5-1-2015

A Multi-Level Trust Management Scheme for the Internet of Things

Anusha Ramanathan
University of Nevada, Las Vegas, ramana1@unlv.nevada.edu

A MULTI-LEVEL TRUST MANAGEMENT SCHEME FOR THE INTERNET OF THINGS

By

Anusha Ramanathan

Bachelor of Science, Information Technology
P.S.G. College of Technology, India
2008

Master of Computer Application
P.S.G. College of Technology, India
2011

A thesis submitted in the partial fulfillment of the requirements for the

Master of Science in Computer Science

School of Computer Science
Howard R. Hughes College of Engineering
The Graduate College

University of Nevada, Las Vegas
May 2015

UNLV | GRADUATE COLLEGE

We recommend the thesis prepared under our supervision by

Anusha Ramanthan

entitled

A Multi-Level Trust Management Scheme for the Internet of Things

is approved in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science
Department of Computer Science

Yoohwan Kim, Ph.D., Committee Chair

Juyeon Jo, Ph.D., Committee Member

Ajoy K. Datta, Ph.D., Committee Member

Venkatesan Muthukumar , Ph.D., Graduate College Representative

Kathryn Hausbeck Korgan, Ph.D., Interim Dean of the Graduate College

May 2015

Abstract

The significance of the Internet of Things (IoT) in current trends is continuously rising. It is an umbrella term that signifies a network of physical devices that are embedded with electronics, software, sensors and connectivity that enable greater functions and services through the exchange of data accomplished through interconnection. The applications of the IoT are varied and numerous; they range from relatively simple home automation scenarios to the much more complex scenarios of interconnected smart cities. IoT is expected to dominate the future with huge amounts of content oriented traffic that is a result of intensive interactions between the millions of devices that will be available by then. The rising popularity of IoT has been accompanied by a corresponding rise in the number of issues. One of the issues is a lack of an established mechanism that deals with the issue of trust management. This issue is well addressed in the field of wireless sensor networks; an analogous framework for trust management does not exist for IoT. The complexity of the networked devices (allied with the complexity of the network itself) in addition to the fact that the environment in which the devices exist is itself continuously changing makes the development of a trust management scheme difficult.

We propose a trust management scheme that helps establish trust between devices taking into account the nature, complexity and category of the interconnected devices. The level of service available to a node that requests a service from a service provider is predicated upon the trust level between the provider and requester. We elaborate on this concept and describe the emergence of trust over time that is also sensitive to the changing environment to which the devices might be subjected.

Acknowledgments

First and foremost I express my deepest gratitude to my advisor, Dr. Yoohwan Kim for his excellent guidance and assistance throughout my thesis. This thesis would not have been possible without his ideas, advice and persistent help.

I would also like to thank Dr. Ajoy K. Datta, Dr. Juyeon Jo and Dr.Venkatesan Muthukumar for their support. It is an honor to have them in my thesis committee. I am grateful to my parents and my brother for their warm support in every phase of my life. Lastly, appreciation goes out to my husband for his encouragement, support and motivation to complete my thesis.

Anusha Ramanathan

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction to the Internet of Things

The vast majority of the devices that are connected to the Internet today are used directly by humans. But a new trend has arrived which has introduced devices that are connected to Internet and are smart enough to accomplish tasks in an autonomous manner without any human intervention. These devices range in complexity from simpler RFID tags and sensors to complex networks of interconnected devices, which are in turn managed by other smart devices leading to smart cities. The Internet of Things is a technological revolution that represents the future of computing and communications, and its development needs the support from some innovational technologies [1]. As an emerging technology, the Internet of Things (IoT) is expected to offer promising solutions to transform the operations and roles of many existing systems such as transportation systems, manufacturing systems etc. [2]. In [4], smart community, an Internet of Things application, which is collection of cooperating objects where smart homes can interact with each other to help implement concepts such as a neighborhood watch and pervasive healthcare. Such applications are demonstrate that the Internet of Things can be interconnected and to work together to create a smarter world. According to Cisco, the number of IoT devices exceeded the population of humans in 2008 and is projected to reach 50 billion by 2020 [3]. Therefore, it is important to salvage information from IoT devices and maximize the efficacy of the same by connecting it with other devices. This is a direct inference from Metcalfe's Law, which states that the value of a network is proportional to the square of the number of devices in it. Various applications

and services of IoT have been emerging into markets in a variety of areas, e.g., surveillance, health care, security, transport, food safety and distant object monitoring and control [6]. Large corporations, such as IBM and Microsoft have recognized the potential inherent in IoT and conduct significant research in the area.

IoT is going to create a world where physical objects are seamlessly integrated into information networks in order to provide advanced and intelligent services for human beings. The ubiquity of interconnected "things" such as stand-alone sensors, sensors attached to mobile devices, mobile devices themselves lead to collection of massive amounts of data about human social interactions. These data can be further aggregated, fused, processed, analyzed and mined in order to extract useful actionable information to provide complex and intelligent services.

1.1 Definition

Internet of Things is formally defined as "Interconnection of sensing and actuating devices providing the ability to share information across platforms through a unified framework, developing a common operating picture for enabling innovative applications."[5].

According to [7], Internet of Things can be defined as "Dynamic global network infrastructure with self configuring capabilities based on standard and interoperable communication protocols where physical and virtual "things" have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network.

‘Things’ are active participants in business, information and social processes where they are enabled to interact and communicate among themselves and with the environment by exchanging data and information sensed about the environment, while reacting autonomously to the real/physical world events and influencing it by running processes that trigger actions and create services with or without direct human intervention”.

IoT forms the basis for the next big leap in technological evolution; exploiting the ubiquity of heterogeneous technological devices to achieve complex intelligent function. IoT helps evolve the currently homogenous nature of the Internet to a mote heterogeneous, fully integrated version. Just as the Internet revolutionized the connectivity of people, similarly IoT seeks to transform the world into a smarter world where devices would be capable of carrying out autonomous interaction to achieve composition of simpler services to be able to achieve richer functionality.

There are several interpretations of IoT. A combination of differing perspectives is provided by Atzori et. al. [12] .

The three visions are:

- Things Oriented Vision
- Internet Oriented Vision
- Semantic Oriented Vision

Things Oriented Vision focuses on hardware components such as RFID, wireless sensors and actuators, Near Field Communication (NFC), smart objects, etc. Internet Oriented Vision focuses on IPSO (Internet for smart objects), web of things, etc. Semantic Oriented Vision focuses on semantic technologies, reasoning over data, etc. [12].

Although the three visions appear to be somewhat distinct, they are in actuality interconnected. Some of the devices used in a particular area will be also used in another and the concepts described also intersect. Fig 1.1 explains this phenomenon that there is an intersection between the different visions of IoT. This also explains why there are so many interpretations for IoT and the emergent nebulousness when we seek to define it.

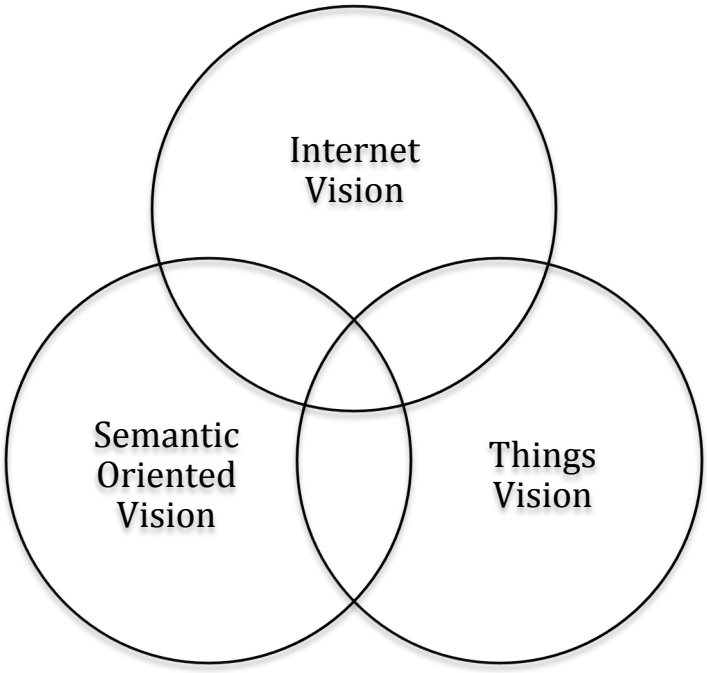

Fig 1.1 Three visions of IoT paradigm

1.2 History of IoT

There were several concepts and areas that were related to IoT which were around before 1992. Kevin Ashton coined the name Internet of Things. Kevin Ashton has stated "I could be wrong, but I'm fairly sure the phrase 'Internet of Things'started life as the title of a presentation I made at Procter & Gamble (P&G) in 1999"[8]. In 1999, Auto-ID Labs

founded by Kevin Ashton, David Brock and Sanjay Sarma was founded. They helped develop the Electronic Product Code (EPC), a global Radio Frequency Identification (RFID)-based item identification system that was mainly aimed at replacing the Universal Product Code (UPC)[9]. Auto-ID refers to identification technologies used in the industry to automate, reduce errors and increases the efficiency of the whole system [5]. In 2003, the EPC symposium that took place in Chicago, Illinois marked the launch of the first EPC network. It was an open technology that allowed computers to automatically identify objects and track them as they move from production plant to their distribution centers [5]. The symposium also highlighted RFID as the key technology for economic growth in next 50 years.

In 2005, a team of students and faculty members led by Massimo Banzi at the Interaction Design Institute Ivrea located in Ivrea, Italy developed the Arduino, a single-board micro controller [9]. Also in 2005, the United Nations first mentions the Internet of Things in an International Telecommunications Union report [9].

After this point in time, several technologies and protocols have been developed in the field of IoT. With these advancements IoT is moving towards enhancing the utilization of technology throughout the world.

1.3 Application of IoT

The Internet of Things is the expansion of the current Internet services so as to accommodate each and every object, which exists in this world or is likely to exist in the coming future [10]. Due to the infinite possibilities to accommodate and combine discrete

systems to create higher-level function, IoT has widespread applications. A few of the applications are listed below.

1.3.1 Smart environment

IoT can be used to create smart homes where the thermostat can change temperature based on your requirements, room lighting is adjusted based on the time of day, etc. Discrete systems in independent housing units can communicate with the respective systems throughout the community to reduce the overall energy footprint [4].

1.3.2 Healthcare

The application of IoT in healthcare can change several aspects of the way the healthcare industry functions. It can help identify the patient by means of a RFID tag. IoT devices are very helpful in assimilating healthcare information from a patient.

Wearable IoT is one the main areas where IoT can be helpful. Through wearable IoT, individuals are seamlessly tracked by wearable sensors for personalized health and wellness information—body vital parameters, physical activity, behaviors, and other critical parameters impacting quality of daily life [11].

1.3.3 Transportation

IoT can forever change the way we travel. Cars, trains, and buses along with the roads and the rails equipped with sensors, actuators and processing power may provide important information to the driver and/or passengers of a car to allow better navigation and safety [12]. We can use IoT to design cars that has self-driving capabilities. For example, Google cars uses sensors and other processing systems to access the conditions of the road and read the area. IoT can provide better logistics for the supply chain of any product from its plant to the store.

1.3.4 Security

IoT can be used in several security related applications. Use of sensors to prevent invasion into property has been one of its most popular applications. IoT products have been used for preventing invasion, automatic property management, real-time alarms, etc. [13]. It has been used in several commercial products like Cox home security, for home security.

1.4 Technologies used in IoT

Taking a very broad view of IoT [27], we can classify the whole IoT network into:

- Things
- Local networks
- Internet
- Servers

1.4.1 Things

A thing, in the Internet of Things, can be a person with a heart monitor implant, a sensor in middle of a farm or a thermal sensor at home. They may be as generic as tiny devices, which can be assigned an address and connected to a network. They may have the ability to transfer the information to a master node. The data collected from the nodes is processed by the master node or in turn forwarded to a server or other devices for processing or generating actionable information. Things occupy the lowest level of the hierarchy and are one of the main components in the whole network.

1.4.2 Local Networks

These are the medium by which the nodes are connected each other and to the Internet. They provide the gateway that helps them interact with each other and to available master nodes (or the internet). There are several technologies used to provide this network. Some of the main technologies used for this are:

- WiFi
- Zigbee
- EnOcean [42]
- 6LoWPAN [43]
- Bluetooth

The choice of technologies affects the way the network is configured and also affects the way how the devices communicate. This in turn affects the lifetime of the devices, the range at which the communication takes place, etc. For example, WiFi needs a lot of power and hence the devices exhaust their energy in a shorter period of time whereas the devices in Zigbee and EnOcean are designed in such a way that the battery can last for a longer time. Zigbee and EnOcean use the 802.15.4 [44] protocol for communication. Thus the choice of technologies affects several factors.

1.4.3 Servers

These are the central authorities for the whole network. As the smaller devices do not have many capabilities and are therefore incapable of performing non-trivial computations, there is a need for a central structure to process all the data. This is also the logical center of the network; operations that can be performed as a whole over the network are defined and executed here. For example, in a home security system if a

sensor detects a possible fire at home, the data is transmitted to the main monitoring network and the personnel at this server can help by calling the appropriate services for help. The server is also helpful for pushing updates into the systems so that vulnerabilities can be reduced. Fig 1.2 provides a sketch of the components in the system.

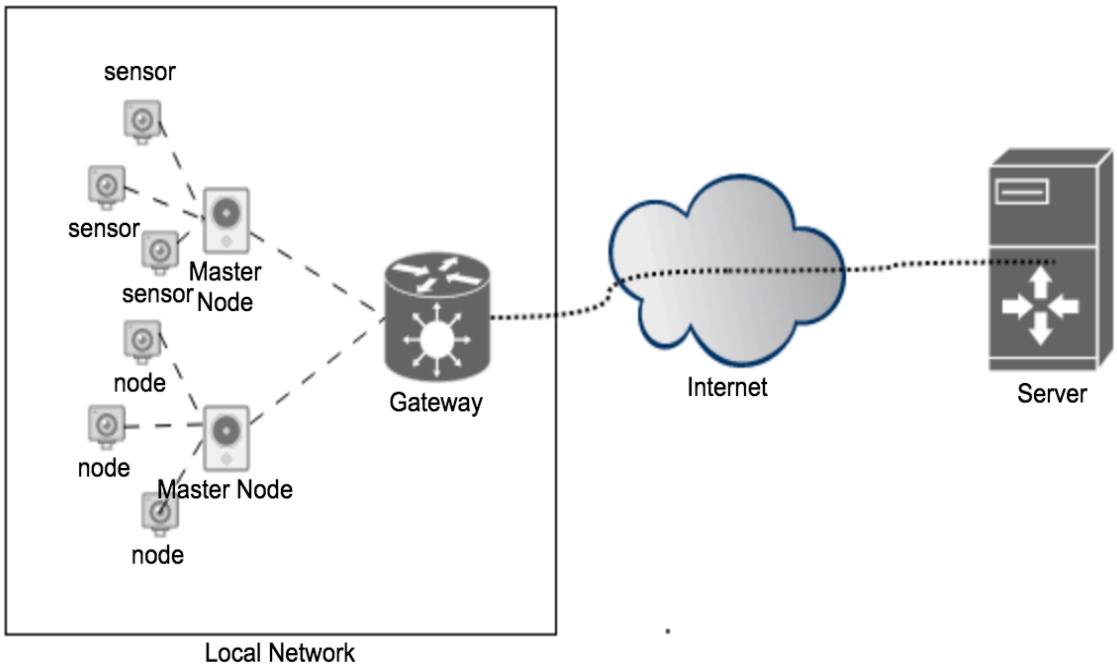

Fig 1.2 Components of an IoT system

1.5 Challenges of IoT

Although there is so much flexibility intrinsic to the structure of IoT and myriad applications of the same, deployment on a large scale brings forth its own set of issues. The following is a small list of challenges in IoT:

- Security is an enemy of complexity. When more things are connected and more information is available readily the door for security vulnerabilities is always open.

- Different manufacturers use different standards to create their own devices. Hence Standardization of Technology is very important as this can help in interoperability of the devices and help them to gather more data before it is synthesized into information.
- IoT can produce copious amounts of data from even the simplest interactions. Hence we need to find ways to store the data reliably, process it and synthesize meaningful information from the same.
- Complex IoT applications can consist of several hundred lines of code to carry out defined functions. These may then have to be interpreted / compiled; that requires the existence of libraries and tool chains in place. While the code one writes may be unit tested, there is no surefire way to verify the good behavior of other code on which the written code depends. In reality systems do arrive with bugs and patches to fix them. Therefore it is difficult to conduct real-world testing under laboratory conditions.
- Concepts of Privacy, confidentiality and anonymity while retaining powerful functionality and preventing leakage of information are an enormous challenge, yet to be fully solved.
- Power efficiency is of paramount importance. A device whose power consumption is inefficient cannot be made a pivotal part of any non-trivial system and cannot be assigned critical tasks. This means that trust management algorithms and security mechanisms need to be fast, secure and draw very little energy from the power source.

Chapter 2

Introduction to Trust

Trust is an important aspect of any application. It helps to define the security policy, who is authorized to perform what actions, how to mange security applications, the techniques to implement the applications.

2.1 Definition

There are several definitions of *trust* in the literature:

The Oxford Dictionary [16] trust defines as “firm belief in the reliability, truth, ability, or strength of someone or something”. According to [17], trust can be defined as “the measurement of the belief from a trusting party point of view (trustor) with respect to a trusted party (trustee) focused on a specific trust aspect that possibly implies a benefit or a risk.”

2.2 Trust in Internet

The adoption of Internet-based services has been on the rise and is expected to continue rising. Web-based access to information and interaction through mails and equivalent mediums has made requirement of trust in Internet important. For example in an internet e-commerce transaction customers must believe that sellers will provide the services they claim to and the information (like credit card number, social security numbers, etc.) provided to them will not be revealed to external third-party sources or

abused/leaked in any way. The level of trust customers have on specific sellers will help them in their choice of determining a safe, trustable seller. Reciprocally, the seller must also believe that the customer will furnish payment for the items purchased or services rendered. This example states a basic concept of a level of trust in day-to-day life vis-a-vis Internet e-commerce transactions.

Trust is a directional relationship between two parties that can be called trustor (giver) and trustee (receiver). In [15], different forms of trust are classified based on

2.2.1 Access to a Trustor's Resource

The trustor trusts a trustee to use the resources that he/she owns or controls. For example, allowing an actor to read a file is indicative of a certain amount of trust placed in that actor. Whereas allowing an actor to execute a program typically shows a correspondingly higher level of trust in the actor.

2.2.2 Provision of Service by the Trustee

The trustor trusts the trustee to provide a service that does not involve access to the trustor's resources.

2.2.3 Certification of Trustees

This type of trust is based on certification of the trustworthiness of the trustee by a third party, so trust would be based on a criteria relating to the set of certificates presented by the trustee to the trustor. For example, a certificate provided by a website from a CA such as VeriSign each time when one accesses a website.

2.2.4 Delegation

A trustor trusts a trustee to make decisions on its behalf, with respect to a resource or service that the trustor owns or controls.

2.2.5 Infrastructure Trust

This refers to the base infrastructure that the trustor must trust. He must trust himself (implicit trust). He should be able to trust his workstation, local network and local servers, which may implement security or other services in order to protect his infrastructure. For example, trusting all the hardware provided with a particular certification.

2.3 What does trust actually mean?

In [15] Tyrone et al. have said that trust is a vast topic that incorporates trust establishment, trust management and security concerns. They have mentioned that there exists no consensus definition of the concept of trust. The word "trust" is used interchangeably with trust, authentication and authorization [15]. The precise meaning has to be inferred from the context in which the concept is applied. Authorization can be a decision that has been arrived at due to a trust we have on someone/something. For example, we can provide access rights to our computer to someone whom we trust. Traditionally, authentication can be defined as process of determining whether someone or something is, in fact, who or what they (it) claim to be. This could be accomplished by verifying credentials such as ID cards, passwords etc.

Tyrone et al. [15] define trust as "the firm belief in the competence of an entity to act dependably, securely and reliably within a specified context" (assuming dependability covers reliability and timeliness).

2.4 Trust Management

Blaze et al. defined trust management as “a unified approach to specifying and interpreting security policies, credentials, relationships which allow direct authorization of security-critical actions”[18].

Trust management is defined as “The activity of creating systems and methods that allow relying parties to make assessments and decisions regarding the dependability of potential transactions involving risk, and that also allow players and system owners to increase and correctly represent the reliability of themselves and their systems”[19].

2.5 Use of Trust Management

Trust management helps us to manage trust between several entities by providing the methods to define the trust between the entities. They provide the methods to help the entities to determine the trustworthiness of the other entity through an automated mechanism. This mechanism can be based on the decision taken by the entity or decision taken based on the information from other entities. The entities can collaborate and transfer the trust among one another to arrive at a more informed decision. Similar to the scenario surrounding the definition of trust, there are several interpretations of the concepts that constitute Trust Management. One of the first implementations of Trust Management is the PolicyMaker trust management system [18]; which is based on the approach that trust management should be decoupled from the needs of a particular product or service. KeyNote [46] appeared as an improvement in PolicyMaker. REFEREE [45] is yet another trust management system that is based on PolicyMaker.

2.6 Attacks on Trust Management

A Trust Management system is mainly aimed at providing (computing) a trust score or some other metric, which informs the decision made by a node on whether to provide/utilize services from another node. There exist several varieties of attacks that are designed to specifically break this service. A malicious node in the network could execute these attacks so to achieve a variety of malicious ends; it could boost its own reputation to gain access to higher functions in the system or generally be disruptive in a manner that brings down the overall efficiency of the system. The following are some of the common attacks that are executed against trust management systems by malicious nodes.

2.6.1. Self-promoting attacks

It can exaggerate its importance (by providing bogus good recommendations for itself) so as to be selected as the service provider, but then stop providing service or provide malfunction service [23]. This lowers the quality of service provided by the entire network.

2.6.2. Bad-mouthing attacks

It can ruin the reputation of well-behaved nodes (by providing bad recommendations against good nodes) so as to decrease the chances of good nodes being selected as service providers [23].

2.6.3. Good mouthing attacks

It can boost the reputation of bad nodes (by providing good recommendations for them) so as to increase the chances of bad nodes being selected as service providers [23].

2.6.4. On-off Attack

A malicious node executing an on-off attack would exhibit a pattern of behavior that alternates between behaving well and behaving badly, hoping that it can remain inconspicuous even while causing damage [29].

2.6.5. Conflicting Behavior Attack

Malicious entities can impair the reputation (trust values) of good nodes by intentionally reporting different values to different nodes for the node in question [29].

2.6.6. Sybil Attack

A malicious node can create fake identifiers for nodes that share or even take the blame, which should instead be given to the malicious node [29].

2.6.7. Newcomer Attack

A malicious node removes its bad history by registering as a new user [29].

The above categories are not exhaustive; it is evident that there exist a lot of ways trust management systems may be attacked. Several works have been carried out to mitigate against such attacks.

Chapter 3

Trust Management in IoT

IoT is composed of a variety of devices from several discrete and disparate domains; hence it is necessary to develop a trust management scheme that the devices may utilize to interact and exchange information. Without a trust management system in place the systems may never be able to leverage a sound decision making process to enable them to communicate appropriately; in this scenario only the extremes of not communicating at all or communicating all the time are available; both of which are disastrous and counter-productive. For example, consider a refrigeration system that is used to store perishable goods. The temperature of this system can be controlled by assimilating and synthesizing measurements from temperature sensors and managing those systems through IoT. Consider a malicious node in the system, this node can transmit incorrect or malicious information system-wide and hence cause the temperature to change inappropriately. In IoT, its highly imperative to know to whom we must trust and whom we should not trust. It is possible that the emerging importance of IoT could be hindered by the emergent and sometimes inherent security problems. Collaboration between the nodes also helps in the following scenario: if a battery-depleted node is in the network and once a node identifies it, the node can inform other nodes about this. Collaboration per se may indeed open the way to a new class of attacks, all the more insidious as they involve internal attackers. A cooperating node owning legitimate cryptographic keys can easily launch an internal attack inside the group by altering data or injecting bogus information without being identified [20].

This area of trust management in IoT is not extensively studied. The unique nature of IoT makes it difficult to define trust management schemes for IoT.

3.1 Literature review

This chapter focuses on previous work carried out that pertains to trust management systems for IoT. Although there exist several works that have been carried out on the issue of trust management in wireless sensor networks these proposals were not designed with the existence of IoT in mind. The increased complexity of the system with regards to IoT makes it more difficult to use the same system here. In the literature, Roman et al. [30] pointed out that the traditional approaches for security, trust, and privacy management face difficulties when applying to IoT systems due to scalability and a high cardinality of relationship types among IoT entities.

There are several studies, which have been done specifically in this area of trust management of IoT. As stated above, these are not nearly as numerous as the ones in the area of wireless sensor networks. This chapter explores some of those works. First we begin with the concept of Social Internet of Things (SIoT) [22].

3.1.1 Social Internet of Things (SIoT)

SIoT [22] overlays the concepts of social networks on to IoT; it is intended as a social network where every node is an object capable of establishing social relationships with other things (nodes) in an autonomous way according to the rules set by the owner. SIoT provides middleware, application functionality and protocols to ease the exploitation of things-related services. SIoT mainly aims at establishing and then

exploiting social relationships among things. Things use this relationship to crawl the IoT and discover services and resources.

According to this model [28], a set of forms of socialization among objects is foreseen. The parental object relationship is defined among similar objects, built in the same period by the same manufacturer (the analog to the role of family is derived from the production batch). Moreover, objects can establish co-location object relationships and co-work object relationships, like humans do when they share personal (e.g., cohabitation) or public (e.g., work) experiences. A further type of relationship is defined for objects owned by the same user (mobile phones, game consoles, etc.) that is classified as ownership object relationship. The last relationship is established when objects come into contact, sporadically or continuously, for reasons purely related to relations among their owners (e.g., devices/sensors belonging to friends); it is named the social object relationship. These relationships are created and updated on the basis of the objects features (such as: object type, computational power, mobility capabilities, brand) and activity (frequency in meeting the other objects, mainly).

The previous work in [22] is short on certain implementation details; it does not describe how the social relationships should be established by objects and does not provide any information regarding the necessary architectural configuration and the appropriate list of protocols to be deployed to achieve this. To this end in [31], Atzori et al. described how a SIoT could be implemented. They provide architectural details for the system by separating the entities into three layers namely a sensing layer, network layer and an application layer. They provide further details on how the functionalities may be implemented so that the end results form complete social networks. Functionalities such

as service discovery (finding objects that provides a service), service composition (enabling interaction between objects to achieve a particular service), owner control (definition of role of an object), profiling (configuring information about the objects) are defined. Although [31] details all the types of information collected which go into a trustworthiness management system that makes a decision, it does not provide any detail on how it should be implemented. The works in [31] discuss several approaches to achieve the functionality, but no concrete implementation details are proposed.

3.1.2. Trustworthiness Management in SIoT

In [25, 28] Nitti et al. provided a trustworthiness model for the SIoT. They have proposed two models for defining the trustworthiness of the system. Both of these models use some shared basic abstractions such as a feedback system, concepts of centrality of a node, etc. to calculate the trust score. These models are:

3.1.2.1.Subjective trustworthiness

In [28], each node computes the trustworthiness of its friends on the basis of its own experience and on the basis of that of its friends. If they are not friends, then the trustworthiness is calculated by word of mouth through a chain of friendships. The relationships, which are defined in SIoT, are each given a particular score and the model uses this score (among other factors) to calculate the trust value. The trust score calculation also uses a short-term opinion and a long-term opinion about a node to provide more accuracy in the calculation of the trust score. In the case where there is more than one degree of separation, the node assigns a feedback value to its adjacent nodes along each of the paths to the provider. The same assignment operation is then performed by all the nodes that are present along the path to the provider, unless a node

with a low credibility is found (in this case the process is interrupted).

3.1.2.2. Objective trustworthiness

In [29], they use the Chord [32] protocol to store the information. The values, which are required to provide the trust score, are distributed. It uses DHT (Distributed Hash Table) to achieve this. A DHT is an abstract data structure, which uses keys to retrieve data. Each node is responsible for a set of keys. They define an overlay network that connects the nodes and helps to find the owner of a given key in a key space. The keys are generated using SHA-1 with Chord and the data is sent to the particular node that is responsible for storing this information. To prevent malicious nodes from being chosen as nodes for storing this information and consequently cause a state of confusion in the whole system, the information is visible to every node but is only managed by special nodes that are called Pre-Trusted Objects (PTOs). The data that is stored in these nodes represents feedback and trustworthiness values.

Both the models use the feedback from a node to calculate the trust score. The feedback system provides an evaluation of the service that it received from a particular provider. The drawback of using a feedback system is that it leads to a lot of traffic throughout the entire network. For every service that has been provided a provider node waits to get back a response evaluation. The service requester also has an additional overhead of sending a response for the service it has received. IoT devices have limited battery life and overhead such as this could deplete the battery life at a greater than sustainable rate.

3.1.3 Trust Management for the IoT and Its Application to Service Composition

In [21], the authors focus on 3 main parameters used to arrive at a numerical

quantity to denote trust. They are honesty, cooperativeness and community-interest. Honesty is defined as the trust property that is used to determine whether a particular node is deemed to be honest. The cooperativeness property represents the extent to which the nodes are socially cooperative. The community interest explains whether the trustor and trustee are in the same social community. They devise their social relationship from [22] which references SIoT where relationships are established between IoT devices based on their social relations such as being manufactured by same company, active in a particular area etc.

Two nodes that participate in an interaction directly or as part of a larger activity are able to observe each other and update their trust assessments. They also exchange their trust evaluation results toward other nodes as recommendations. The aforementioned triplet of parameters is used in calculating a trust value. If a node i has a direct link to node j then trust is calculated based on the trust value for these three parameters between node i and node j. If there is no direct link between node i and node j, then recommendation system is leveraged to obtain a recommendation from its neighboring node. The recommending node will provide the values for all three parameters to make the assessment. The equation includes a temporal component that favors more recent measurements over older ones.

[21] did not address issues pertaining to scalability as applied to large networks and the lack of concrete implementation level data structures or abstractions that may be used to store and retrieve the information. It only addresses the theoretical framework of the system and the implementation has yet to be formally declared.

3.1.4 Scalable, Adaptive and Survivable Trust Management for Community of Interest Based IoT Systems

This paper is extension of work done in [21]. In [23], a Community of Interest (CoI) based trust management scheme is defined. The paper defines inter-CoI and intra-CoI measures for the trust management system. This paper references SIoT [22] for defining the trust management scheme. The work assumes that two nodes belonging to the same CoI have specific social interests and strong social ties whereas different communities may have different or controversial views of trust towards the same trustee due to their differing social interests. The goal of the proposed trust management scheme is to make sure that each node's trust evaluation converges to its community agreement (also referred to as CoI ground truth).

It makes use of the same parameters that were defined in [21], namely honesty, cooperativeness and community-interest. The trust calculations are similar to the ones already described in [21]. The underlying idea of the trust protocol is a Bayesian reputation system [34] where each node calculates the trust using Bayesian estimation techniques over historical observations. In [22] more work has been done to explain about the storage mechanism in every node.

The following rules are applied when a decision has to be made about storing a trust value. A particular node can store only a limited set of trust values. For a storage size of n, when a slot is needed, for a node trust value to be retained, it must be in top Ω (e.g. 50%) of the n trust values, or the node is one that has interacted recently.

The following conditions are applied when a node i computes the trust between itself and node j:

1. If the storage space is not full or node i does have the trust information pertaining to node j in its storage space, node i will simply save the computed trust value towards node j.

2. If the storage space is full and node i does not have the trust information pertinent to node j in its storage space, node i will store the computed trust value towards node j and pop out the trust value computed for the earliest interacting node with a trust value that is below the median trust value.

Although the paper addresses scalability issues, it is very computation intensive. As we have seen before performing excessive computations may be okay for a certain device in IoT, but not all devices. Also it assumes that a CoI will have same social interest, which may not be true always.

3.1.5 Trust Management for Service Composition in SOA-based IoT Systems

In [33] Chen et al. propose a trust management scheme for service composition in a SOA-based system. They have used the previously defined social parameters such as friendship, social contact and community of interest (CoI). They have also developed a distributed collaborative filtering technique to collate the feedback from owners sharing similar social interests. Each user maintains a profile, which is a collection of lists: friends list (list of current friends), CoI list (list of devices directly interacted with) and a location list (locations frequently visited for social contact) and trust value list. In addition to these lists, a user experience list is also maintained. The model assumes that

each user will have at least one high-end device on which intensive computations can be performed and data can also be maintained. All the devices of a particular user can share the lists maintained in this high-end device.

When the devices of two users have direct interactions, they can exchange their profiles and provide trust recommendations. The protocol first measures its social similarity with a recommender in friendship, social contact and CoI and then decides if the recommendation is trustable.

In [33] it assumes the availability of a high-end device for every user, which cannot be guaranteed in every network; let alone across the heterogeneity of IoT.

3.1.6 Trust Management mechanism for IoT

In [24], a trust management mechanism for IoT has been proposed. In order for the author to be able to define a trust mechanism, he proposes that the whole of IoT be divided into three layers, namely: sensor, core and application. The sensor layer holds the physical devices and their base station. The core layer consists of networking devices and the Internet. This layer is responsible for the interconnection and routing of information. The main job of the application layer is to process and store the data.

In [24], in each layer a trust mechanism with different parameters specific to the layer is used. Accordingly the parameters are tightly coupled to the layer that they describe. The trust scheme uses fuzzy set theory to arrive at the set of results (trust information) in each layer. The trust information collected is subjected to a decision making process. The overall decision is made based on the trust information collected, in addition to any policies that are specific to the service requester.

3.1.7 Trust Management system design for the IoT: A context-aware and multi-service approach

The main objective of [26] is to manage cooperation in a heterogeneous IoT architecture involving nodes with different resource capabilities, in order to establish a community of trusted elements assisting one another with respect to the operation of a set of collaborative services. A centralized approach for the trust management system is used so that different trust management servers might handle the wide range of reports in that area.

During the bootstrapping period the system collects information about the entire network. This is presented in terms of reports, which are stored in the trust manager and used as inputs for the trust management system. When a request is received from the node the trust manager performs entity selection and returns a set of trustworthy nodes to the requester. This selection process goes through several criteria such as restriction of proxies, restriction of reports, computation of weights for remaining reports, computation of trust values for each proxy, etc. The client node now performs an assessment on the set of nodes it received from the trust manager. Finally it sends a reply with a positive or negative value for a node. The type of assessment used to get at this value depends on the type of service the node has requested. The final step in the process is the learning phase where based on the quality of recommendation the reputation of recommending nodes is increased or decreased.

3.1.8 Related work

In [35] Chen et al. propose a security architecture, which has four layers, namely the data perception layer, network access layer, data management layer and intelligent

layers. The security measures, which can be used in each of the layers, are explored in detail. A feasible trust based suggestion was finally made.

In [36] Chen et al proposed a trust management model to provide cooperation between things in a network based on their behaviors. This model aims at providing a similar behavior and fuzzy-theory based trust and reputation model for sensor nodes or sensor embedded nodes, where every node develops a direct reputation for every other node by making direct observations and indirect reputation based on recommendations. They have oriented their design based on a very specific IoT environment, which consists of wireless sensors only and have evaluated their packet forwarding service. This model needs further investigation for all other types of devices in an IoT environment.

In [37] the author proposes the use of a framework called SecKit [38], which is a Model-based Security Toolkit to address security aspects of distributed systems. There are several configurable options that might be leveraged to enable us to setup the network securely.

In [39], users' trustworthiness in social networks is used to assist service composition between objects. [6] provides a survey of various trust management ideas on IoT. It classifies all the work done in this area into different categories based on some parameters like Trust relationship and decision, Data perception trust, etc.

In [40] Guinart et al. proposed a SOA-based IoT architecture where devices offer their functionalities via SOAP-based web services or RESTful API. This is helpful in discovery, query selection and on demand provisioning of web services. In [41] Zhou and Chao proposed media-aware traffic security architecture for IoT. However, their approach did not discuss the issue of scalability.

Chapter 4

Multi-level Trust Management

As explained in the previous chapter a majority of the trust management schemes for IoT have either been based either on SIoT or on dividing IoT into several layers based on certain criteria. SIoT concepts help in the development of relationships between the things. These relationships might be based on locality, ownership or a variety of other factors, however the category of relationships is sometimes too broad; in certain cases it is preferable that the relationships be more granular and the select ability criterion is more stringent. We intend to interconnect smart objects in an autonomous manner; this requires the infusion of a little more information into the system. In [29] SIoT has been proposed: it describes how the concept of a system that is based on IoT could be implemented. This involves several phases like the entry of a new node, service discovery and composition, newly created co-location relationships and service provisioning. One of the most important components of service composition is the concept of trust management. At this point in the overall process, all the information that has been collected from other phases needs to be summarized, synthesized and based on this synthesis a decision has to be made about whether the particular system is to be trusted. Trust management is particularly helpful in this regard at this juncture.

When a new node attempts to connect to a network to request a service (or to become a provider) it is necessary that a decision be made on whether the new node can be trusted to join. There is probability that the node is not malicious and assimilating it into the network allows for a more complex function to be achieved; equally probable however is that the node is a malicious one and assimilating it would be counter-

productive or even inimical.

Consider a scenario where we need to create a network with all nodes in the neighborhood or in the locality to achieve a specific objective. A typical example would be to inter-network the home monitoring systems in a selected geographical area to be able to implement an autonomous neighborhood watch. A node that is already a part of a home automation system is a node that would have a high trust value in its own network; however while attempting to join an external network (another home automation system), it is a new (external) node that is attempting to join the combined system. The information that is to be transmitted to achieve this unification has not been seen before and it is also infrequent (possibly even one-time). At this juncture, the information available in the network is not sufficient to be able to base a sound conclusion about whether the new node is to be trusted. Now, we may need to query information that is available with other home security systems (if they have interacted with this one) or query a node that is a component in the same network as the node that may have a record of previous interaction and is attempting to join. This describes the concept of indirect information or indirectly communicated information.

4.1 Purpose

The aim of this paper is to propose and develop an improved trust management mechanism for IoT. Currently IoT devices are organized in closed, self-contained networks; the overall information is available, but not accessible; therefore we are unable to utilize it to arrive at a decision. SIoT enumerates a few scenarios about that describe the procedural aspects of achieving interconnection of devices from separate networks

and concepts on information passing between them to be able to achieve higher function. The maximal utilization of IoT can only be achieved when the constraints are removed and true interoperability is achieved. It is to this end that we propose a trust management scheme that would help these disparate devices to establish communication with each other with a scheme that borders on complete autonomy and bereft of human intervention. Each of the devices has to have access to information in the form of a list of capabilities that would allow it to decide whether it would be able to provide a particular to service to a particular requester (or not as the case may be). Let us consider the following example: if we were to be staying at a hotel and desired to know the ambient temperature in the hotel lobby area. This could be accomplished by sending a request to a temperature sensor that is located in the lobby. At this juncture, the onus is now transferred on to the temperature sensor in the lobby; if the sensor deems that the request is from a legitimate and non-malicious source, it would provide the information requested. If, however, the temperature sensor's evaluation deems the requesting node to be malicious, it would instead choose to not provide the information requested. This can be achieved if and only if the temperature sensor is intelligent in an a-priori manner or it is able to compute the necessary variables that are required to make this decision.

SIoT focuses mainly on a declarative approach to the set of services that a node could provide and by polling other available service providers in the region, a list of services that can be obtained. This model however could be enhanced with the addition of more granularity to achieve a more responsive system. The information that is made available may not be sufficient if it only addresses the services (as a whole) at a broader level; it may sometimes be as important to be able to infer that it provides perhaps exactly

the level of service the requester is looking for and no more or even restrict the service interface based on a level of trust. Hence we need a notion of an escalating level of services that is somewhat more granular, which requires that additional information is available at the time when a relationship is made.

When a decision has to be made, it is ideal that the information is directly available. However, in most cases, direct information would be unavailable and we would have to resort to basing decisions on information gathered from indirect interactions or inferred information. It is also important to know the nature of the indirect interaction to be able to decide its applicability to the current decision. For example, a new node that enters a system could request for a connection to printing services or even to a particular printer, if this is known ahead of time. In this case, the printer either already knows of the new node directly (in which case there would exist past values assigned to past interactions) or more likely, that it does not know about the new node. The printer would then request other nodes in the same network for information about any past interactions with the new (arriving) node. It is possible that the other nodes do have a record of past interaction. In this case, it could even be from a node that is from a slightly different type of device. For e.g. a thermal sensor in the area may have interacted with this new node and deemed it trustworthy. However, this information is of limited importance when deciding whether this node should be given access to printing services. It is entirely possible that the new node may be malicious, but it exhibited good behavior whilst collecting information related to the temperature and intends to exhibit malicious behavior when engaging print services. In other words information from different categories may not translate perfectly. If, however the request for information returns a value from another printer, this

information is considered far more relevant since it is from the same category of device. This necessitates a weighing scheme that assigns weights to information obtained from differing sources that should assign a higher weight for information that is considered more relevant. The process of accomplishing this scheme further necessitates a notion of categories of information that need to be assigned the appropriate nodes. In our example, if the only sources of information were a thermal sensor and a printer, the printer's information would be assigned a much higher weight.

Expanding on our scenario of a hotel lobby, the steady state of the system would consist of the appearance of a variable number of new nodes at sporadic intervals. A policy of denying service to these nodes on the basis that their computed trust value is low due to lack of previously available actionable information would not be satisfactory. It would result in dissatisfaction and underutilization. Therefore it is prudent to tie the levels of service available to the trust value that is computed. This enables a very basic set of services to be made available, while reserving higher function for nodes that have a higher trust value. This would result in far fewer nodes being turned away without service. In our example, if a new unseen node appears and requests for printing services; the printer would then assimilate available information (in this case since it is previously unseen, even the indirectly aggregated information would be scarce/unavailable) and compute a trust value that is low. In this case we just do not have enough information to be able to make an informed decision. If the node is turned away, it is an unsatisfactory scenario. So, we would make available only a very basic level of service that is available at that particular level of trust: e.g. permit only 5 pages to be printer per hour.

In this way, the level of service availability is contingent on the computed trust

value for the node. The evolution in this scenario takes place as described. The node is now known and has behaved in a non-malicious manner for an hour and availed printing services for an hour. The next time this node requests a service; we have the previously recorded information detailing good behavior. This would make it eligible for a higher trust value and a higher level of service. Each good interaction contributes to a higher trust score on subsequent requests, which in turn unlocks access to a higher level of service. In our example after sufficiently demonstrated good behavior, we may permit the node to request services that allow for 10 pages to be printer per hour, and so on.

The main ideas in our proposed scheme are:

a) Introduce a weighing factor that is tied to, and accurately represents the category information about the provider of the information.

b) Provide service levels that increase in function based on the computed trust value for the requesting node.

4.2 Proposed Scheme

4.2.1 The Model

Let $X = \{X_0, X_1, X_2, \ldots\ldots, X_n\}$ represent the set of things in the network and $S = \{S_0, S_1, S_2, \ldots . S_m\}$ be the set of services each of the things can provide. The number of services provided varies depending on the type and nature of the thing. For example, X_0 can provide services $\{S_0, S_1, S_2\}$ whereas X_1 can provide the set of services $\{S_1, S_2, S_3\}$.

The system is composed of service requesters and service providers; these are roles rather than designations (a service provider for one node is quite capable of being a service requester in other interactions). A service requester role is played by a thing that

requests a particular service from the network (or another thing). A service provider is the other side of this interaction; it provides a node the requested service after performing the necessary trust computations. Each thing publicly announces the set of services (and their levels) that it can provide. When a node needs a particular service it sends a request to the particular thing requesting for that service.

The trust management protocol proposed here is event driven; it performs the computation only on the receipt of a request for service or in other selected circumstances, such as trying to find a relationship using owner relationship (as explained in SIoT). The particular node now computes and consolidates the trust value and decides the course of action.

Hence the main steps involved in this process are:

1. Request for Service: The event that triggers the process.
2. Computation of trust value: This is aggregated from values derived from the available results of direct and indirect communications. The category of the device that provides the information is a part of the calculation as well.
3. Decision on offering (a level of) service: This is the step where the service provider X_i evaluates trust with respect to the service requester X_j and makes a decision on the level of service that will be offered.

Each service level has a threshold H_j^l, which is the minimum trust value that is required of a requester for a service provider to be able to provide that service at level l. Also a particular service also has a value $S_j^{add_l}$, which represents the quantum added to the trust value for every operation at level l. Any malicious operation will result in a negative $S_j^{add_l}$ value whereas every positive experience will result in a positive $S_j^{add_l}$

value. $S_j^{add_l}$ is used during the calculation of trust value in step 2. So every service level can be represented as the tuple $S^l{}_j = \{H_j^l, S_j^{add_l}\}$.

Each thing can be classified into a category $C = \{C_0, C_1, C_2, \ldots., C_l\}$. The category is an analogue for the capabilities and ratings of the device. This classification is carried out using several parameters like memory capacity, type of device, etc. Less formally it can be thought of as an analogy for a class of devices. For example, the total set of black and white printers occupies a category whereas the set of all color printers occupies a different category.

4.2.2 Trust Value Calculation

Let us consider a sample network with 4 nodes.

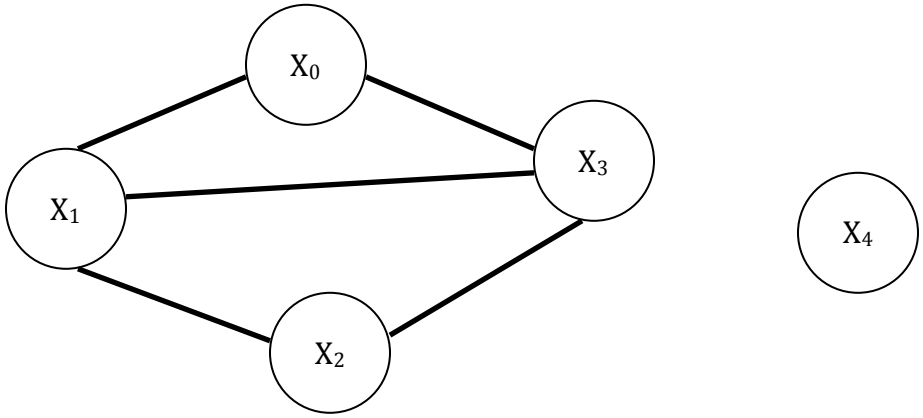

Fig. 4.1 Sample Network

Fig. 4.1 provides a sample scenario with devices X_0, X_1, X_2, X_3, X_4. The devices X_0, X_1, X_2, X_3 form a network and X_4 is a new node that requests a particular service from X_3. Each node maintains a list of services that it is able to provide. Let us consider that X_3 is able to provide the set of services(including level information) denoted by $S_1^1, S_1^1, S_2^1, S_2^2, S_2^3, S_3^1$ services. X_3 would then maintain a couple of data abstractions like the ones shown below in Table 4.1 and Table 4.2.

Services S_j (provided by X_3)	Thresholds (for service S_j at level l) H_j^l	Ratings Quantum (success) $S_j^{add_l}$	Ratings Quantum (malice) $S_j^{add_l}$
S_1^1	0	0.0010	-0.0015
S_1^2	0.10	0.0015	-0.0025
S_2^1	0.20	0.0025	-0.0040
S_2^2	0.40	0.0050	-0.0070
S_2^3	0.60	0.0075	-0.0095
S_3^1	0.75	0.0500	-0.0600

Table 4.1 Services Declaration Table

Table 4.1 is a data structure abstraction that describes the declarative interface exposed by the node enumerating the services at the configured service levels it is able to perform. Here H_j^l is the minimum trust value required for the new node X_4 to successfully request and obtain the particular service S_j at service level l. For example, if X_4 requires the service need S_2 service at level 3 it needs the provider X_3 to have computed the trust value for X_4 above the threshold (0.5) or else the request will be denied.

The trust value between a node a to node b is calculated using the following formula:

$$T_{ab} = \gamma\, D_{ab} + (1 - \gamma) I_{ab} \qquad (1)$$

Here γ represents the weighting factor that is used in the calculation. It is a measure of how much more weight a direct reading D_{ab} carries over an indirect one I_{ab}.

It is envisioned that the direct rating be weighted heavier as we expect that the best estimate for a probabilistic evaluation of node behavior is obtained from direct interactions rather than a transitively propagated value. The trust value computed is asymmetric; that is $T_{ab} \neq T_{ba}$.

When a new node requests an existing device for a particular service, the device attempts to perform the trust calculations for the direct and indirect components. To this end, it checks its own history of interactions to determine if this particular node is truly new (i.e. it has never interacted before) or whether it has indeed interacted previously. If no information is available about previous interactions with the particular node (D_{ab}) then it asks its neighbors about the node. In our scenario, X_3 would query the nodes X_0, X_1, X_2. This constitutes the indirect/transitive trust magnitude. If the neighboring nodes also do not have any actionable information about the new node then a default value T_{ab}=0 (that is $T_{X_3X_4}$= 0) is assigned. The trust value starts out with a zero magnitude and is increased/decreased accordingly with the values $S_j^{add_l}$ defined in Table 4.1.

Node (X_i)	Trust Value (T_{V_i})	Last Updated Time (epoch)
X_0	0.250	1426980525000
X_1	0.440	1426155525000
X_2	0.170	1424000525000

Table 4.2 Trust Table

Table 4.2 contains the trust values that each of the nodes in the network maintain about each other. The time that the last trust computation was updated is also stored. The direct trust value D_{ab} between a node a and node b at time t for a service S_j at service level l is formulated as

$$D_{ab}(t) = \alpha\, D_{ab}(t_{prev}) + S_j^{add_l} \qquad (2)$$

That is, if node a has requested service S_j at service level l [denoted as S_j^l], from node b then $S_j^{add_l}$ is the value that successfully modifies the previously stored trust value for this interaction. $S_j^{add_l}$ is explained earlier along with the service. $D_{ab}(t_{prev})$ is the trust value between node a and node b which is already stored in the table of node a. α is a parameter that describes a mathematical decay over time. If new trust value is calculated at time t and previously existing trust value that is already stored in the table of node a is taken at time t_{prev} then

$$\alpha = \frac{t - t_{prev}}{kx}\ ; k \geq 1$$

where x is a constant value that is proportional to the passage of time. For example, if x is defined as 5 minutes (or 300 seconds) and k periods of length x have elapsed then then α is suitably modified. This means that a rating was calculated within the past 5 minutes, we re-use the rating. If a rating is calculated as of ten minutes ago , it will be suitably reduced in magnitude , for we do not have any information about the behavior of the requester for that time interval.

If there are a number of successful interactions the calculated value D_{ab} reaches a pre-defined maximum value D_{max} ; if the trust level reaches this value, the requester has

unlocked access to the next higher level of service. If all the relevant maximums have been reached, all the services are made available.

Indirect trust value between node a and node b can be calculated as

$$I_{ab} = \sum_{i=1}^{n} T_{ai} w_c TV_i \beta$$

Here T_{ai} denotes the trust value of node a with respect to node i. The overall computed indirect trust value is a summation of the trust values for a node that is obtained from each of the nodes that has interacted with it previously modified appropriately by the trust value between the node that is requesting the trust values and the node providing the trust values. The technique of summation avoids any one particular term from exercising too much influence of the final computed value. For example, if node a does not trust node i then the T_{ai} would be 0; so any value that is provided by node becomes 0. This is equivalent to discarding this particular contribution to the summation. The idea behind this is to reduce the efficacy of both bad mouthing and good mouthing attacks wherein a malicious node reports low trust values for trustworthy nodes and high trust values for known malicious nodes respectively.

The quantity denoted by T_{V_i} is a measure of the trust value provided by node i about the node b to node a.

β denotes the time-decay factor that modifies T_{V_i}. This not the same as the previously defined time-decay factor α ; it's role however is analogous as it modifies T_{V_i} in a similar manner that α modifies $D_{ab}(t_{prev})$. The definition of β is naturally, similar to that of α.

$$\beta = \frac{t - t_{prev}}{kx} \; ; k \geq 1$$

where t, t_{prev} and k represent the same quantities but modify the analogous parameters above.

4.2.3 Trust Category

We augment our equations to include the previous proposed notion of categories. Categories, as we recall are numbers that are assigned keeping in mind the computation capabilities, energy consumption metrics and similarity to other devices etc. The rationale behind assigning a category to a node (device) is derived from the principle that we would choose to assign a higher probability to the trust value emanating from a device that is of a similar category to the provider than from a device that is from a dissimilar category. Put simply, it means we would have a higher confidence in the value of trust that has been computed by a device that is most similar to us. To elucidate, consider that the case of a printer providing trust information about a particular node to another printer is considered more relevant than if the source of the information were to be an optical sensor. The light sensor, having considerably less computation power than a printer, would not be able to assess a large range of risky behavior, therefore is likely to provide a less accurate value that models the behavior of a node it has seen previously. The printer, on the other hand, due to the relatively enhanced computational capability and the virtue of being in the same category (i.e. printer) is able to assess more relevant parameters before arriving at a trust value. Hence in this scenario a printer is able to provide information of higher quality to another printer. The devices are assigned categories during their initial installation and setup. This can be inferred from a variety of sources (e.g. manufacturer's spec, the complexity of circuit boards, power rating etc.) When a device with higher category provides a trust value it is implicitly assumed to be of a

higher quality than a rating coming from a device in a lower category, if we apply the rationale described above.

Accordingly this is represented as a weighing factor w_c; which is the contributing weight of the node i's measured trust value that is suitably modified by the category of device that node i belongs to. The overall trust is a summation of the sub-components of the equation (1).

There are several factors to consider how this weight could be assigned. We summarize a couple of them below.

- Each category can be assigned with a constant value based on the intelligence of the device.
- The value can be varied based the current situation. For example, the weight can be assigned in a manner that it peaks at the category that the device itself belongs to and gradually falls off in all the other directions. In this method the calculation can be computed in such way that $\sum w_c = 1$.

In the case where the direct information D_{ab} is available, it is weighted appropriately by γ. However, we may not have this information in most of the cases. In such cases, we fall back upon the indirect component of the trust value T_{ab}.

- When a new node b sends a request to node a for a particular service and the node a does not have any information on this new node then it asks all of its neighboring nodes on information about the new node so that it can build up some trust value based on other nodes judgment (appropriately modified by the trust node a has in its neighboring nodes).

- Periodic updates from other nodes. The nodes exchange information in a periodic manner defined by an elapsed time Δt, so that all information collected by the various nodes is used to actively better the trust measurements. For example, the results of attempts to perpetrate an attack on any particular device by a malicious node are propagated; therefore even though all the nodes were not subject to the attack, they are now aware of it and can modify their behavior accordingly.

4.3 Observations

Fig 4.2 is a function of the residual trust value T versus time elapsed since the last measurement t modified by the trust decay factor α. e-10 denotes a modeling where α undergoes radioactive decay with a half life of 10 minutes. Similarly e-5 corresponds to a scenario where the half life is 5 minutes. l-10 and l-5 denote scenarios where α decays linearly with a stepped interval of 10 minutes and 5 minutes respectively subject to a cycle of 60 minutes.

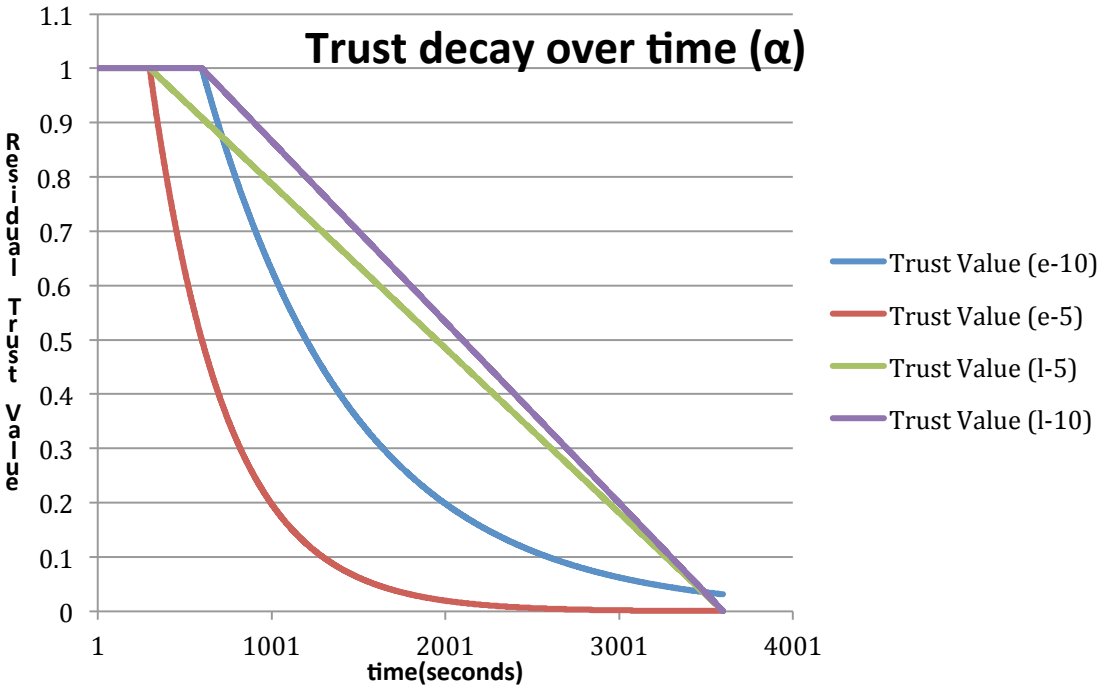

Fig 4.2 Modeling the temporal trust decay factor (α)

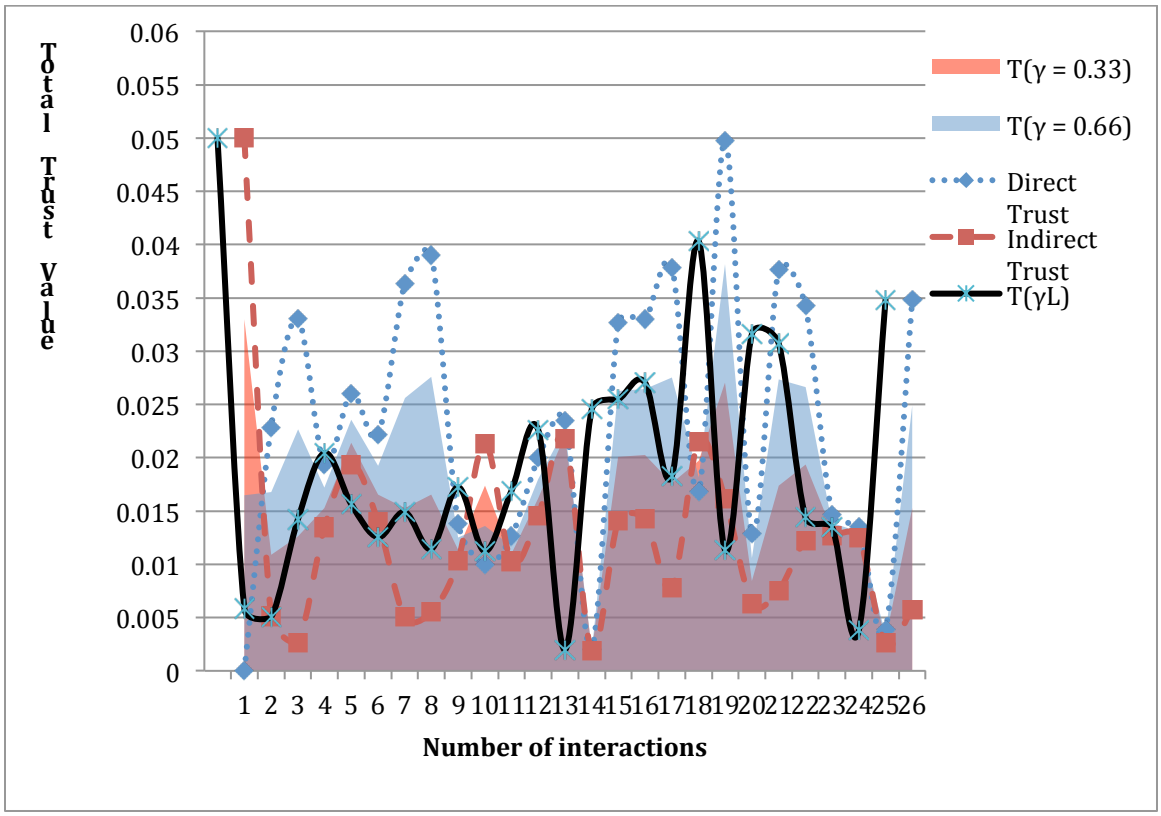

Fig 4.3 Modeling the direct-indirect weighing factor (γ)

Fig 4.3 describes the behavior of the total trust value T as the system evolves over a number of interactions modified by the trust direct-indirect weighing factor γ. The direct and indirect trend lines denote the measured direct trust D_{ab} and the measured indirect trust I_{ab}. The total trust value T is then plotted over a number of interactions for γ values of 0.33 and 0.66. The black trend line $T(\gamma L)$ denotes the case when we parametrically vary the evolution of γ over the number of interactions ; that is at each point the value of γ is a function of the number of interactions seen by the nodes. γ is an input parameter to our model ; therefore we need to simulate a number of scenarios to observe the trust value computed to assign it to a particular scenario.

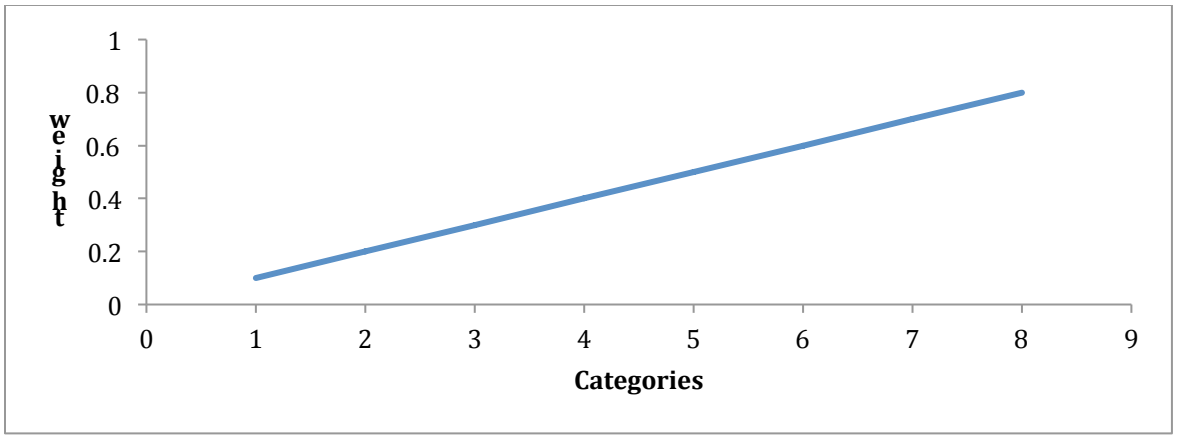

Fig 4.4 Weighing factor w_C (linear)

Fig 4.4 plots the category weighing factor w_C that modifies the reported trust score by the category of the reporting node represented by C. This is a simple assignment function where a higher category results in a higher weight ; the implicit assumption here is that a device in a higher category is more capable of looking at a variety of factors to arrive at a particular trust value for a node and is therefore preferred.

Fig 4.5 plots the category weighing factor w_C that modifies the reported trust score by the category of the reporting node represented by C. This assignment is weighted toward category similarity. For example the weight assigned by a category C_i device to the reported value from a category C_j device depends directly on the category distance between (i, j). When it is zero, the maximal weight is reached; i.e. if $i = j$, a category C_i device assigns the maximal wieght to it; the weight assigned to other categories is tapered accordingly by the distance between the categories.

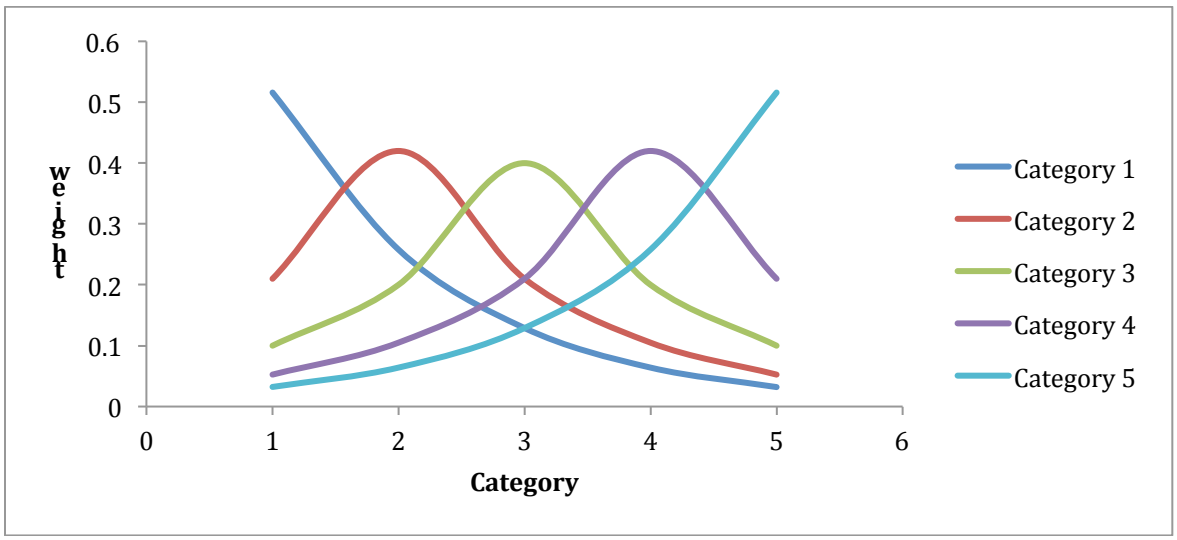

Figure 4.5 Weighing factor w_C (weighted towards category similarity)

Chapter 5

Conclusion and Future Work

This thesis explores the introduction of information about device categories into the process of arriving at a trust value to be used in a trust management scheme. The computed trust value is completely convertible between device categories, with appropriate weights. We have included temporal evolution of the system into the equations as well; which would allow the trust value to be influenced by the frequency of interactions. The quality of the interaction boosts or reduces the trust value accordingly. The equations proposed still contain a couple of parameters that are not known a priori. We could fix these values by extensively simulating the expected scenarios and then plugging the value observed into the model. This information would help achieve better real-world behavior and improve overall efficiency of IoT networks.

References

[1] Lu Tan and Neng Wang, "Future internet: The Internet of Things", in Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on, pp. V5-376-V5-380, 2010.

[2] Li Da Xu, Wu He and Shancang Li, "Internet of Things in Industries: A Survey," Industrial Informatics, IEEE Transactions on, vol. 10, pp. 2233-2243, 2014.

[3] The Internet of Things. http://share.cisco.com/internet-of-things.html. [Accessed on Feb 20,2015]

[4] Xu Li, Rongxing Lu, Xiaohui Liang, Xuemin Shen, Jiming Chen and Xiaodong Lin, "Smart community: an internet of things application," Communications Magazine, IEEE, vol. 49, pp. 68-75, 2011.

[5] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Comput.Syst., vol. 29, pp. 1645-1660, 9. 2013.

[6] Z. Yan, P. Zhang and A.V. Vasilakos, "A survey on trust management for Internet of Things," Journal of Network and Computer Applications, vol. 42, pp. 120-134, 6. 2014.

[7] H. Sundmaeker, P. Guillemin, P. Friess, S. Woelfflé, “Vision and challenges for realising the Internet of Things,” Cluster of European Research Projects on the Internet of Things—CERP IoT, 2010.

[8] Internet of things: http://www.rfidjournal.com/articles/view?4986 [Accessed on Feb 20,2015]

[9] History of Internet of Things http://postscapes.com/internet-of-things-history [Accessed on Feb 20,2015]

[10] D. Singh, G. Tripathi and A.J. Jara, "A survey of Internet-of-Things: Future vision, architecture, challenges and services," in Internet of Things (WF-IoT), 2014 IEEE World Forum on, pp. 287-292, 2014.

[11] S. Hiremath, Geng Yang and K. Mankodiya, "Wearable Internet of Things: Concept, architectural components and promises for person-centered healthcare," in Wireless Mobile Communication and Healthcare (Mobihealth), 2014 EAI 4th International Conference on, pp. 304-307, 2014.

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

[13] Jihong Liu and Li Yang, "Application of Internet of Things in the Community Security Management," in Computational Intelligence, Communication Systems and Networks (CICSyN), 2011 Third International Conference on, pp. 314-318, 2011.

[14] Cox Home Security : http://www.cox.com/residential/homelife.cox [Accessed on Feb 20,2015]

[15] T. Grandison and M. Sloman, "A survey of trust in internet applications," Communications Surveys & Tutorials, IEEE, vol. 3, pp. 2-16, 2000.

[16] Trust definition: http://www.oxforddictionaries.com/us/definition /american_english/trust [Accessed on Feb 20,2015]

[17] Neisse, R.; Wegdam, M.; van Sinderen, M. “Trust Management Support for Context-Aware Service Platforms”,. Book Chapter in User-Centric Networking, Lecture Notes in Social Networks 2014, pp 75-106

[18] M. Blaze, J. Feigenbaum and J. Lacy, "Decentralized trust management," in Security and Privacy, 1996. Proceedings, 1996 IEEE Symposium on, pp. 164-173, 1996.

[19] A. Josang, C. Keser, and T. Dimitrakos. “Can we manage trust?”, in iTrust 2005, Trust Management, Third International Conference, pp. 93-107, 2005.

[20] Y. Ben Saied, A. Olivereau, D. Zeghlache and M. Laurent, "Trust management system design for the Internet of Things: A context-aware and multi-service approach," Comput.Secur., vol. 39, Part B, pp. 351-365, 11. 2013.

[21] Fenye Bao and Ing-Ray Chen, "Trust management for the internet of things and its application to service composition," in World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a, pp. 1-6, 2012.

[22] L. Atzori, A. Iera and G. Morabito, "SIoT: Giving a Social Structure to the Internet of Things," Communications Letters, IEEE, vol. 15, pp. 1193-1195, 2011.

[23] F. Bao, I. Chen and J. Guo, "Scalable, adaptive and survivable trust management for community of interest based Internet of Things systems," in Autonomous Decentralized Systems (ISADS), 2013 IEEE Eleventh International Symposium on, pp. 1-7, 2013.

[24] Gu Lize, Wang Jingpei and Sun Bin, "Trust management mechanism for Internet of Things," Communications, China, vol. 11, pp. 148-156, 2014.

[25] M. Nitti, R. Girau and L. Atzori, "Trustworthiness Management in the Social Internet of Things," Knowledge and Data Engineering, IEEE Transactions on, vol. 26, pp. 1253-1266, 2014.

[26] Fenye Bao and Ing-Ray Chen, "Trust management for the internet of things and its application to service composition," in World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a, pp. 1-6, 2012.

[27] IoT devices: http://micrium.com/iot/devices/ [Accessed March 2,2015]

[28] M. Nitti, R. Girau, L. Atzori, A. Iera and G. Morabito, "A subjective model for trustworthiness evaluation in the social Internet of Things," in Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on, pp. 18-23, 2012.

[29] Y.L. Sun, Zhu Han, Wei Yu and K.J.R. Liu, "A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks," in INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, pp. 1-13, 2006.

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

[31] L. Atzori, A. Iera, G. Morabito and M. Nitti, "The Social Internet of Things (SIoT) – When social networks meet the Internet of Things: Concept,

architecture and network characterization," Computer Networks, vol. 56, pp. 3594-3608, 11/14. 2012.

[32] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for Internet applications,” SIGCOMM Comput. Commun. Rev., vol. 31, no. 4, pp. 149–160, 2001.

[33] I.R. Chen, J. Guo and F. Bao, "Trust Management for SOA-based IoT and Its Application to Service Composition," Services Computing, IEEE Transactions on, vol., pp. 1-1, 2014.

[34] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava, “Reputation-Based Framework for High Integrity Sensor Networks,” ACM Transactions on Sensor Networks, vol. 4, no. 3, May 2008, pp. 1-37.

[35] Dong Chen, Guiran Chang, Lizhong Jin, Xiaodong Ren, Jiajia Li and Fengyun Li, "A Novel Secure Architecture for the Internet of Things," in Genetic and Evolutionary Computing (ICGEC), 2011 Fifth International Conference on, pp. 311-314, 2011.

[36] 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.” Comput. Sci. Inf. Syst., vol. 8, no. 4, pp. 1207–1228, 2011.

[37] I. Kounelis, G. Baldini, R. Neisse, G. Steri, M. Tallacchini and A. Guimaraes Pereira, "Building Trust in the Human?Internet of Things Relationship," Technology and Society Magazine, IEEE, vol. 33, pp. 73-80, 2014 .

[38] R. Neisse, I.N. Fovino, G. Baldini, V. Stavroulaki, P. Vlacheas and R. Giaffreda, "A Model-Based Security Toolkit for the Internet of Things," in Availability,

Reliability and Security (ARES), 2014 Ninth International Conference on, pp. 78-87, 2014.

[39] Yang Liu, Zhikui Chen, Feng Xia, Xiaoning Lv and Fanyu Bu, "A Trust Model Based on Service Classification in Mobile Services," in Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom), pp. 572-577, 2010.

[40] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess and D. Savio, "Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services," Services Computing, IEEE Transactions on, vol. 3, pp. 223-235, 2010.

[41] Liang Zhou and Han-Chieh Chao, "Multimedia traffic security architecture for the internet of things," Network, IEEE, vol. 25, pp. 35-40, 2011.

[42] EnOcean: https://www.enocean.com/en/home/ [Accessed on March 2, 2015]

[43] 6LoWPAN: http://datatracker.ietf.org/wg/6lowpan/charter/ [Accessed on March 2, 2015]

[44] IEEE 802.15.4: http://www.ieee802.org/15/pub/TG4.html [Accessed on March 2, 2015]

[45] Y. Chu, J. Feigenbaum, B. LaMacchia, P. Resnick and M. Strauss, "REFEREE: trust management for Web applications," Computer.Networks ISDN Syst., vol. 29, pp. 953-964, 9. 1997.

[46] Keynote Trust Management System: https://tools.ietf.org/html/rfc2704 [Accessed on March 3,2015]

Curriculum Vitae

Anusha Ramanathan

Degrees:

- Bachelor of Science, Information Technology, 2008

  P.S.G. College of Technology, India

- Master of Computer Application, 2011

  P.S.G. College of Technology, India

- Master of Science, Computer Science, 2015

  University of Nevada, Las Vegas

Thesis title: A multi-level trust management scheme for the Internet of Things

Thesis Committee:

Chair Person: Dr. Yoohwan Kim, Ph.D.

Committee Member: Dr. Juyeon Jo, Ph.D.

Committee Member: Dr. Ajoy K Datta, Ph.D.

Graduate College Representative: Dr. Venkatesan Muthukumar, Ph.D.