A Hierarchical Cloud Architecture for Integrated Mobility, Service, and Trust Management of Service-Oriented IoT Systems

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

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

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

Abstract — **In this paper we propose a hierarchical cloud architecture for integrated mobility, service, and trust management of service-oriented Internet of Things (IoT) systems. This architecture supports scalability, reconfigurability, fault tolerance, and resiliency against cloud node failure and network disconnection, and can benefit both network operators and cloud service providers. In particular, leveraging this architecture we develop a cloud-based trust protocol to provide trustworthiness assessment of IoT devices. With air pollution detection as an example IoT application, we demonstrate that our cloud-based trust protocol built upon the proposed cloud hierarchy outperforms existing distributed IoT trust management protocols.**

Keywords— Cloud computing; IoT; mobility management; service management; trust management; service-oriented computing.

I. Introduction

There is an urgent need to integrate Internet of Things (IoT) and cloud computing both of which have proliferated over the past 5 years to reach a global scale [1]. In service-oriented IoT systems [2], each IoT device is a service requester requesting services or resources from other IoT devices it encounters, as well as a service provider itself offering services or sharing resources to other IoT devices. Service-oriented IoT devices interact with each other via compatible service APIs (such as WS-*, REST, and CoRE) for publishing, discovery, selection, and composition of services.

In this paper we propose a hierarchical cloud architecture for integrated mobility, service, and trust management of *service-oriented* IoT devices and demonstrate its usage with a *Trust as a Service* (TaaS) cloud utility applying to an air pollution detection IoT application. In this example environmental monitoring application, IoT devices (e.g., smart phones carried by humans, public transportation buses, taxis, city light/utility poles, smart buildings, etc.) would collect environmental data (air pollutant level, location, time) and submit via wireless data communication links to a processing center located in the cloud for environmental data analysis. In return, a user would send a query to the cloud to query a location's air pollution level.

The proposed hierarchical cloud architecture can benefit both mobile network operators and cloud providers for several reasons. First, it opens up business cooperation opportunity between mobile network operators and cloud providers. Existing mobile network base stations and routers after incorporating cross-layer designs can be transformed into *nano* or *micro* clouds sitting at the bottom layers of the cloud hierarchy. In fact, mobile network operators are in unique positions to make possible integrated mobility, service, and trust management for service-oriented IoT applications, because they know location information of mobile IoT devices all the time. This allows location-based cloud service to be easily implemented. For example, in an air pollution detection and response situation, the cloud could ask for sensing services from IoT devices located in a particular location [3]. Second, proximity communication between a huge number of IoT devices and the local clouds can effectively reduce the traffic flow to *macro* or *mega* clouds sitting at the top layers of the cloud hierarchy, making cloud computing feasible for large-scale service-oriented IoT applications. Third, IoT cloud applications can be easily deployed to benefit from locality and a tighter coupling to the mobile networks. Resource-limited IoT devices can benefit from low energy consumption rate and low service latency since an IoT device only needs to interact with a local cloud (which can be just a base station) for service request/response.

The potential of the cloud hierarchy for IoT applications is tremendous. However, not all IoT devices will be trustworthy and some IoT devices may behave maliciously for disrupting the network or service (e.g., in a terrorist attack scenario) or just for their own gain (e.g., in a service provider bidding scenario), so trust management is essential to cope with misbehaving IoT devices with an effective recommendation filtering being applied to screen out untrustworthy recommendations or feedbacks [4-11]. The literature is thin in IoT trust management. To date only [2, 12-15] are found and none of them addresses the scalability issue to deal with a huge number of IoT devices. Out of these existing works, [2, 12, 13] discussed distributed IoT trust management. Although their trust protocols achieve a reasonable degree of trust accuracy, convergence, and resiliency, the scalability issue to cope with a huge number of IoT devices is not addressed. The works in [14, 15] discussed cloud-based IoT trust management assuming that a powerful cloud is in place for trust management. Their design also does not address scalability and energy consumption or service latency issues of resource-constrained IoT devices. Further the cloud is a single point of failure. Because of the distance of IoT devices from the cloud, network disconnection can often cause service disruption. In this paper, we develop a *Trust as a Service* (TaaS) cloud utility built on our proposed hierarchical cloud architecture to provide efficient and reliable trustworthiness assessment of IoT devices. We verify our trust proposal's convergence, accuracy, and resiliency properties against self-promotion, discriminatory, bad-mouthing, and ballot-stuffing attacks due to the presence of malicious IoT devices for their own gain, and illustrate the effectiveness of TaaS with an air pollution detection IoT application.

This paper has the following unique contributions:

1. We bring up the notion of integrated mobility, service, and trust management of service-oriented IoT devices utilizing a scalable cloud hierarchy to benefit both mobile network operators and cloud service providers. In addition to scalability, this architecture supports reconfigurability, fault tolerance, and resiliency against cloud failure and network disconnection, while reducing energy consumption rate and service latency of IoT devices as well as traffic flow to the cloud because IoT devices only need to interact with local clouds.
2. We develop a TaaS cloud utility leveraging the hierarchical cloud architecture and demonstrate that it can achieve trust accuracy, convergence, and resiliency against self-promoting, bad-mouthing, ballot-stuffing, and opportunistic service attacks., and can outperform existing non-scalable distributed IoT management protocols including EigenTrust [16], PeerTrust [17], ServiceTrust [18], and Adaptive IoT Trust [2] because it can leverage cloud service to aggregate broad evidence from all nodes having interaction experiences with a target IoT device. We use an air pollution detection application to demonstrate the feasibility.

The rest of the paper is organized as follows: In Section II we discuss the system model and the threat model. In Section III, we provide a description of the proposed cloud hierarchy for providing integrated mobility, service, and trust management of service-oriented Internet of Things (IoT) systems. Section IV develops a TaaS cloud utility leveraging the hierarchical cloud architecture and demonstrates that it can outperform existing non-scalable distributed IoT management protocols. Finally Section V concludes the paper and outlines future research areas.

II. System Model

A. Service-Oriented IoT Environments

We consider service-oriented IoT environments with no centralized trusted authority. Each IoT device has its unique identity which can be achieved through standard techniques such as PKI. A device communicates with other devices through the overlay social network protocols, or the underlying standard communication network protocols (wired or wireless). Our social IoT model is based on social relationships among humans who are owners of IoT devices. Every IoT device has an owner who could have many IoT devices. Social relationships between owners is translated into social relationships between IoT devices as follows: Each owner has a list of friends (i.e., other owners), representing its social relationships. This friendship list varies dynamically as an owner makes or denies other owners as friends. If the owners of two nodes are friends, then it is likely they will be cooperative with each other. A device may be carried or operated by its owner in certain community-interest environments (e.g., work vs. home or a social club). Nodes belonging to a similar set of communities likely share similar interests or capabilities [2].

B. Threat 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 [19-24] and is not addressed in this paper. In the context of service-oriented IoT environments, we are concerned with trust-related attacks that can disrupt the trust system. Bad-mouthing and ballot-stuffing attacks are the most common forms of reputation attacks. Self-promoting and opportunistic service attacks are the most common forms of attacks based on self-interest. Thus, a malicious IoT device (because its owner is malicious) 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 a service provider (SP), but then can provide bad or malfunctioned service.
2. Bad-mouthing attacks: it can ruin the reputation of a well-behaved device (by providing bad recommendations against it) so as to decrease the chance of that good device being selected as a SP. This is a form of collusion attacks, i.e., it can collaborate with other bad nodes to ruin the reputation of a good node.
3. Ballot-stuffing attacks: it can boost the reputation of a malicious node (by providing good recommendations) so as to increase the chance of that bad device being selected as a SP. This is a form of collusion attacks, i.e., it can collaborate with other bad nodes to boost the reputation of each other.
4. Opportunistic service attacks: it 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.

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

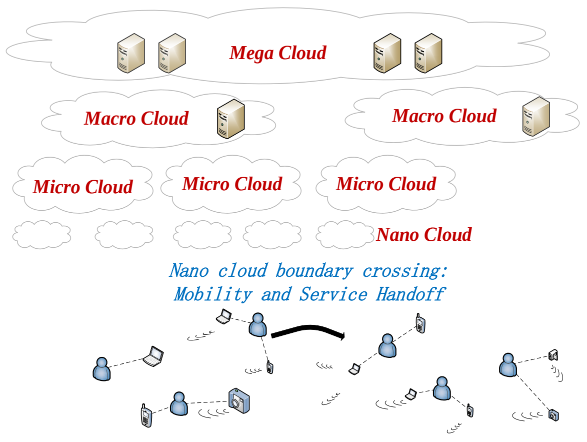

Fig. 1: Hierarchical Cloud Architecture.

III. A Hierarchical Cloud Architecture for Integrated Mobility, Service, and Trust Management

Fig. 1 illustrates the proposed cloud hierarchy. We label the clouds from top to bottom as mega, macro, micro, and nano clouds. The nano and micro clouds can be base stations and routers owned by network operators, while macro and mega clouds can be mini and big data centers owned by cloud service providers. Each nano 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.

A. *Mobility Management*

Mobility management is the basic functionality of a mobile network. Since the bottom levels are base stations and routers in a mobile network, the location information (or the region a user is in) is maintained automatically as an added benefit. Specifically, following the concept of using a home location server for tracking the location of a mobile node in a mobile network [25-28], we track the location of an IoT device using a family of "home" clouds, starting from the home nano cloud (base station) at the bottom layer, home micro cloud (router) at the second bottom layer, home macro cloud at the second top layer, and home mega 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 [25-28]. When an IoT device moves from one region to another region (if the IoT device is mobile), a "mobility handoff" ensues by which the nano cloud which the IoT device just roams into will inform all home clouds of the IoT device of the new location. When a caller IoT device wants to locate a callee IoT, it simply sends a query to the least common "home" cloud of the caller IoT device and the callee IoT device and this least common "home" cloud will return the callee IoT's current location that was recorded in its database. Essentially, the location information is known to the system all the time, thus facilitating location-based service to be invoked by the cloud as IoT devices move from one region to another. One cloud application that can benefit from this capability is participatory sensing [3] allowing a huge amount of location-based sensing information to be collected by IoT devices (e.g., smart phones, public transportation vehicles, city utility poles, etc.) and analyzed in the cloud for situation awareness and decision making.

B. *Service Management*

Service management is the basic functionality of cloud computing service. Nano and micro clouds in the bottom layers of the cloud hierarchy augment macro and mega clouds in the top layers to collectively provide cloud service to IoT devices. An IoT device will only interact with its current nano cloud for service invocation to minimize energy consumption and service latency. The current nano cloud will examine the service request to decide if it can process locally. If yes, it will complete the request-response cycle locally without disturbing the upper layer clouds. This is true if the request is just to do computation offloading, or if the service request only requires local service data to answer the service request. If the service request is to report new service data such as a feedback or a sensing outcome, the current nano cloud will follow the *store-process-forward* procedure. i.e., it will store a replicated copy [29, 30] 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 involves several IoT devices some of which are not under the current nano cloud, then the nano 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. If the service request is a query regarding a target IoT device, then the current nano 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. Then it will wait for a reply to return to it after which it will forward the reply to the requesting IoT device. Last, when an IoT device moves from one region to another region, a "service handoff" ensues. Among other things, the IoT device's Virtual Machine (VM) [31-33] is migrated from the old nano cloud to the new nano cloud so as to maintain service continuity. A forwarding link is connected between the old nano cloud and the new nano cloud, so if the old nano cloud receives a reply, it will forward to the new nano cloud.

C. *Trust Management as a TaaS Cloud Utility*

Trust management of IoT devices is of fundamental importance for any service-oriented IoT application that must incorporate feedbacks or recommendations for decision making because one must identify trustworthy raters or recommenders and apply filtering mechanisms to filter out untrustworthy feedbacks before data analysis and decision making. With the cloud architecture, trust management can be realized as a TaaS cloud utility using standard store-process-forward and forward-wait-reply procedures described earlier.

An IoT device can report user satisfaction in the range of [0, 1] toward another IoT device who just completed a service of a specific service type (e.g., sensing noise or sensing air pollution) to its current nano cloud. The nano cloud upon receiving a user satisfaction report would follow the standard store-process-forward procedure described earlier.

An IoT device (on behalf of its owner) can simply query its current nano cloud about the trustworthiness of a target IoT device for providing service of a specific service type. The current nano 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 reports in its local store and apply a trust protocol 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 nano cloud forwarding the query who in turn will forward the response to the requesting IoT device for decision making.

D. *Reconfigurability, Fault Tolerance, and Resiliency*

Failure recovery is critical for service-oriented IoT applications [34, 35]. The proposed hierarchical cloud architecture is highly reconfigurable and fault tolerant to failures. When a nano cloud fails, one of its neighbor nano clouds can take over its duty to continue with the ongoing services. This may involve restarting the VM process originally running on the failed nano cloud. When a micro cloud fails, its function can be covered by all nano clouds under it, since the service data stored in the failed micro cloud can be recovered from all service data stored in all nano clouds under it based on the store-process-forward procedure. The same fault tolerance process is applied hierarchically up.

The proposed hierarchical cloud architecture is also highly resilient to network disconnection. When answering a query involving a requesting IoT device and a target IoT device, the nano cloud needs to find the least common home cloud by following the standard forward-wait-reply procedure. If the last common home cloud cannot be found because of disconnection, then it can always go to the mega cloud which is the common home cloud of all IoT devices, at the expense of energy consumption and service latency. If the nano cloud cannot find any home cloud because of a total disconnection, then it can attempt to answer the query using the service data stored locally. The accuracy of the reply largely depends on the amount of service data for the target IoT device stored locally in the nano cloud. If the target IoT device does not move often and stay in the nano cloud over the recent past, then the nano cloud would store most of the service data of the target IoT device (because nearly all reports would come to this nano cloud) which would be sufficient for accurately answering the query. This robust response strategy improves resiliency against disconnection to maintain service continuity.

IV. Evaluation

We demonstrate the effectiveness of the proposed hierarchical cloud architecture by applying it to an air pollution detection IoT application. While the proposed cloud hierarchy can be used for integrated mobility, service, and trust management, we will only focus on demonstrating its effectiveness on trust management via the TaaS cloud utility discussed in Section III.C.

The example air pollution detection IoT application for performance evaluation is taken from [13] as follows: A smart city IoT application running on Alice tries to avoid stepping into high air pollution areas (in terms of the levels of carbon dioxide, PM10, etc.) for health reasons. Alice's smartphone is a member of the air pollution awareness social network. She decides to invoke her smartphone to connect to sensor devices in an area she is about to step (or drive) into. Alice knows that many IoT devices will respond, so she needs to make a decision on which sensing results to take. She instructs her smartphone to accept results only from 5 most "trustworthy" sensors and she will follow a trust-weighted majority voting result. That is, each yes or no recommendation is counted as 1 weighted by Alice's trust toward the recommender. If the total trust-weighted "yes" score is higher than the total trust-weighted "no" score, Alice will step into the area; otherwise, she will make a detour to avoid the area. This smart city air pollution detection application is essentially a simple trust-based IoT application in which Alice will select 5 IoT devices for which she trusts the most. Therefore, the trustworthiness score (or utility score) of this service application which it aims to maximize is simply the average of the 5 individual trustworthiness scores Alice has toward the 5 IoT devices. We use the application utility score as a metric for performance evaluation. We also use the percentage of malicious nodes selected for service execution as an additional performance metric.

We compare our proposed TaaS cloud utility with existing non-scalable distributed IoT management protocols including EigenTrust [16], PeerTrust [17], ServiceTrust [18], and Adaptive IoT Trust [2] for this smart city air pollution detection application. For fair comparison, the environment is setup as in [2]. Specifically, we simulate 2000 IoT devices roaming in 16x16 regions. These IoT devices are randomly assigned to 500 users (including Alice) connected in a social network represented by a friendship matrix and a community of interest (CoI) matrix. These users move according to the small world in motion (SWIM) mobility model [36] modeling human social behaviors. Nano, micro, macro, and mega clouds are setup following the cloud hierarchy illustrated in Fig. 1, with each cloud type serving 1, 4, 16, and 256 distinct regions, respectively. Trust data collection and trust computation are performed as in [2]. Specifically, when two users encounter each other they exchange friendship and CoI social information (in a privacy preserving way [2]) so as to measure the social similarity of each other. Each user passes social similarity information to its nano home cloud following the standard store-process-forward procedure described earlier in Section III.C. A user after completing a service experience toward an IoT device reports its service experience (positive or negative) to its nano home cloud via the standard store-process-forward procedure. A user wishing to know the trust status of an IoT device simply sends a query to its nano home cloud via the standard forward-wait-reply procedure.

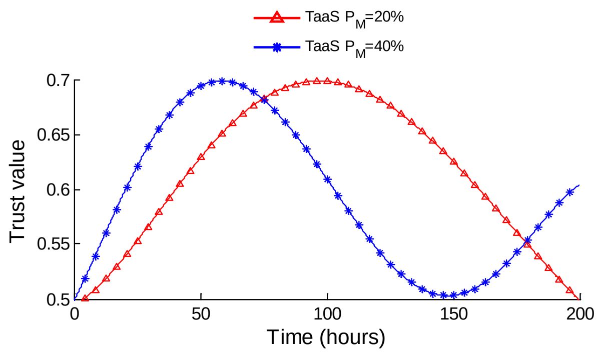

Fig. 2: Trust Value of a Malicious Node under varying P_M.

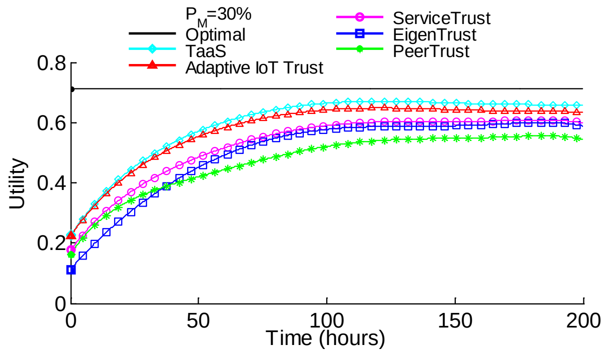

Fig. 3: Utility Score of the Smart City Air Pollution Detection Application.

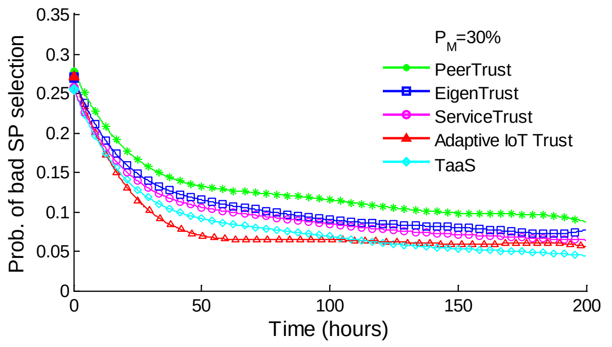

Fig. 4: Probability of Bad SP Selection for the Smart City Air Pollution Detection Application.

Fig. 2 demonstrates the trust convergence, accuracy and resiliency properties of the TaaS cloud utility built on top of the proposed cloud hierarchy. It displays the trust value of a "malicious" node that performs attacks as described in Section II.B. Among all attacks, this malicious node performs *discriminatory attacks* based on its social relationships towards other nodes, and *opportunistic service attacks* with the high trust threshold being 0.7 and the low trust threshold being 0.5. Specifically, the malicious node provides good service to gain high reputation opportunistically when it senses its reputation drops below 0.5. Once it reputation rises to 0.7, it provides bad service again. We see from Fig. 2 that TaaS is able to accurately track trust fluctuation of the malicious node performing opportunistic service attacks. We observe that the rate of trust fluctuation is higher when the percentage of bad nodes (P_M) is higher because more malicious nodes can collude to quickly bring the trust level of this malicious node to 0.7.

Fig. 3 shows performance comparison results with the percentage of malicious nodes P_M set at 30%. We observe that TaaS (cyan line) upon convergence approaches the performance of "optimal" service composition (black line) based on ground truth. Further, although all trust protocols achieve convergence, TaaS outperforms EigenTrust, PeerTrust, and ServiceTrust for trust-based service composition. TaaS also consistently performs better than Adaptive IoT Trust [2] due to its ability to effectively aggregate trust evidence from all nodes in the system through the simple store-process-forward and forward-wait-reply cloud computing paradigms to achieve high trust accuracy.

Fig. 4 shows the percentage of malicious nodes selected for the smart city air pollution detection application. TaaS performs comparably with Adaptive IoT Trust, both of which outperform EigenTrust, PeerTrust, or ServiceTrust by a significant margin nearly cut in half in the percentage of bad nodes selected. Moreover, TaaS outperforms Adaptive IoT Trust as time progresses because TaaS can leverage cloud service to aggregate broad evidence from all nodes having interaction experiences with a target IoT device.

V. Conclusion

In this paper we proposed a hierarchical cloud architecture with moderately powerful base stations and routers (owned by mobile network operators) in the lower layers and more powerful mini and big data centers (owned by cloud service providers) at the higher layers for supporting integrated mobility, service, and trust management of service-oriented Internet of Things (IoT). To demonstrate the utility, we developed a *Trust as a Service* (TaaS) cloud utility leveraging the hierarchical cloud architecture and showed that TaaS can outperform existing non-scalable distributed IoT management protocols because it can leverage simple yet powerful store-process-forward and forward-wait-reply cloud computing paradigms to aggregate broad evidence from all nodes having interaction experiences with a target IoT device. In the future, we plan to further validate the proposed hierarchical cloud architecture with real-world service-oriented IoT applications that can fully explore the benefit of integrated mobility, service and trust management, such as those discussed in [13, 33].

Acknowledgements

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

References

[1] A. Botta, W. de Donato, V. Persico, and A. Pescape, "Integration of Cloud Computing and Internet of Things: A Survey," *Future Generation Computer Systems*, vol. 56, 2016, pp. 684-700.

[2] I.R. Chen, J. Guo, and F. Bao, "Trust Management for SOA-based IoT and Its Application to Service Composition," *IEEE Trans. on Service Computing*, vol. 9, no. 3, 2016, pp. 482-495.

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

[4] H. Mousa, et al. "Trust management and reputation systems in mobile participatory sensing applications: A survey" *Computer Networks*, vol. 90, 2015, pp. 49-73.

[5] J.H. Cho, A. Swami, and I.R. Chen, "Modeling and analysis of trust management with trust chain optimization in mobile ad hoc networks" *Network and Computer Applications*, vol. 35, 2012, pp. 1001-1012.

[6] J. H. Cho, A. Swami, and I. R. Chen, "Modeling and Analysis of Trust Management for Cognitive Mission-driven Group Communication Systems in Mobile Ad Hoc Networks," *Inter. Conf. Computational Science and Engineering*, Vancouver, Canada, 2009, pp. 641-650.

[7] Y. Wang, et al., "CATrust: Context-Aware Trust Management for Service-Oriented Ad Hoc Networks," *IEEE Transactions on Services Computing*, 2016, in press.

[8] K. Govindan and P. Mohapatra, "Trust Computations and Trust Dynamics in Mobile Ad Hoc Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, 2012, pp. 279-298.

[9] F. Bao, et al., "Scalable, Adaptive and Survivable Trust Management for Community of Interest Based Internet of Things Systems," *11th IEEE International Symposium on Autonomous Decentralized Systems* (ISADS 2013), Mexico City, March 2013.

[10] 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.

[11] I.R. Chen, F. Bao, M. Chang, and J.H. Cho, "Dynamic Trust Management for Delay Tolerant Networks and Its Application to Secure Routing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, 2014, pp. 1200-1210.

[12] G.C. Chen, 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.

[13] I.R. Chen, F. Bao, and J. Guo, "Trust-based Service Management for Social Internet of Things Systems," *IEEE Trans. on Dependable and Secure Computing*, 2016, in press.

[14] M. Nitti, R. Girau, and L. Atzori, "Trustworthiness Management in the Social Internet of Things," *IEEE Trans. Knowledge and Data Management*, vol. 26, no. 5, 2014, pp. 1253-1266.

[15] 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.

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

[17] 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.

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

[19] R. Mitchell and I. R. Chen, "Adaptive Intrusion Detection of Malicious Unmanned Air Vehicles using Behavior Rule Specifications," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 44, no. 5, 2014, pp. 594-604

[20] H. Al-Hamadi and I.R. Chen, "Adaptive Network Management for Countering Smart Attack and Selective Capture in Wireless Sensor Networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 3, 2015, pp. 451-466.

[21] A.P.R. daSilva et al., "Decentralized Intrusion Detection in Wireless Sensor Networks," *ACM 1st Workshop on Quality of Service and Security in Wireless and Mobile Networks*, Montreal, Quebec, Canada, 2005.

[22] R. Mitchell and I. R. Chen, "Behavior-Rule Based Intrusion Detection Systems for Safety Critical Smart Grid Applications," *IEEE Trans. Smart Grid*, vol. 4, no. 3, pp. 1254-1263, 2013.

[23] R. Mitchell and I.R. Chen, "A survey of intrusion detection in wireless network applications," *Computer Communications*, vol. 42, 2014, pp. 1-23.

[24] H. Al-Hamadi and I.R. Chen, "Redundancy Management of Multipath Routing for Intrusion Tolerance in Heterogeneous Wireless Sensor Networks," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, 2013, pp. 189-203.

[25] 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.

[26] 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," *Proceedings of the 36th annual symposium on Simulation*, 2003, pp. 65-72.

[27] B. Gu and I. R. Chen, "Performance analysis of location-aware mobile service proxies for reducing network cost in personal communication systems," *Mobile Networks and Applications*, vol. 10, no. 4, 2005, pp. 453-463.

[28] 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.

[29] 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.

[30] 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, pp. 1-13, 2003.

[31] Y. Wang, I.R. Chen, and D.C. Wang, "A Survey of Mobile Cloud Computing Applications: Perspectives and Challenges," *Wireless Personal Communications*, vol. 80, no. 4, 2015, pp. 1607-1623.

[32] M. Satyanarayanan, et al., "The Role of Cloudlets in Hostile Environments," *IEEE Pervasive Computing*, Oct. 2013, pp. 40-49.

[33] A.U.R. Khan et al., "A Survey of Mobile Cloud Computing Application Models," *IEEE Communications Surveys & Tutorials*, Vol. 16, No. 1, Nov. 2014, pp. 393-413.

[34] 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.

[35] 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, pp. 115-122, 2005.

[36] S. Kosta, A. Mei, and J. Stefa, "Small World in Motion (SWIM): Modeling Communities in Ad-Hoc Mobile Networking," *7th IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, Boston, MA, USA, 2010.