Guarantor and Reputation Based Trust Model for Social Internet of Things

Hannan Xiao, Nitin Sidhu, and Bruce Christianson
School of Computer Science
University of Hertfordshire
College Lane, Hatfield, UK
Email: {h.xiao, b.christianson}@herts.ac.uk

***Abstract*— The addition of social networking to Internet of Things (IoT) has led to the paradigm of Social Internet of Things (SIoT). It is where digital devices called objects mimic the social behavior of their human counterparts and build up social relationships with other objects in order to provide services to the humans. One of the obstacles in realizing this idea is trust management. The most complex question in trust building is 'How can objects trust the information provided by other objects?' This paper proposes a new trust model based on Guarantor and Reputation for SIoT. The former requests an object to find a guarantor at an agreed commission rate in order to get service from another object; the latter uses reputation of a node to measure its trustworthiness. By simulating this model using an architecture based on real world scenarios, it was concluded that the trust model can be employed in different scenarios in SIoT and the use of penalties for malicious activity enables the model to detect and isolate malicious nodes.**

Keywords*— *IoT, social IoT (SIoT), trust, reputation, guarantor

I. INTRODUCTION

Internet of Things (IoT) envisages a world where a large number to smart objects, e.g., smartphones, smart cameras, sensors, and RFID, and technologies are interconnected via a unique addressing scheme such as IPv6 and use standard communication protocols to interact with other smart objects and also build up relationships in order to accomplish a common goal [1]-[4]. A group of people tied in a social network can provide far better results than an individual to complex problems, which has been proved scientifically [5]. This property of a group of individuals providing better results can be exploited in the IoT. Several researches have applied the idea of social networking to the IoT arguing that if the IoT can be made to mimic the social behavior of the humans then those smart objects will be able to provide a better service than locally connected objects. This gives rise to a new idea called Social Internet of Things (SIoT) [6]. It can be envisioned as a parallel universe where all the objects like RFIDs, smartphones, GPS and many more held by an individual will communicate with other objects in their vicinity and perhaps globally and make relations with other objects based on their location, capabilities and requirements. SIoT applications can be a valuable resource in numerous areas like domestics, business perspective, SIoT can be helpful in the field of automation and industrial manufacturing, logistics, intelligent transportation of people and goods [1].

978-1-4799-5344-8/15/$31.00 ©2015 IEEE

IoT is considered by the US National Intelligence Council as the one of the top six "disruptive Civil Technologies" with the potential impacts on US national power [4]. However, the things, i.e., the everyday objects also become information security risks and the IoT could distribute those risks far more widely than the Internet has to date [1]. Therefore, in order to tackle the privacy, security and trust issues related to the SIoT, these issues need to be addressed and incorporated in the design and implementation phases of SIoT.

A. Motivations

In the area of trust management in SIoT, work in [6] suggests that an object can trust another object based on the reputation of that object. However, this approach considers that trust is transitive which can be a dangerous assumption as shown by [7], [8]. Another work [9] demonstrates how a guarantor based trust mechanism can make use of local trust relations between a trustee and a trustor to provide an end-to-end guarantor based service. This approach is however impractical when the connection between two objects spans a long distance, then a chain of guarantors will be required which can slow down the communication setup time. If no guarantor is available, then a connection cannot be setup between the objects. Another problem with the guarantor based trust model is that if the service provider is a malicious node, it can still setup a connection with an object requesting the same service and the connection is terminated only after a few transactions have taken place. Transaction ceases when commission and forfeit rate rise beyond the limit of the object requesting the services and the guarantors. Assuming a malicious node just needs only one transaction to insert a malware into another object then this scenario can be devastating for the user.

Motivated by the above research gap, this paper proposes a novel guarantor and reputation based trust model, which uses two parameters, credit (of an object to afford communion to a guarantor) and reputation (that measures the trustworthiness of the object), for trust management and detecting malicious nodes.

B. Related Work

Apart from the reputation-based approach in [6] and the guarantor-based approach in [9] as discussed above, other work in the trust and reputation in IoT are also briefly discussed here. In [10], a trust management protocol was

proposed that considers both social trust and Quality of Service (QoS) trust metrics, using both first-hand observations and second-hand recommendations to update trust. One of the trust properties is cooperativeness, evaluating the degree that a trustee is socially cooperative with a trustor. The definition of the trustor in [10] is different from that in [9] and this paper, where a trustor fully trusts a trustee using local guarantees. Differently, the work in [11] proposed a trust management model using fuzzy logic approach to calculate the reputation for IoT. In a survey on trust management for Internet of Things, the authors proposed a holistic trust management framework of IoT based on IoT's system model, which includes three sectors: Cyberphysical trust relationship, trust management objectives, and IoT trust management [12]. Trust and reputation management has also been studied extensively in mobile ad hoc networks and wireless sensor networks, which can be envisioned as a partition of IoTs [13].

In the rest of the paper, section II highlights the architecture of Guarantor and Reputation based trust model, section III evaluates the model followed by discussion under section IV. Finally, section V ends the paper with conclusion.

II. GUARANTOR AND REPUTATION BASED TRUST MODEL

A. *The Architecture*

We consider a network comprising of trillions of objects and all these objects are not always be connected in an ad-hoc manner. Take an example of a simple object like smartphone. Most of the time a smartphone is connected either to its cellular service provider or internet service provider or both. A smartphone has the capability to be connected in an ad-hoc manner but it is pragmatic to do so only under certain situations like when a group of smartphones need to be connected to play a game or to transfer a file, or in emergency situation when smartphone losses connection with its primary service provider (gateway). Fig. 1 shows an overview of the Guarantor and Reputation based trust model based on real world scenarios described above. The components are:

- *SIoT Objects and Gateways*

Alice and Carol are SIoT objects. Alice and Carol are connected to the outside network via their SIoT gateways (GW_A and GW_C respectively). GW_A and GW_C are the home network gateways for Alice and Carol respectively. Alice trusts its gateway as the gateway is bound to provide a good service because of the service level agreement (SLA). This is shown as Direct Trust (*dt)* in Fig. 1. For example, Alice can be a smart phone and its cellular provider can be its gateway. Alice has a local trust with its local service provider.

- *Reputation Server*

This model has a central reputation server, which stores and updates the reputations of the objects, and responds to reputation enquiry. This reputation server receives feedback of transactions by the objects and based on those feedbacks the reputation server calculates new trustworthiness and reputation ratings for the object. The reputation server manages the reputation calculations, thus taking off the burden of such complex calculations from the objects. The value of reputation rating is a measure of how trustworthy an object is. There can be a chain of reputation server in order to provide worldwide service if the reputation server does not have the reputation value for an object. In the chain, a reputation server can inquire about the reputation from the next server in line. Although redundant servers can be used, for the simplicity of the model just a single server is considered in this paper.

- *Database Server*

A database server can be used in the SIoT network that has the responsibility of service and discovery of the quickest and most trusted path. This server also stores the type of services provided by different objects. By using the database server, SIoT process such as service discovery and path discovery are no longer necessary to perform by the nodes. In many cases, a SIoT server can act as a database server because SIoT server has the information about the objects registered in the SIoT network, their services and their location. The only capability difference between the SIoT server and the database server is the service discovery and path discovery capability of the database server.

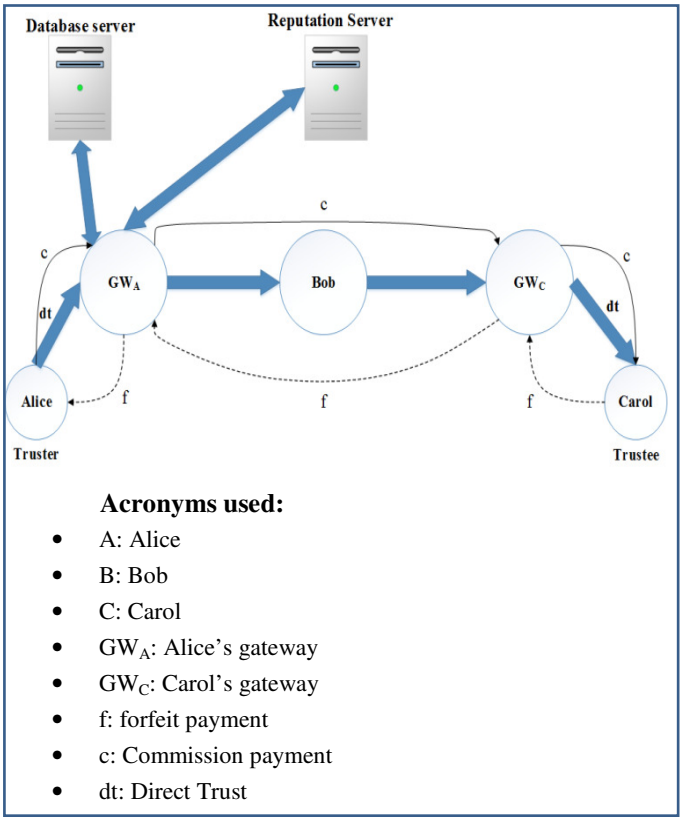

Fig.1. Guarantor and Reputation based Trust Model

- Cooperation of Components

When a new object joins the SIoT network, it goes through the *new object entrance* process. Other SIoT processes like *service discovery, new object relationship* and *service provisioning* still exist in the proposed model but are now handled by the gateways. *Service discovery* and *service provisioning* are also processed by gateways in collaboration with the database server. *New object relationship* is handled

jointly by SIoT server, database server and the owner controlling the objects. For example, if two objects have been in close proximity for a long time then they can initiate *co-location relationship*. The current and last locations of the object are known to both SIoT server and database server. The end decision lies with the owner in control of the object.

B. Trust Management

Every object has a reputation rating associated with it, which is stored in the object itself and is tamper proof. It can only by updated by the reputation server. Agents are built into the objects to update the reputation. This storage of reputation information in the node itself is similar to rate-based storage [6]. When a new object in added to SIoT network, it can have a baseline reputation of neutral (that is 0.5 in this paper). The range of reputation is in the domain of [0, 1], 0 being totally dishonest and 1 being completely trustworthy. Objects are associated with their owners. If the owner buys a new object and associates it with oneself then that object's baseline reputation will be the same as other SIoT objects owned by the same person.

The nodes (including objects and gateways) use credits to get services. If a node acts as an intended server and provides the correct service then the node is paid some credits as commission. If a node acts maliciously and does not provide the correct service then it has to give some credits to the other nodes as forfeit payment. The commission and forfeit rates act as guarantees for an objects behavior. This process is similar to the Trust* as described in [9]. The flow of forfeit and commission payment is shown in Fig. 1 as f and c respectively. At the start of a transaction, nodes send out their commission and forfeit rates that they are willing to offer. These rates can be negotiated with other nodes.

C. Example Scenario

Consider a scenario where Alice wants to get a file from Carol. Alice and Carol are trustworthy, i.e., Carol will not send a corrupted file and Alice will not lie about the quality of the received file. If Alice receives the correct file, she will acknowledge that she has received a good file/service.

The following assumptions are made:

- Alice and Carol have their reputation rating stored in themselves as well as in the reputation servers. It is in the domain [0, 1], 1 being totally trustworthy and 0 being completely dishonest.
- The reputation server handles reputation calculation.
- All objects have been authenticated with the SIoT before they become a SIoT object.
- All objects have been associated with an owner.

The flow of communication is as follows.

(1) A → GW_A: Alice needs a specific file. She asks its gateway (GW_A) if it can get that file for her at her commission and forfeit rates.

(2) GW_A → A: If the gateway accepts the offer, it accepts the commission and forfeit or it can negotiate with Alice before proceeding. After accepting the commission and forfeit rates, GW_A sends an acknowledgement to A.

(3) A → GW_A: Alice sends a request to GW_A to get a specified file.

(4) GW_A → Database Server: GW_A sends a query to the database server enquiring who may have this file. Assume GW_A gets a response from the database server indicating that Carol has this file and providing the safest and quickest path to reach Carol.

(5) GW_A → Reputation Server: GW_A sends a request to the reputation server asking for the reputation of Carol.

(6) Reputation Server → GW_A: The reputation server replies back with the reputation of Carol. Assume that Carol's reputation is above the minimum threshold required by Alice to communicate. GW_A forwards a copy of Carol's reputation to Alice.

(7) GW_A → B: GW_A knows the path to reach Carol, which is included in the reply from database server. GW_A sends the request for file to the next node (Bob) along with the commission and forfeit rate for GW_A. Intermediate nodes just provide the packet forwarding function. GW_A negotiates the commission and forfeit rates only with the end node GW_C.

(8) B → GW_C: Bob forwards the file request along with the commission rate and forfeit rate for this transaction. GW_C can either accept these rates or negotiate for them.

(9) GW_C → C: GW_C forwards the request to Carol. It also sends its own commission and forfeit rates.

(10) C → GW_C: Assume Carol accepts the request and associated commission and forfeit rate, Carol sends back the requested file.

(11) GW_C → B: GW_C forwards the file to Bob.

(12) B → GW_A: Bob forwards the file to GW_A.

(13) GW_A → A: GW_A passes the file to Alice. Along with the file GW_A also sends a copy of Carol's reputation rating so that Alice can itself see the reputation of Carol before proceeding.

(14) A → GW_A: Assume Alice accepts the file and it is a good file. Alice now gives the service feedback. She passes a positive feedback that she received a good service along with any relationship it might hold with Carol to GW_A. Alice also gives the commission to GW_A.

(15) GW_A → Reputation Server: GW_A forwards the feedback from Alice to the reputation server and the reputation server updates the reputation of Carol.

(16) GW_A → Database server: GW_A sends an update to the database server including the last known location of Alice and Carol.

(17) $GW_A \rightarrow GW_C$: GW_A gives commission to GW_C.

(18) $GW_C \rightarrow C$: GW_C gives commission to Carol.

III. Evaluation

A. Implementation, Scenario and Assumptions

In order to test the workings of the proposed model, an application was built in Microsoft Visual Studio 2013 to simulate the data flow in the model. The scenario used in the evaluation as shown in Fig. 2 is the same as the 'Example Scenario' in section II. The gateways and Bob are not shown here because the gateways are just used to forward data. The end-to-end flow of data is between Alice and Carol, and between Alice and the reputation server.

Alice is the initiator of the transaction. She asks for a particular file/service and sends out a query to its gateway GW_A, which in turn queries the database server. Assuming this service is found available at Carol. GW_A checks the reputation of Carol. If it is above the minimum threshold (0.5 in this scenario) GW_A forwards the query to Carol and end-to-end transaction takes place between Alice and Carol. Once Alice gets the required file/service, it sends a transaction feedback to the reputation server via GW_A, which forwards the feedback to the reputation server. Feedback consists of the quality of the service (good or bad) and the type of relationship that Alice holds with Carol. Based on the feedback, the reputation of nodes is calculated by assigning different weights to different relationship. The reputation server is responsible of all these calculations.

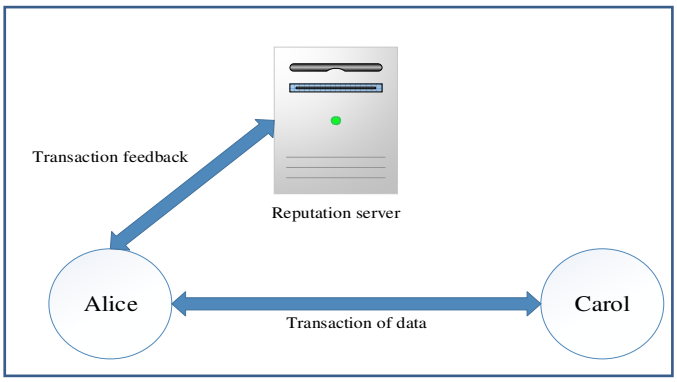

Fig.2. Data flow used in evaluation

The following assumptions are made in the implementation.

- Alice and Carol are two mobile nodes.
- Alice and Carol have an agent built into them that controls the credit logic and is responsible for giving out commission and forfeit rates.
- Both nodes commission and forfeit rates are set to a default value of 1.
- Nodes reputation values are stored in the reputation server and have a default value of 0.5. Reputation rating is in the domain of [0, 1], 0 meaning a node is dishonest and 1 meaning the node is completely trustworthy.
- The minimum reputation threshold at which a node can participate in communications is 0.5, below which the node could be isolated from the network.
- If Carol provides a good service/file then its reputation increases by 0.1 and if it defaults, its reputation drops by 0.3.
- If Carol provides a good service/file then Alice is obliged to give 1 credit as commission to Carol. If Carol defaults then Carol gives 1 credit to Alice as forfeit payment.

B. Simulations Configurations

Each test was run for 10 runs with varying probability that the file/service received from Carol is a malicious service/file. Probability of malicious file/service provided is denoted by *Malware probability* and ranges from 0 to 1, 0 being Carol providing good service all the time and 1 being Carol providing bad service all the time. The chance that the file/service sent back by Carol to Alice is good or bad depends on the parameter *Malwarechance*, where *Malwarechance* has a value of either 0 or 1. 0 meaning file/service sent is good and 1 meaning the file/service is malicious or bad. Depending upon the *Malware probability* and *Malwarechance*, tests were run and values of 'Alice credit', 'Carol credit' and 'Carol reputation' were taken.

In each test, the first few runs have a *Malwarechance* value of 0 as it can be assumed that a malicious node will first try to gain trust and then starts defaulting, which is similar to real life situations. The objective of these tests and of Guarantor and Reputation trust model is to see how quickly the model can identify a malicious node and isolate from the network.

C. Result Analysis

In Test 1 (Fig. 3), *Malwarechance* was always 0 as this was the perfect scenario where Carol never defaulted. This scenario can run for infinite time. Alice was obliged to give 1 of its credits as commission to Carol therefore Alice's credit decreased after paying commissions to Carol and Carol's credit increased. Carol's reputation increased to the maximum and then stabilized.

In Test 2 (Fig. 4), the probability that Carol would default is 0.2, which means that Carol defaulted twice in 10 runs. Assume that Carol built up trust in the first few transactions, reaching the reputation value of 0.9 and then defaulted once, the reputation of Carol dropped to 0.6. At this point, if there was another node available in the network that could provide the same file/service and had a reputation higher than 0.6, Alice would use that node for further transactions. When Carol defaulted, she was obliged to give 1 credit as forfeit payment back to Alice. In case of good transactions, Alice is obliged to give 1 credit as commission to Carol. Changes of Alice and Carol's credits verified these.

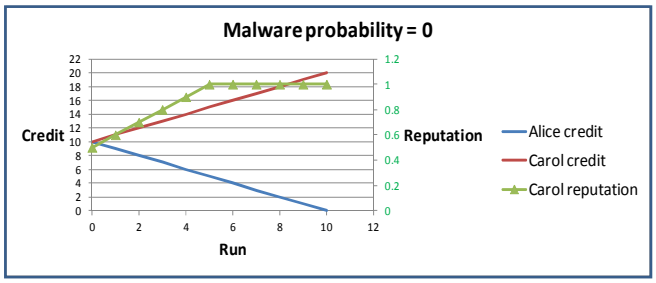

Fig.3. Test 1: Malware probability = 0

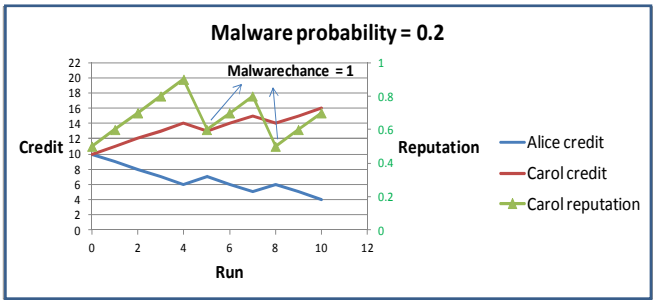

Fig.4. Test 2: Malware probability = 0.2

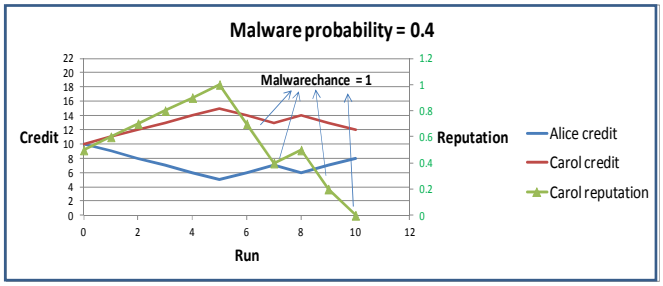

Fig.5. Test 3: Malware probability = 0.4

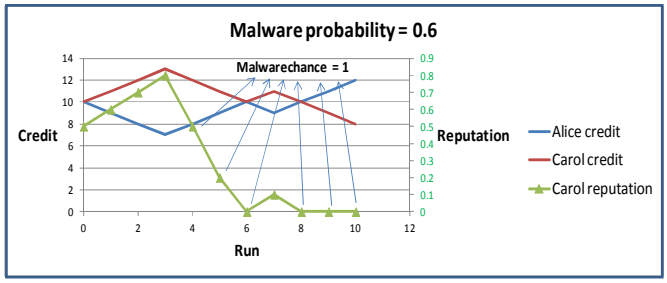

Fig.6. Test 4: Malware probability = 0.6

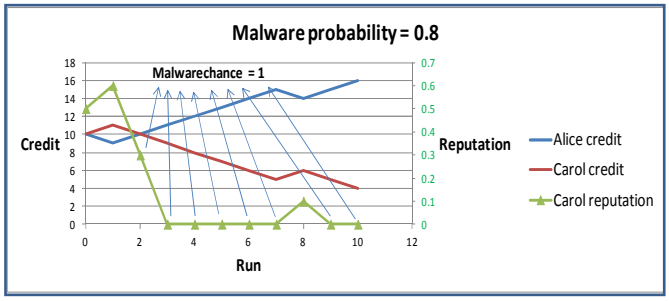

Fig.7. Test 5: Malware probability = 0.8

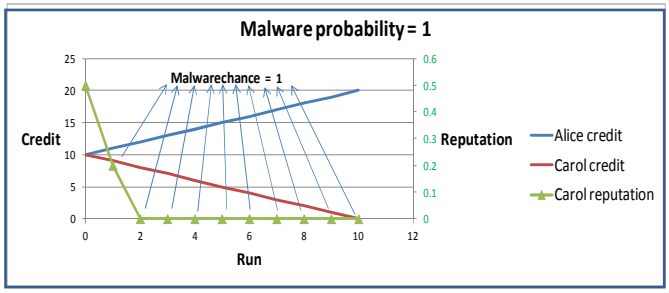

Fig.8. Test 6: Malware probability = 1

In Test 3 (Fig. 5) Carol had a Malware probability of 0.4. It was assumed that Carol built up trust in the first few interactions and increased its reputation to the value of 1 and then defaulted. In just two bad transactions, Carol's reputation decreased to a value of 0.4, which was lower than the minimum threshold of 0.5 and thus it was isolated from the network. Carol also lost her credit when it defaulted.

Test 4 had a similar pattern to Test 3 as can be seen from Fig. 6. Carol had a malware probability of 0.6 where Carol defaulted 60 percent of the time. Again since it built up reputation in the first few interactions, it just took 2 interactions for its reputation to drop below the minimum threshold value. Thus, Carol was isolated in just two bad transactions and lost credits.

In Test 5 (Fig. 7), Carol defaulted 80 percent of the time. In this case, it just took one bad transaction to isolate Carol.

In Test 6 (Fig. 8), Carol defaulted every time and thus was rejected from the network in the first transaction.

IV. Discussions

A. *Use of Guarantor and Credit*

The use of credit, commission and forfeit payment is taken from [9]. The benefits these bring to the model are multifold:

Nodes that provide services use credits to check if the initiator of the transaction is malicious or not. For example, as from the scenario used in the testing, assume Alice receives a good file from Carol but still sends a negative feedback and thus gains 1 credit from Carol. In this case, Carol is losing credit despite sending a good file. Carol can request that the sent file to be tested in a neutral environment by a third party and the file's authenticity to be verified [9]. If the file is found to be authentic, then Alice can be penalized on its reputation value.

Nodes can use credit to get services from home networks as well as visiting networks. When a node connects with a new network**,** it can use its credit to get service from the new network. At the same time, the node will send its ID to the gateway who will check the node's reputation from the reputation server. If the reputation rating of the node is above a defined threshold then the gateway can add this node into its network and start providing service on the node's behalf. It will also update the nodes location in the database server by sending an update message to database server.

If nodes make up an ad hoc network, then the nodes have some capabilities built into themselves to carry out different SIoT process like *new object entry, service discovery and composition,* and *object relationship*. Nodes will now use their credit to establish communication with other nodes. If

any node acts maliciously then it will be isolated from the network based on the number of forfeit payments made by intermediate guarantor nodes. This scenario is similar to the peer-to-peer scenario used in [9]. Moreover, the nodes will store the feedback on every transaction they had and once they connect to their gateways. They will forward this feedback back to the reputation server, which will update the reputation ratings of the nodes.

B. *Use of Reputation*

The Guarantor and Reputation trust model allows devices to build up their reputations, who in turn share the reputations. As the reputation values are stored in a centralized server, these are readily available to whosoever needs it. This model provides a secure and reliable way of communication in SIoT network. This model also acts as a deterrent to malicious activity as it penalizes hard who misbehaves. Thus, it also acts as a prevention model.

The reputation server has the responsibility of calculating the reputation of the nodes. Reputation server assigns weights to the type of relationship in the received feedback. Based on the type of relationships and weights assigned to these relationships, the reputation server calculates a reputation of the nodes. The rationale behind updating of reputations of nodes is such that the reputation decreases at a much higher rate than increasing. For examples, as used in the testing of the model, increase in reputation is only by 0.1 but decrement is three times faster (that is: 0.3) than the increment.

C. *Advantages of the Proposed Guarantor and Reputation Trust Model*

The hierarchical approach will help in the monitoring and management of data as nodes will communicate with other nodes but the traffic will pass either the gateway or the core network. Nodes need not to be involved as transit nodes, which lowers the computational and power needs of the nodes. In addition, the functionalities required to implement SIoT are distributed in different nodes. The gateways are responsible for passing of messages, the reputation server handles the complex calculations involved in producing a reliable reputation, and the database server is involved with service and path discovery process.

V. Conclusions and Future Work

This paper presents a novel trust model for SIoT, the Guarantor and Reputation based trust model. As it does not burden the nodes with reputation calculation and path finding procedure, this model can scale easily to large networks. The model uses two parameters. One is the credit, which reflects the affordability and the cost of a node to find a guarantor for a required a service, and forfeit, the cost of a node to provide faulty service. The other is reputation, which measures the trustworthiness of a node. The use of the two parameters (credit and reputation) for building trust and detecting malicious nodes makes this model reliable. The use of penalties for malicious activity enables this model to detect and isolate malicious nodes. By simulating this model using an architecture based on real world scenarios, it was concluded that the trust model can be employed in different scenarios in SIoT and the use of penalties for malicious activity enables the model to detect and isolate malicious nodes.

Future work can include testing this model's scalability, reliability and data handling capability in a large-scale network. This model can be compared with the 'objective trustworthiness' model by [6] and also guarantor based trust model by [9] by testing all these three model in a common platform and for large-scale networks. Testing of this model should be under ad-hoc and hybrid environment. With some further research about the performance of Guarantor and Reputation trust model in large-scale network, this model can be adopted as a trust model for SIoT.

Acknowledgment

The Authors would like to thank U. Rangaraju for writing the simulation.

References

[1] L. Atzori, A. Iera and G. Morabito, "The Internet of Things: a survey," Computer Networks, 54(15), Oct. 2010, pp. 2787-2805.

[2] R. Sherwood, S. Lee and B. Bhattacharjee, "Cooperative peer groups in nice," INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies. Vol. 2. pp. 1272-1282, 2003.

[3] R. Das and P. Harrop, "RFID forecasts, players and opportunities 2011-2021," IDTechEx, 2011.

[4] National Intelligence Council. (2008). "Disruptive civil technologies - six technologies with potential impacts on US Interest Out to 2025". [Online] Available at: http://www.dni.gov/nic/NIC_home.html.

[5] J. Surowiecki, "The Wisdom of Crowds", Doubleday Press, United States, 2004.

[6] L. Atzori, A. Iera, G. Morabito and M. Nitti, "The social Internet of Things (SIoT) - when social networks meet the Internet of Things: concepts, architecture and network characterization," Computer Network, 56(16), pp. 3594-3608, Nov. 2012.

[7] B. Christianson and W. Harbison, "Why isn't trust transitive?" In Proceedings of the International Workshop on Security Protocols, London. UK. 1997. Springer- Verlag, Lecture Notes in Computer Science, pp. 171-176.

[8] W. Harbison, "Trusting in Computer Systems," Technical Report 437, Wolfson College. University of Cambridge. 1997.

[9] S. Clarke, B. Christianson and H. Xiao, "Trust*: using local guarantees to extend the reach of trust", Proceedings of the International security protocols workshop, Cambridge, UK, XVII. Springer, 2013. Lecture Notes in Computer Science; Vol. 7028, pp. 171-178.

[10] F. Bao and L. Chen, "Trust management for the Internet of Things and its application to service composition", Proceedings of the IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM): 2012, pp.1-6.

[11] 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 2011, 8(4), pp. 1207-28.

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

[13] E. Chiejina, H. Xiao and B. Christianson, "A dynamic reputation management system for mobile ad hoc networks", Computers, 2015, 4(2),pp.87-112.