. TaaS outperforms Adaptive IoT Trust and ObjectiveTrust as time progresses as more evidence are collected. The results can be explained as follows: Compared with Adaptive IoT Trust, TaaS is not being limited by encountering experiences and can leverage cloud service to aggregate broad evidence from all nodes who have had sensing service experiences with participating IoT devices. Compared with ObjectiveTrust which is based on "objective trust" (i.e., common belief), TaaS is based on "subjective trust" (with one-to-one belief) and can adaptively put a higher weight on a participant if it has had good O3 sensing experiences with the particular participant. This allows TaaS to more effectively select trustworthy participants among all participants that had submitted O3 sensing reports.

5.2 Smart City IoT Application 2: Travel Planning

This smart city IoT application is taken from [7]: Ed has never visited New York City. He wants to plan his travel early on from Seattle, including airline reservation from Seattle to New York, ground transportation after reaching New York, hotel reservation at New York, entertainments and attractions while in New York, hotel shuttle to the airport, etc. Ed instructs his smart phone to first construct a workflow structure for the travel and then select from a myriad of IoT SPs to populate the workflow structure.

Figure 4 shows the service flow structure constructed by his smartphone. There are 9 atomic services connected by three types of workflow structures, namely, *sequential*, *parallel* (AND), and *selection* (OR). Each service would have multiple SP candidates. The "service trustworthiness score" of a candidate service composition based on the service flow structure in Figure 4 can be calculated recursively in the same way the reliability of a series-parallel connected system is calculated. Specifically, the service trustworthiness score of a composite service (whose trustworthiness score is T_s) that consists of two subservices (whose trustworthiness scores are T_1 and T_2) depends on the structure connecting the two subservices as follows:

- Sequential structure: $T_s = T_1 \times T_2$;
- Selection structure (OR): $T_s = \max(T_1, T_2)$;
- Parallel structure (AND): $T_s = 1 - (1 - T_1) \times (1 - T_2)$.

Hence given the knowledge of the trustworthiness scores of individual IoT service providers (SPs) in a composite service and the configuration of the service composite, we can recursively compute the overall trustworthiness score of the composite service.

We again compare TaaS with ObjectiveTrust [19] and Adaptive IoT Trust [8]. We measure two performance metrics for performance evaluation for this IoT application:

1. the overall trustworthiness score (called *utility score*) of the composite service after service composition and binding;
2. the accuracy of selecting trustworthy IoT service provider for service composition and binding.

In this IoT application we consider service constraints in terms of a budget limit. Simply selecting the most trustworthy SPs may lead to infeasible solutions. Suppose that an IoT device acting as the SR has a budget limit for the travel planning composite service. Each IoT SP announces its price when publishing its service (e.g., car rental, public transportation, or taxi for transportation service). The SR would calculate the overall utility score and the overall price for each candidate composite service based on the configuration of the composite service as described above. Then the SR would select the composite service candidate with the highest utility score among all composite service candidates with the overall price below the budget limit.

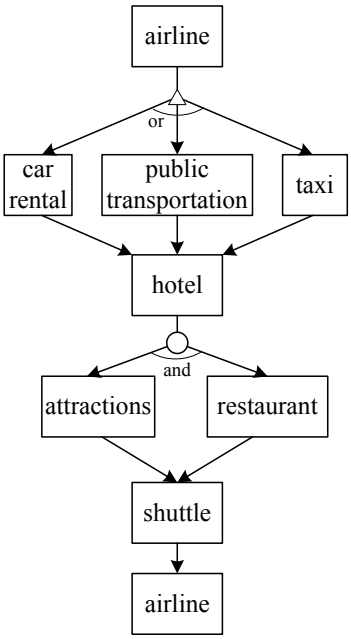

Figure 4: A Service Flow Structure for the Smart City Travel Planning IoT Application.

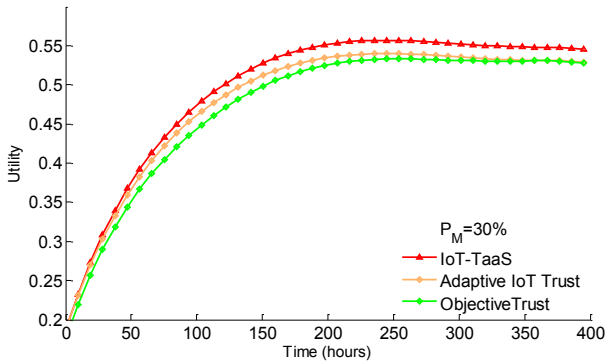

Figure 5: Utility Score of the Travel Planning IoT Application for TaaS against Adaptive IoT Trust and ObjectiveTrust.

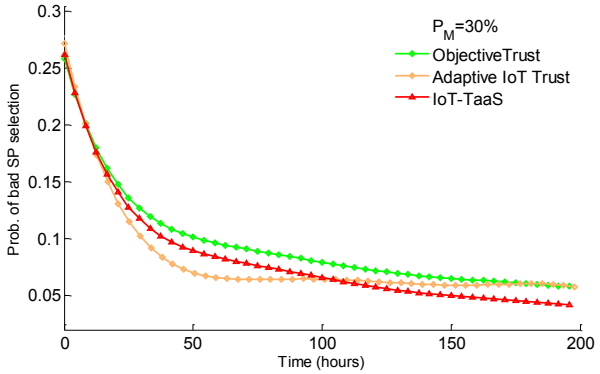

Figure 6: Probability of a Bad SP Being Selected for the Travel Planning IoT Application for TaaS against Adaptive IoT Trust and ObjectiveTrust.

Figure 5 shows the utility score obtainable vs. query time for TaaS against Adaptive IoT Trust [8] and ObjectiveTrust [19] based on ns-3 simulation results with P_M=30%. We again observe that the trend is similar in terms of performance ranking, with TaaS (red curve) outperforming Adaptive IoT Trust and ObjectiveTrust.

Figure 6 shows the percentage of bad nodes selected for service composition with budget limit constraints. TaaS (red curve) again outperforms Adaptive IoT Trust and ObjectiveTrust by a significant margin, especially as time progresses allowing the TaaS to gather broad evidence from all IoT devices that have had prior service experiences to other IoT devices. We attribute the superiority of TaaS over Adaptive IoT Trust and ObjectiveTrust to its adaptability in response to a high percentage of nodes (30% in this case study) performing malicious attacks.

6. Conclusion

In this paper, we designed and analyzed a cloud utility called TaaS for service management of IoT objects for smart cities. We demonstrated via ns-3 simulation the superiority of TaaS over contemporary, most-cited IoT trust protocols to date, namely, Adaptive IoT Trust [8] and ObjectiveTrust [19], in trust-based service management for two real-world smart city IoT applications. We attribute TaaS's superiority to its subjective trust evaluation, report-and-query, and adaptability designs resulting in high trust accuracy and resiliency against a high percentage of malicious nodes performing self-promoting, bad-mouthing, ballot-stuffing, discriminatory, and opportunistic service attacks.

In the future we plan to validate TaaS with more real-world smart city IoT applications such as those discussed in [2, 7]. In this paper we assumed a centralized cloud. This increases the energy consumption of IoT devices for long haul communication with the remote cloud. In addition the centralized cloud is a single point of failure. A future research direction is to devise a hierarchical cloud architecture that can achieve scalability, fault tolerance, and resiliency against trust-related attacks, while reducing the energy consumption of IoT devices.

Lastly, there is a lack of a holistic design for scalable, adaptive and survivable trust computation for IoT systems. A future 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. A possible solution is to integrate the design concepts currently existing in hierarchical trust management [44, 45], hierarchical mobility management [28-31], resilient failure recovery management [32-38], admission control [39-43], and tiered cloud architectures with edge computing capability. 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 is 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.

Acknowledgment

This work is supported in part by the U.S. AFOSR under grant number FA2386-17-1-4076. This work was also supported in part by the Institute for Information & Communications Technology Promotion (IITP) of the Government of South Korea MSIT under grant number 2017-0-00664 as well as the Soonchunhyang University Research Fund.

References

[1] H. Al-Hamadi and I.R. Chen, "Trust-Based Decision Making for Health IoT Systems," *IEEE Internet of Things Journal*, vol. 4, no. 5, Oct. 2017, pp. 1408-1419.

[2] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," *Computer Networks,* vol. 54, pp. 2787-2805, Oct. 2010.

[3] F. Bao and I. R. Chen, "Dynamic Trust Management for Internet of Things Applications," *2012 International Workshop on Self-Aware*

Internet of Things, San Jose, California, USA, 2012.

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

[5] 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, 2013.

[6] 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, pp. 1207-1228, Oct., 2011.

[7] I. R. Chen, F. Bao, and J. Guo, "Trust-based Service Management for Social Internet of Things Systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 6, Nov-Dec 2016, pp. 684-696.

[8] 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.

[9] I.R. Chen, J. Guo, F. Bao and J.H. Cho, "Trust Management in Mobile Ad Hoc Networks for Bias Minimization and Application Performance Maximization," *Ad Hoc Networks*, vol. 19, August 2014, pp. 59-74.

[10] J.H. Cho and I.R. Chen, "PROVEST: Provenance-Based Trust Model for Delay Tolerant Networks,"*IEEE Transactions on Dependable and Secure Computing,* vol. 15, no. 1, 2018, pp.151-165.

[11] J. H. Cho, A. Swami, and I. R. Chen, "Modeling and analysis of trust management with trust chain optimization in mobile ad hoc networks," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 1001-1012, May 2012.

[12] J. Guo, I. R. Chen, and J.J.P. Tsai, "A survey of trust computation models for service management in internet of things systems," *Computer Communications,* vol. 97, 2017, pp. 1-14.

[13] Z. Huang, D. Zeng, and H. Chen, "A Comparison of Collaborative-Filtering Recommendation Algorithms for E-commerce," *IEEE Intelligent Systems,* vol. 22, pp. 68-78, 2007.

[14] A. Jøsang, and R. Ismail, "The Beta Reputation System," *Bled Electronic Commerce Conference*, Bled, Slovenia, 2002, pp. 1-14.

[15] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust algorithm for reputation management in P2P networks," *12th International Conference on World Wide Web*, Budapest, Hungary, May 2003.

[16] W.Z. Khan, Y. Xiang, M.Y. Aalsalem, and Q. Arshad, Q, "Mobile Phone Sensing Systems: A Survey," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 1, 2013, pp. 402–427.

[17] R. Mitchell and I. R. Chen, "A Survey of Intrusion Detection Techniques for Cyber Physical Systems," *ACM Computing Survey*, vol. 46, article no. 4, 2014.

[18] R. Mitchell and I. R. Chen, "Modeling and Analysis of Attacks and Counter Defense Mechanisms for Cyber Physical Systems," *IEEE Transactions on Reliability*, vol. 65, no. 1, March 2016, pp. 350-358.

[19] 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.

[20] M. Nitti, L. Atzori, and I.P. Cvijikj, "Friendship Selection in the Social Internet of Things: Challenges and Possible Strategies," *IEEE Internet of Things Journal*, vol. 2, no. 3, 2015, pp. 240-247.

[21] ns-3 Network Simulator, http://www.nsnam.org, Release ns-3.27 with core, network, Internet, and mobility models, Oct. 2017.

[22] B. Pires, et al, "Towards an in silico Experimental Platform for Air Quality: Houston, TX as a Case Study," *Computational Social Science Society of America Conference*, Santa Fe, NM, USA, 2015.

[23] 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, Nov. 2013, pp. 351-365.

[24] 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.

[25] Z. Su, L. Liu, M. Li, X. Fan, and Y. Zhou, "Service Trust: Trust Management in Service Provision Networks," *IEEE International Conference on Services Computing*, Santa Clara, 2013, pp. 272-279.

[26] L. Xiong, and L. Liu, "PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities", *IEEE Trans. on Knowledge and Data Engineering*, v.16, pp. 843-857, July 2004.

[27] Y. Wang, I.R. Chen, J.H. Cho, A. Swami, and K.S. Chan, "Trust-based Service Composition and Binding with Multiple Objective Optimization in Service-Oriented Ad Hoc Networks," *IEEE Trans. Services Computing*, vol. 10, no. 4, 2017, pp. 1939-1374.

[28] 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.

[29] 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.

[30] 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.

[31] I. R. Chen, T.M. Chen, and C. Lee, "Agent-based forwarding strategies for reducing location management cost in mobile networks," *Mobile Networks and Applications*, vol. 6, no. 2, 2001, pp. 105-115.

[32] I.R. Chen, B. Gu, S.E. George, and S.T. Cheng, "On failure recoverability of client-server applications in mobile wireless environments," *IEEE Trans. Reliability*, vol. 54, no. 1, 2005, pp. 115-122.

[33] S.E. George, I.R. Chen, and Y. Jin, "Movement-based checkpointing and logging for recovery in mobile computing systems," *5th ACM International Workshop on Data Engineering for Wireless and Mobile Access,* 2006.

[34] I.R. Chen and D.C. Wang, "Analysis of Replicated Data with Repair Dependency," *The Computer Journal*, vol. 39, no. 9, 1996, pp. 767-779.

[35] I.R. Chen and D.C. Wang, "Analyzing dynamic voting using Petri nets," *15th Symposium on Reliable Distributed Systems*, 1996, pp. 44-53.

[36] I.R. Chen and F.B. Bastani, "Effect of Artificial-Intelligence Planning Procedures on System Reliability," *IEEE Trans Reliability*, vol. 40, no. 3, pp. 364-369, 1991.

[37] F.B. Bastani, I.R. Chen, and T. Tsao, "Reliability of Systems with Fuzzy-Failure Criterion," *Annual Reliability and Maintainability Symposium*, 1994, pp. 442-448.

[38] I.R. Chen, F.B. Bastani, and T.W. Tsao, "On the Reliability of AI Planning Software in Real-Time Applications," *IEEE Trans Knowledge and Data Engineering*, vol. 7, no. 1, pp. 4-13, 1995.

[39] I.R. Chen, O. Yilmaz, and I.L. Yen, "Admission control algorithms for revenue optimization with QoS guarantees in mobile wireless networks," *Wireless Personal Communications*, vol. 38, no. 3, 2006, pp. 357-376.

[40] I.R. Chen and T.H. Hsi, "Performance analysis of admission control algorithms based on reward optimization for real-time multimedia servers," *Performance Evaluation*, vol. 33, no. 2, pp. 89-112, 1998.

[41] S.T. Cheng, C.M. Chen, and I.R. Chen, "Dynamic quota-based admission control with sub-rating in multimedia servers," *Multimedia Systems*, vol. 8, no. 2, 2000, pp. 83-91.

[42] S.T. Cheng, C.M. Chen, and I.R. Chen, "Performance evaluation of an admission control algorithm: dynamic threshold with negotiations," *Performance Evaluation*, vol. 52, no. 1, 2003, pp. 1-13.

[43] O. Yilmaz and I.R. Chen, "Utilizing Call Admission Control for Pricing Optimization of Multiple Service Classes in Wireless Cellular Networks," *Computer Communications*, vol. 32, no. 2, 2009, pp. 317-323.

[44] J. Guo, I.R. Chen, and J.J.P. Tsai, "A Mobile Cloud Hierarchical Trust Management Protocol for IoT Systems," *5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, San Francisco, April 2017.

[45] J. Guo, Trust-based Service Management of Internet of Things Systems and Its Applications, ETD, Virginia Tech, 2018.