Trust-based IoT Participatory Sensing for Hazard Detection and Response

Jia Guo[1], Ing-Ray Chen[1], Jeffrey J.P. Tsai[2], Hamid Al-Hamadi[3]

[1]Virginia Tech
{jiaguo,irchen}@vt.edu
[2]Asia University
jjptsai@gmail.com
[3]Kuwait University
hamid@cs.ku.edu.kw

Abstract. The physical world can be monitored by ubiquitous Internet of Things (IoT) devices through participatory sensing by which a huge amount of data is collected and analyzed in the cloud for hazard detection and response. In this paper, we propose a Trust as a Service (TaaS) cloud utility leveraging a cloud hierarchy for assessing service trustworthiness of IoT devices so as filter out untrustworthy sensing data before hazard detection and response are taken. We demonstrate that our TaaS utility achieves accuracy, convergence, and resiliency compared with contemporary IoT/P2P distributed trust protocols while achieving scalability to cope with a huge number of IoT devices. We demonstrate the feasibility with an air pollution detection and response application.

Keywords: Internet of Things; trust; participatory sensing; cloud computing.

1 Introduction

The physical world can be monitored by ubiquitous Internet of Things (IoT) devices through participatory sensing by which a huge amount of data is collected and analyzed for hazard detection and response [1]. One possible IoT participatory sensing application is environmental monitoring where IoT devices (e.g., smart phones carried by humans) collect environmental data (noise, air pollution, temperature, humidity, light, etc.) and submit via wireless data communication links to a processing center located in the cloud for environmental data analysis [2]. In return, a user (e.g., emergency response personnel) can send a query to the cloud to query a location's air pollution levels of CO, NO_2, SO_2, and O_3. Another possible application is road/traffic monitoring by which traffic flows, pot-holes, bumps, braking, and honking information reported from IoT devices (smart phones carried by passengers/drivers in a car) are aggregated by a data processing center located in the cloud to unveil traffic patterns previously unobserved with existing monitoring infrastructure. This is especially useful in disaster response situations after the occurrence of a public hazard such as a hurricane or a terrorist attack.

The major challenges for detection and response participatory sensing applications are scalability and selection of trustworthy participants [3]. Scalability is needed considering that the number of IoT devices will grow exponentially in the next decade. Selection of trustworthy participants is needed because not all IoT devices will be trustworthy and some IoT devices may behave maliciously to disrupt the network or service (e.g., in a terrorist attack scenario) or just for their own gain (e.g., in an evacuation scenario following a disaster).

While selection of trustworthy participants has attracted some attention [4-9], scalability remains an open problem. In this paper, we develop a "Trust as a Service" (TaaS) cloud utility leveraging a cloud hierarchy so as to cope with a huge number of IoT devices to address the scalability issue. We also demonstrate that as an added benefit, TaaS addresses the selection of trustworthy participants issue better than existing distributed IoT trust protocols because it is able to aggregate broad evidence from all nodes having interaction experiences with a target IoT device.

The rest of the paper is organized as follows. Section 2 discusses how TaaS is implemented leveraging a cloud hierarchy. Section 3 demonstrates the utility of TaaS with an air pollution detection and response IoT application for which TaaS is shown to outperform existing non-scalable distributed IoT trust protocols. Section 4 summarizes the paper and outlines future work.

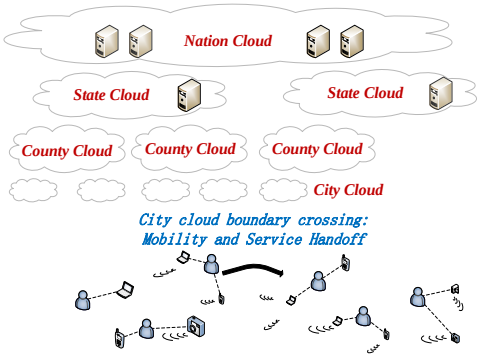

Figure 1: Hierarchical Cloud Architecture for Hazard Detection and Response.

2 TaaS Cloud Utility Leveraging a Cloud Hierarchy

Our TaaS cloud utility leverages a cloud hierarchy as illustrated by Figure 1 for integrated mobility, service, and trust management of a huge number of IoT devices [10]. We label the clouds from top to bottom as nation, state, county, and city clouds as would be needed in a federal emergency assessment, management, and response system. The city and county clouds can be base stations and routers owned by mobile network operators, while state and nation clouds can be mini and big data centers owned by cloud service providers. Each city cloud at the bottom layer can be just a base station covering a geographical region, providing a communication path for IoT devices (e.g., sensors,

smart phones, vehicles) in a region to interact with the cloud via wireless communication.

Each user is associated with a family of "home" clouds, starting from the home city cloud (base station) at the bottom layer, home county cloud (router) at the second bottom layer, home state cloud at the second top layer, and home nation cloud at the top layer. These home clouds are assigned based on the "home" geographical location of an IoT device similar to the home location register (HLR) in mobile networks [11, 12]. When an IoT device moves from one region to another region (if the IoT device is mobile), a "mobility handoff" ensues by which the city cloud which the IoT device just roams into will inform all home clouds of the IoT device of the new location.

An IoT device will only interact with its current city cloud for service invocation to minimize energy consumption and service latency. There are two standard cloud computing operations to be performed by a city cloud, *store-process-forward* and *forward-wait-reply*, described as follows. The local city cloud will examine a service request from an IoT device. If the service request is to report new service data such as a feedback or a sensing outcome, the current city cloud will follow the *store-process-forward* procedure. i.e., it will store a replicated copy of the service data, process it locally as needed, and pass the new service data to the home clouds of the IoT device. If the service request is a query regarding a target IoT device, then the current city cloud will follow the *forward-wait-reply* procedure. That is, the query will be forwarded to the least common "home" cloud of the requesting IoT device and the target IoT device. If the service request involves several IoT devices some of which are not under the current city cloud, then the city cloud will pass the request to the least common "home" cloud of these IoT devices for processing because the least common "home" cloud will store location and service data of these IoT devices. Then it will wait for a reply to return to it after which it will forward the reply to the requesting IoT device. A "service handoff" is triggered when an IoT device goes to a new city cloud, necessitating the migration of the virtual machine for cloud computing.

Our TaaS cloud utility is implemented utilizing the standard *store-process-forward* and *forward-wait-reply* procedures described above. Specifically, IoT device *i* (acting as a service requester) can simply report to its current city cloud a *user satisfaction* report of its service trustworthiness assessment toward IoT device *j* who just completed a sensing service of a specific air pollutant level (e.g., sensing CO air pollution) in the form of $(i, j, T_{ij}, apt_s, l_s, t_s)$ where T_{ij} (in [0, 1]) is the sensing result trustworthiness of *j* as assessed by *i*, apt_s is the air pollutant type (e.g., apt_s = "CO"), l_s is the location at which sensing is perform, and t_s is the time of sensing. T_{ij} can be assessed after fact, i.e., after *i* itself experiences it or verifies it after reading official reports about the level of this particular air pollutant at the particular location and particular time. If neither is accessible, T_{ij} can be assessed by the discrepancy between *j*'s sensing result from the average sensing result from all sensing results received by *i* for the same location at the same time. T_{ij} is set to 1 and can go down to 0 proportional to the amount of discrepancy detected. The city cloud upon receiving a user satisfaction report would follow the standard *store-process-forward* procedure described earlier to store the user satisfaction report to all home clouds of IoT device *i*.

An IoT device (on behalf of its owner) can simply query its local city cloud about the trustworthiness of a target IoT device for providing sensing service of a specific pollutant type. The current city cloud would follow the standard *forward-wait-reply* procedure described earlier. The least common home cloud of the requesting IoT device and the target IoT upon receiving the query will simply use all user satisfaction reports stored in its local store and apply a trust computation method such as Beta Reputation [5] or Adaptive IoT Trust [9] to assess the trustworthiness of the target node. When the trust assessment is completed, the home cloud will return the response (i.e., the trustworthiness of the target IoT device for providing service) to the city cloud who received the query who in turn will forward the response to the requesting IoT device for decision making.

3 Case Study: Air Pollution Detection and Response

The case study is for the *Fairfax County Hazard Detection and Response Team* charged to monitor the pollution levels of CO, NO_2, SO_2, and O_3 for all cities under the county so as to take appropriate actions if the air pollution level is above a tolerance threshold. Since the area to be covered is rather large, the county officials only install a few county-sensors in more strategic and populated areas to collect air pollution data. To cover the whole county area air quality detection, the county officials also encourage environment-health-conscious civilians driving or carrying air pollution detection capable vehicles or smartphones [2] to report air pollution data.

In case of emergency, the county officials can request IoT devices in a particular location to immediately report their sensing results to their respective city clouds, as the cloud hierarchy knows the locations of all home county IoT devices. Also the county officials send queries via TaaS to get the trustworthiness scores of these IoT devices. To know if a location has acceptable air quality, the county officials (running as node *i*) accept results (S_j) from 200 most trustworthy IoT devices (which have the highest T_{ij} trust values) for the air quality detection service out of a total of 2000 nodes, and compute a trust-weighted average $\sum_{j=1}^{200}(T_{ij} / \sum_{j=1}^{200} T_{ij}) \times S_j$ for each air pollutant (e.g., CO). If the level exceeds a minimum threshold (e.g., above 70 ppm for CO), the county officials push alerting text to IoT devices in the affected area.

We simulate the above system populated with 2000 IoT devices capable of detecting and reporting CO air pollutant levels using the ns3 simulator. The CO level is simulated to be in the range of [60, 70 ppm] in various locations. The percentage of bad nodes is set at P_M in the range of [0, 30%]. A malicious node always reports CO readings above 70 ppm in the range of [70, 120 ppm] regardless of location in order to confuse the county official. It can perform attacks on and off in order to evade detection. We simulate this by a random attack probability P_a in the range of [0, 100%]. Also a malicious node always performs 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).

We compare our cloud hierarchy based TaaS with existing non-scalable distributed IoT/P2P management protocols including EigenTrust [6], PeerTrust [7], ServiceTrust

[8], and Adaptive IoT Trust [9] for which each IoT device keeps own trust data based on own experiences and service satisfaction feedbacks from its peers that it encounters. For fair comparison, the environment is setup as in [9] and we also adopt Adaptive IoT Trust in [9] for trust computation. We measure two performance metrics for performance analysis: (a) the trust-weighted average CO reading vs. ground truth (i.e., the actual CO level at a specific location and a particular time); (b) the accuracy of selecting trustworthy participants.

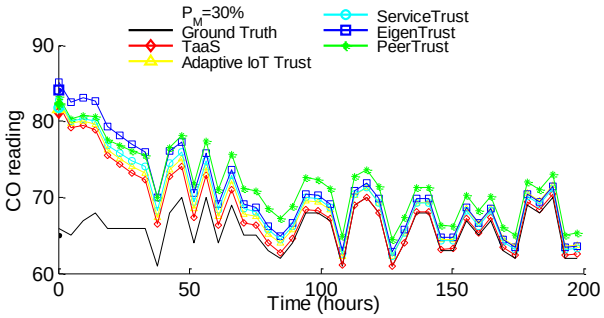

Figure 2: Performance Comparison of Trust-Weighted Average CO Readings of the Air Pollution Detection and Response Application.

Figure 2 shows the trust-weighted average CO readings vs. time (each time point is a CO detection service request) with the percentage of bad nodes P_M set at 30% and P_a set at 100%. We observe TaaS (red line) levering the proposed cloud hierarchy can provide CO readings very close to ground truth (black line) as time progresses. Further, our TaaS cloud utility outperforms EigenTrust, PeerTrust, ServiceTrust, and Adaptive IoT Trust in terms of accuracy, convergence, and resiliency due to its ability to effectively aggregate trust evidence from all nodes in the system through the simple standard store-process-forward and forward-wait-reply cloud computing paradigms.

Figure 3 shows the percentage of bad nodes selected to provide sensing results. TaaS outperforms EigenTrust, PeerTrust, ServiceTrust, and Adaptive IoT Trust as time progresses because unlike Adaptive IoT Trust [9], TaaS can leverage cloud service to aggregate broad evidence from all nodes having service experiences with IoT devices reporting sensing results.

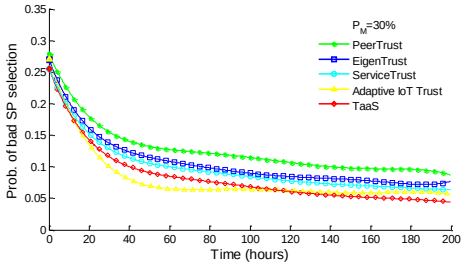

Figure 3: Percentage of Bad IoT Devices Selected to Provide CO Sensing Service for the Air Pollution Detection and Response Application.

4 Conclusion

In this paper we developed a scalable TaaS cloud utility leveraging a cloud hierarchy that can provide integrated mobility, service, and trust management of a huge number of IoT devices. Through an air pollution detection and response IoT application, we demonstrated that our TaaS cloud utility outperforms existing distributed IoT/P2P trust protocols while achieving scalability and accuracy of selecting trustworthy participants, because it can leverage simple yet powerful store-process-forward and for-ward-wait-reply cloud computing paradigms to aggregate broad service evidence from all nodes in the system.

In this paper we only conducted performance comparison of TaaS against existing distributed IoT trust protocols in terms of the accuracy of selecting trustworthy participants. In the future, we plan to conduct more experiments to quantify the gain of our scalability design in terms of performance metrics such as resource overhead, energy consumption, and service latency.

References

1. Khan, W.Z., Xiang, Y., Aalsalem, M.Y., Arshad, Q.: Mobile Phone Sensing Systems: A Survey. IEEE Communications Surveys and Tutorials, 15, 402–427 (2013)
2. Devarakonda, S. et al.: Real-time Air Quality Monitoring Through Mobile Sensing in Metropolitan Areas. In: UrbComp, Chicago, Illinois, USA (2013)
3. Mousa, H., et al.: Trust Management and Reputations Systems in Mobile Participatory Sensing Applications: A Survey. Computer Networks, 90, 49-73 (2015)
4. Amintoosi, H., Kanhere, S.S., Allahbakhsh, M.: Trust-based Privacy-aware Participant Selection in Social Participatory Sensing. J. Information Security and Applications, 20, 11-25 (2015)
5. Jøsang, A., Ismail, R.: The Beta Reputation System. In Bled Electronic Commerce Conference, Bled, Slovenia, pp. 1-14 (2002)
6. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The EigenTrust Algorithm for Reputation Management in P2P Networks. In: 12th International Conference on World Wide Web, Budapest, Hungary (2003)
7. Xiong, L., Liu, L.: PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities. IEEE Trans. on Knowledge and Data Engineering, 16, 843-857 (2004)
8. Su, Z., Liu, L., Li, M., Fan, X., Zhou, Y.: ServiceTrust: Trust Management in Service Provision Networks. In: IEEE International Conf. on Services Computing, Santa Clara, pp. 272-279 (2013)
9. Chen, I.R., Guo, J., Bao, F.: Trust Management for SOA-based IoT and Its Application to Service Composition. IEEE Transactions on Services Computing, 9, 482-495 (2016)
10. Chen, I.R., Guo, J., Tsai, J.J.P., Al-Hamadi, H.: A Hierarchical Cloud Architecture for Integrated Mobility, Service, and Trust Management of Service-Oriented IoT Systems. In: 6th IEEE International Conference on Innovative Computing Technology, Dublin, Ireland (2016)
11. Chen, I.R., Chen, T.M., Lee, C.: Performance evaluation of forwarding strategies for location management in mobile networks. The Computer Journal, 41, 243–253 (1998)
12. Chen, I.R., Verma, N.: Simulation Study of A Class of Autonomous Host-centric Mobility Prediction Algorithms for Wireless Cellular and Ad Hoc Networks. 36th Annual Symp. Simulation (2003)