A Trust-based Resilient Routing Mechanism for the Internet of Things

Zeeshan Ali Khan
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
zakhan@ntnu.no

Johanna Ullrich
SBA Research
Vienna, Austria
jullrich@sba-research.org

Artemios G. Voyiatzis
SBA Research
Vienna, Austria
AVoyiatzis@sba-research.org

Peter Herrmann
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
herrmann@ntnu.no

ABSTRACT

Local-area networks comprising the Internet of Things (IoT) consist mainly of devices that have limited processing capabilities and face energy constraints. This has an implication on developing security mechanisms, as they require significant computing resources. In this paper, we design a trust-based routing solution with IoT devices in mind. Specifically, we propose a trust-based approach for managing the reputation of every node of an IoT network. The approach is based on the emerging Routing Protocol for Low power and Lossy networks (RPL). The proposed solution is simulated for its routing resilience and compared with two other variants of RPL.

CCS CONCEPTS

• **Networks** → **Routing protocols**; **Mobile and wireless security**; **Topology analysis and generation**; *Network performance analysis*; *Short-range networks*; *Network reliability*;

KEYWORDS

Internet of Things (IoT), Routing, Security, Attacks, Resiliency, Reliability

ACM Reference format:
Zeeshan Ali Khan, Johanna Ullrich, Artemios G. Voyiatzis, and Peter Herrmann. 2017. A Trust-based Resilient Routing Mechanism for the Internet of Things. In *Proceedings of ARES '17, Reggio Calabria, Italy, August 29-September 01, 2017,* 6 pages.
https://doi.org/10.1145/3098954.3098963

1 INTRODUCTION AND RELATED WORK

Gartner predicts that in 2020 more than 25 billion devices will be connected to the Internet [21]. Many of these new devices are small and can be used to realize multiple applications. All combined, they realize the concept named the *Internet of Things* (IoT) that will revolutionize our life.

IoT devices and applications, however, have many vulnerabilities and are threatened to be attacked by malicious nodes (intruders). In fact, attackers already managed to infect IoT devices that acted as the nodes of the Mirai botnet that launched in 2016 the largest Distributed Denial of Service (DDoS) attack seen on the Internet until now [4]. Despite the limited capabilities of the devices, the attack achieved an aggregate traffic volume of more than 1 Tbps.

The rapidly growing use of potentially connected devices like digital video recorders (DVRs), IP cameras, or smart thermostats offers intruders new ways to carry out successful assaults. Moreover, the ubiquitous use of IoT systems in our natural surroundings may lead to more serious effects of the attacks, e.g., by voluntarily depleting the battery of a node making it unserviceable. Hence, appropriate countermeasures and protections are necessary when designing an IoT network. Such defenses need to consider the processing capabilities and constraints of the IoT devices [17]. For instance, they need to conserve the limited energy resources of the devices.

A promising approach towards network-level defenses is intrusion detection of malicious nodes based on trust development among the IoT network nodes [13, 19]. However, trust development is susceptible to malicious nodes that falsify the ratings of their neighbors, either by reporting a legitimate one as misbehaving or boosting the rating of a malicious one [5]. Such kind of network-level attacks reduce the effectiveness of trust-based solutions and lead to the disruption of communications across the network [3].

To the best of our knowledge, only a handful of approaches discuss trust management systems tailored for IoT networks and propose lightweight security solutions. An intrusion detection system for IoT is proposed in [18]. The proposal is a centralized system that requires a lot of message exchanges to reach a decision. It also exhibits a high percentage of false positives due to time inconsistencies, as discussed in [15]. A distributed intrusion detection mechanism targeting primarily mobile nodes is proposed in [2]. This is further adapted for IoT-based systems, see [13].

In this paper, we propose a novel network-level framework that aims to support trust development in an IoT network. Our approach is based on the evaluation of the interactions between network nodes. The nodes can have positive and negative experiences with other nodes, as network packets are routed across the IoT network.

ARES '17, August 29-September 01, 2017, Reggio Calabria, Italy

ACM ISBN 978-1-4503-5257-4/17/08…$15.00
https://doi.org/10.1145/3098954.3098963

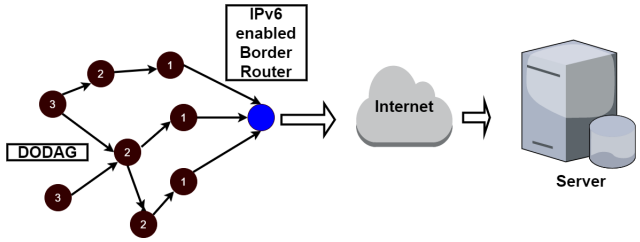

Figure 1: An example of a DODAG network connected to an external server

This computation defines the *direct trust* of a node on its neighbor, see [1]. Based on these information, a central node creates a *rating* for each and every node of the network by combining the collected trust values for each node using Jøsang's Subjective Logic [11]. This contributes in detecting node ratings that fall below a predefined threshold and, thus, detect nodes with suspicious behavior. We assess the routing resiliency of our proposal in the presence of network-level trust attacks.

The remainder of this paper is organized as follows. Section 2 discusses resilient routing with emphasis on RPL-based IoT networks. Section 3 presents our proposal for a trust-based RPL and Section 4 evaluates its performance under different metrics. Finally, Section 5 concludes our paper and describes future directions of work.

2 RESILIENT ROUTING IN AN IOT WORLD

2.1 Network resilience

Malicious attacks on the network level are one of the most critical assault types for the IoT. In this respect, *resilience* refers to the ability of a network to defend itself against such attacks and to maintain an acceptable level of service in the presence of challenges, including malicious actions [20]. A large body of research is devoted to devising appropriate resilience metrics that are systematically surveyed in [20].

2.2 RPL protocol

Many IoT networks employ a distance vector routing algorithm to transport packets across the network. A quite common choice for routing is the Routing Protocol for Low power and Lossy networks (RPL) [10]. RPL uses a hierarchical topology called Destination Oriented Directed Acyclic Graph (DODAG) [16].

Each network node has a specific rank that indicates the number of hops towards the *root* or the *Border Router* of the DODAG. Based on the application requirement, each node can choose an Objective Function (OF) to select a particular path towards the Border Router. In particular, the OF defines the mapping of the data properties and predefined goals such as hop-count, link quality, or energy consumption to a ranking value that approximates the distance of the node to the root of the DODAG. An example of a DODAG for an RPL-based IoT network is depicted in Figure 1.

There can be multiple active DODAGs in an RPL network. However, a node can only join a single DODAG at a time. The node sends a broadcast message if it wants to join the DODAG. Once it receives a reply from its neighbors, it can select a parent based on the chosen OF.

The messages are forwarded from each node to its parent only (i.e., nodes with lower rank). Thus, routing loops are avoided. Nodes can, however, disappear at any time due to energy exhaustion or technical issues. In this case, a repair operation is required that can be either global or local. A global repair covers the complete network while a local one affects only a small portion. The local repair is mostly a relatively cheap operation while the global one is often a costly remedy [14].

2.3 Resilience of RPL routing

Since RPL is a relatively new protocol, it has not been fully and deeply studied and assessed, yet. A first attempt to study the resilience characteristics of RPL is provided in [9]. There, classical RPL performance is assessed in the presence of malicious insiders that selectively drop packets. In that context, *resilience* is defined as the "*capacity to deal with node and link unreliability and node compromise due to an insider attacks*" and as "*the ability of a network to absorb the performance degradation under some failure pattern (random or intentional) and to continue delivering messages with an increasing number of k compromised nodes*" [6].

The resilience of RPL is enhanced by introducing random-path routing and data duplication [9]. The key concept here is that each packet is not forwarded to the best next (lower-ranked) hop as originally defined for DODAGs. Instead, the transmitter selects randomly one of all possible lower-ranked neighbors to forward the packet. This way, a malicious node can, probabilistically, be avoided; when enhanced with data duplication, the approach exhibits improved delivery rates. The proposed algorithm does not induce significant performance and energy penalties and can cope quite well (in simulation) with up to 30% malicious nodes.

Nevertheless, this method does not contain means to eliminate the malicious nodes actively such that those can continue to make trouble. To avoid that, we explore a new RPL routing scheme in the next section that can discard malicious nodes and therefore perform better than the random selection of the next DODAG hop.

3 TRUST-BASED RPL ROUTING

We propose a new RPL routing scheme based on lightweight trust computations as an OF. These computations can execute at a local, cluster, or global scale. Such an approach is already shown to be effective for realizing intrusion detection in RPL-based networks [13]. We extend the approach to a trust-based RPL routing mechanism that not only detects but also eliminates misbehaving nodes altogether.

There are three main kinds of trust-based attacks that can be carried out against the routing functionality of RPL [3]:

- Self-promoting attacks: In a trust management system, an attacker can promote itself by providing good recommendations for itself. That is how it can attract more traffic that can used to carry out selective forwarding attacks.
- Bad-mouthing attacks: An attacker can try to ruin the reputation of non-malicious nodes by providing unjustified bad recommendations for them. Thus, it can reduce the traffic passing through these good nodes.

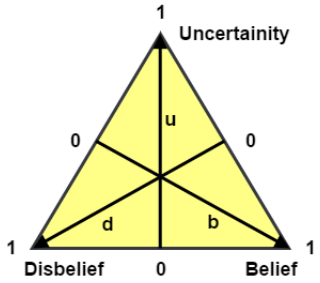

Figure 2: Opinion Triangle (taken from [11])

- Ballot-stuffing attacks: To realize such an attack, an attacker can increase the reputation of other malicious nodes by increasing their recommendation values. Promoting bad nodes will eventually lead to selective forwarding or sink hole attacks.

In the next paragraphs, we explain how the scheme builds the trust relations in an IoT network, thereafter how the mechanism can be used by the nodes for managing network trust, and finally how the trust information can be combined in an efficient way.

3.1 Trust evaluator

Trust between entities can be expressed by discrete values in the case of communicating objects. One technique for expressing trust is the Subjective Logic [11]. There, trust values are represented by opinion triangles that refer to trust, distrust, and uncertainty about a trustee, see Fig. 2. Hence, trust is represented by three variables: belief (b), disbelief (d), and uncertainty (u). The value of these variables vary between 0 and 1, and their sum must be equal to 1, i.e., $b, d, u \in [0, 1] : b + d + u = 1$.

The trust values are based on the positive and negative interactions with a trustee [12]. They are computed as:

$$b = \frac{p}{p + n + k} \tag{1}$$

$$d = \frac{n}{p + n + k} \tag{2}$$

$$u = \frac{k}{p + n + k} \tag{3}$$

The number of positive interactions is denoted with p, while n denotes the negative interactions. A constant value k is used to simplify computations; often it is set $k = 1$ or $k = 2$. A forgetting factor can be used so that more recent interactions get preference (i.e., higher weight) over the older ones [8].

For the case of an RPL-based IoT network, we assume that the network nodes support listening in idle mode for their neighbor's traffic activity. In particular, a node s that has sent a packet to a neighbor t, checks if t, indeed, forwards this packet correctly and timely towards the intended receiver. If that is the case, s increments the p value for t leading to a better trust value. Likewise, if t fails to forward a packet correctly, s increments the n value for t.

Algorithm 1 Reputation computation

```
if Periodic Trust packets are received from network nodes then
    Combine trust values for every node to its reputation value
    for All Nodes do
        if Disbelief > bad_threshold then
            Block the node as a bad node
            Notify the operator and network nodes
        end if
    end for
end if
```

Algorithm 2 DODAG Update Process

```
if My parent is a Bad Node then
    Wait for periodic DIO messages
    Select a new parent that is not a Bad Node
end if
```

3.2 Trust value combination

In intervals, the nodes send the trust value for their neighbors to a central entity (e.g., the RPL Border Router or the Clusterhead). This entity can then combine all collected evaluations for a specific node to derive its overall reputation. If the reputation value falls below a certain level, the DODAG is updated and the misbehaving node is removed from the graph and, in effect, from the IoT network, as it will not be able to route information to the Border Router.

The Subjective Logic defines a consensus operator $\oplus$ that is used for aggregating the various trust values about a node. Assume that $v_{xy} = (b_{xy}, d_{xy}, u_{xy})$ is the trust values vector of node x in node y and $v_{zy} = (b_{zy}, d_{zy}, u_{zy})$ is the trust values vector of another node z in the same node y. The combined trust values vector of the x and z in y is $v_y = v_{xy} \oplus v_{zy}$ and is computed as shown in Equation 4 where κ is equal to $u_{xy} + u_{zy} - u_{xy}u_{zy}$:

$$\left(\frac{b_{xy}u_{zy} + b_{zy}u_{xy}}{\kappa}, \frac{d_{xy}u_{zy} + d_{zy}u_{xy}}{\kappa}, \frac{u_{xy}u_{zy}}{\kappa}\right) \tag{4}$$

This operator can be used to combine all trust values since it is commutative and associative [11]. So, the trust value resulting from the aggregation of all trust values about a node y is a good means to reflect its overall reputation. If the aggregated trust value shows distrust above a certain threshold in y by its neighbors and, therefore, a bad reputation, the network can assume that y is misbehaving. The reason for that can be that the node is compromised and used for a malicious attack. Therefore, in this case a rebuild of the DODAG is re-initiated as described in Algorithm 1. Moreover, the neighboring nodes avoid the badly rated node by executing Algorithm 2.

4 EVALUATION RESULTS

We evaluated the performance of our trust-based RPL (tRPL) with the help of simulations in MATLAB. The proposed scheme is compared against the classical RPL (cRPL) and the resilient RPL (rRPL) proposed in [9]. In particular, we evaluated the performance of our proposal according to the following metrics:

- *Number of bad paths:* The number of paths which include a bad node, after each RPL tree update round.

Table 1: Simulation parameters

Parameter	Value
Network size	500-1500 nodes
Area	100m x 100m
Number of Bad Nodes	20-200
Routing Protocol	RPL
MAC Protocol	Ideal

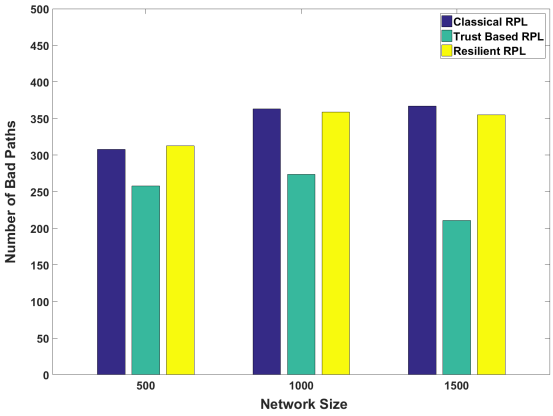

Figure 3: Number of bad paths versus network size (the less the better)

- *Path length:* The length of each node towards the border router.
- *Number of bad nodes:* The number of misbehaving nodes that are identified in the network after every RPL round.
- *False positives:* The number of good nodes that are rated as bad nodes by the network.
- *Average delivery ratio:* The ratio between the total number of packets received by the border router and the total number of packets sent by the IoT nodes.

The setup of the experiment consists of up to 1,500 nodes that are randomly deployed in a 100m x 100m grid. Two sets of simulations were carried out to compare the three RPL versions (tRPL, cRPL, and rRPL). In the first set, we vary the number of nodes in the network. In the second set, we vary the number of malicious nodes in the network.

The simulation parameters are summarized in Table 1. We discuss in the following the results of our evaluation.

4.1 Network size

The number of nodes varied between 500, 1,000, and 1,500. We observed the effect of the different numbers on the network parameters. When the network size increases, the number of bad paths is reduced in the case of tRPL. However, the number of bad path *increases* for both cRPL and rRPL, as depicted in Figure 3.

This can be explained by the fact that tRPL has an inherent mechanism to detect bad (malicious) nodes in the network and that it tries to avoid paths including these bad nodes. Some false

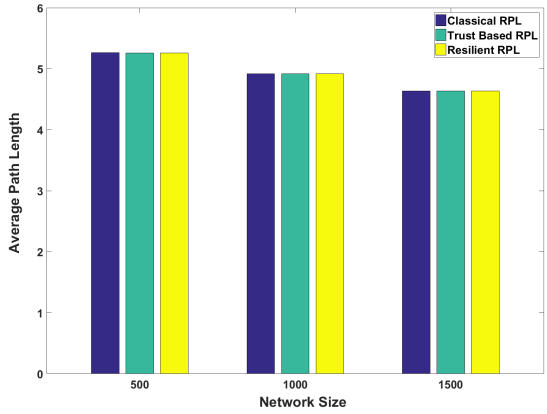

Figure 4: Average path length versus network size (the less the better)

positives are still possible; this is discussed in Section 4.3. Both cRPL and rRPL exhibit a similar (and greater than tRPL) number of bad paths. This, despite rRPL trying to avoid bad paths by selecting randomly one of the available parent nodes.

When the network size *increases*, the average path length *slightly decreases* for all the three RPL versions, as depicted in Figure 4. It decreases from 5.3 to 4.6 hops, if the network size increases from 500 to 1,500. This is to be expected due to the random node placement; on average the nodes will be well-connected to the Border Router the more nodes are available. It is interesting to note that in the case of tRPL, the average path length does not differ significantly compared to cRPL and rRPL, despite using a completely different OF.

The average delivery ratio in relation to network size for the three RPL versions is depicted in Figure 5. Our trust-based RPL achieves a better packet delivery ratio in all cases. This can be accredited to the availability of additional paths for packet delivery as the network size increases and to the exclusion of misbehaving nodes that drop packets.

4.2 Bad nodes

In the second set of simulations, we varied the number of bad nodes between 0 and 200 while keeping the network size fixed at 1,000 nodes. The effect on network parameters is that when the number of bad nodes increases, the number of bad paths also increases for all the three RPL versions, as depicted in Figure 6. This is due to the fact that bad nodes drop out all incoming traffic instead of forwarding it to the next hop. Increasing the ratio of bad nodes has a negative effect on the number of available (legitimate) paths. Still, the number of bad paths in the case of tRPL is lower than those of cRPL and rRPL. This is accredited to the detection mechanism used by tRPL. The performance of rRPL is slightly better than that of cRPL. The former avoids bad nodes sometimes due to the random selection of the parent among all the available ones.

The average path length is more-or-less independent of the RPL version used when the ratio of bad nodes is increased, as depicted in Figure 7. This can be explained by the fact that the network size

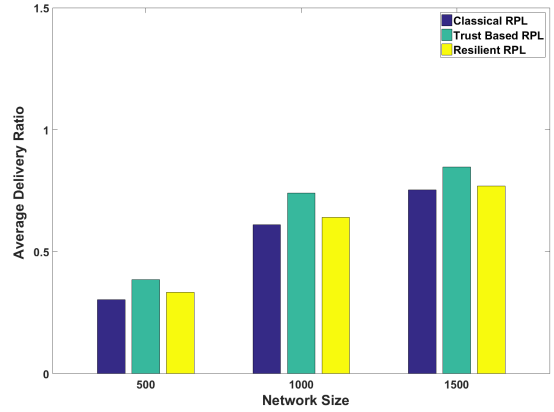

Figure 5: Average delivery ratio versus network size (the more the better)

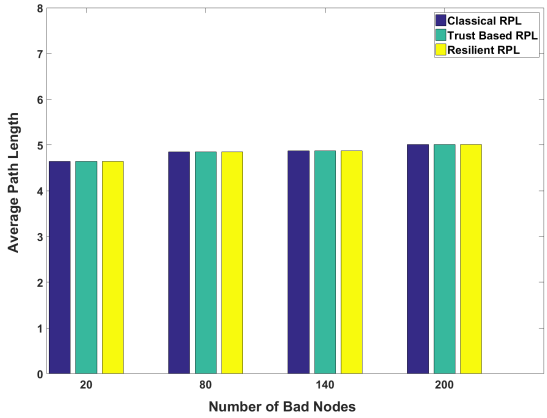

Figure 7: Average path length versus number of bad nodes (the less the better)

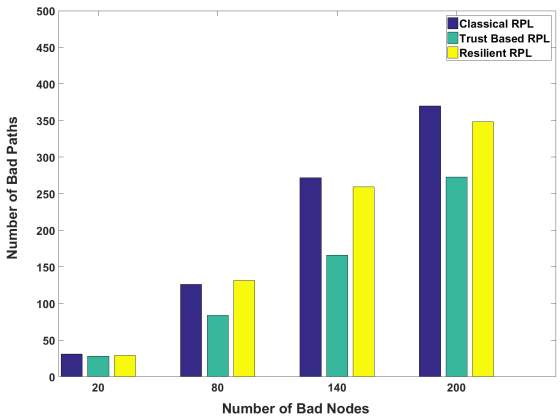

Figure 6: Number of bad paths versus number of bad nodes (the less the better)

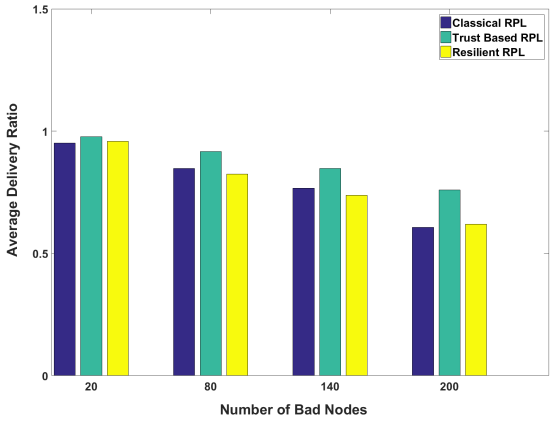

Figure 8: Average delivery ratio versus number of bad nodes (the more the better)

is always the same (1,000 nodes). However, the average path length *increases* as the ratio of bad nodes *decreases*. This is due to the fact that the number of long paths that contribute to an increased average path length is actually reduced.

The average delivery ratio for all the three RPL versions as a function of the number of bad ones is depicted in Figure 8. The tRPL exhibits a *better* packet delivery ratio compared to cRPL and rRPL. Again, this can be accredited to the wider availability of network paths for packet transmission. The delivery ratio decreases when the ratio of bad nodes increases; again due to the reduced availability of paths for packet transmission.

4.3 Detection and error rates for tRPL

The tRPL is the only of the three studied RPL versions that can detect and isolate the presence of bad nodes in the network. The rRPL utilizes random path selection to probabilistically avoid bad nodes on a per-packet basis. The classical RPL does not consider the presence of malicious or misbehaving nodes at all.

Our simulation results are summarized in Table 2. The tRPL can detect the presence of nearly 80% of the bad nodes after just five simulated rounds. However, tRPL marks some good nodes as bad, i.e., it suffers from false positives. Still, the rate is fairly low, as depicted in Table 2. Hence, we argue that tRPL can be deployed for effective detection and isolation of misbehaving nodes in an RPL-based network.

4.4 Discussion of the three RPL versions

The performance of cRPL and rRPL is quite similar in terms of numbers of bad paths and average path lengths. The additional complexity introduced by rRPL does not bring a significant benefit in the presence of bad nodes (intruders). In contrast, tRPL provides better results than the two other schemes with respect to the number of bad paths, the number of bad nodes, and the average delivery

Table 2: Detection and error rates for tRPL

Number of Bad Nodes	Number of Bad Nodes Detected	Number of Good Nodes Identified as Bad Ones
20	20	1
80	75	2
140	127	15
200	175	36

ratio. The average path length is not affected by the selection of the RPL version. Moreover, the computations required for tRPL are lightweight and be realized from the local up to the global level. Thus, we do not expect a significant penalty on resource utilization for real-world implementations of tRPL.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we studied the case of network-level attacks on RPL-based IoT networks. We are the first to introduce a trust-based RPL routing mechanism for such networks, building atop existing proposals for intrusion detection.

We evaluated the performance of our proposal against both the classical and the resilient variants of RPL. Our trust-based RPL exhibits better performance characteristics. This is since tRPL integrates a lightweight mechanism to detect and also isolate the bad nodes from the network, resulting in better network resilience.

In the next step, we will implement tRPL, cRPL, and rRPL on emulated and real-world IoT systems. Currently, we build a testbed consisting of up to 20 Z1 devices that run on the operating system Contiki. In addition, we will be able to add an emulator for an arbitrary number of further devices of all kinds which will allow us to repeat our evaluations for tRPL and the two other schemes for varying topologies. Moreover, we plan to experiment with different trust metrics and thresholds in order to reduce the number of false positives.

Currently, our mechanism is not actively protected against the trust-based attacks listed in Sect. 3. Therefore, we plan to extend it by not only considering direct trust but also trust in recommendations, see [7]. The nodes will not only build trust values about the effective behavior of their neighbors but also about the correctness of the neighbors' recommendations of third parties. Then, we can devaluate the direct trust values of nodes with a bad recommendation trust value in the computation of aggregated trust values. The Subjective Logic offers a discounting operator making it easy to use recommendation trust in a resource-preserving way [11].

ACKNOWLEDGMENTS

Z.A. Khan ackowledges the support by European Research Consortium for Informatics and Mathematics (ERCIM), as this work was partially carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. Further, this work received support by the Austrian Research Promotion Agency (FFG) through the Bridge Early Stage grant P842485 (CyPhySec).

REFERENCES

[1] Thomas Beth, Malte Borcherding, and Birgit Klein. 1994. Valuation of Trust in Open Networks. In *European Symposium on Research in Security (ESORICS) (LNCS 875)*. Springer-Verlag, 3–18.

[2] Christian Cervantes, Diego Poplade, Michele Nogueira, and Aldri Santos. 2015. Detection of Sinkhole Attacks for Supporting Secure Routing on 6LoWPAN for Internet of Things. In *13-th IFIP/IEEE International Symposium on Integrated Network Management (IM-2015)*. IEEE Computer, 606–611.

[3] Ray Chen, Jia Guo, and Fenye Bao. 2016. Trust Management for SOA-based IoT and its Application to Service Composition. *IEEE Transactions on Services Computing* 9, 3 (2016), 482–495.

[4] Sam Cohen. 2016. Internet of Things as the next Industrial Revolution - How it will change the industry in 2016. Huffington Post. (2016). http://www.huffingtonpost.com/sam-cohen/internet-of-things-as-the_b_10937956.html Accessed: 2017-03-16.

[5] Chrysanthos Dellarocas. 2000. Immunizing Online Reputation Reporting Systems Against Unfair Ratings and Discriminatory Behavior. In *2nd ACM Conference on Electronic Commerce (EC)*. ACM Press, 150–157.

[6] Ochirkhand Erdene-Ochir, Apostolos Kountouris, Marine Minier, and Fabrice Valois. 2012. A New Metric to Quantify Resiliency in Networking. *IEEE Communications Letters* 16, 10 (2012), 1699–1702.

[7] Peter Herrmann. 2003. Trust-Based Protection of Software Component Users and Designers. In *1st International Conference on Trust Management (LNCS 2692)*. Springer-Verlag, 75–90.

[8] Peter Herrmann. 2006. Temporal Logic-Based Specification and Verification of Trust Models. In *4th International Conference on Trust Management (iTrust-2006) (LNCS 3986)*. Springer Verlag, 105–119.

[9] Karel Heurtefeux, Ochirkhand Erdene-Ochir, Nasreen Mohsin, and Hamid Menouar. 2015. Enhancing RPL Resilience Against Routing Layer Insider Attacks. In *29th IEEE International Conference on Advanced Information Networking and Applications (AINA-2015)*. IEEE Computer, 802–807.

[10] IETF. 2012. RfC 6550 — RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. https://tools.ietf.org/html/rfc6550. (2012). Accessed: 2016-10-24.

[11] Audun Jøsang. 2001. A Logic for Uncertain Probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems* 9 (2001), 279–311.

[12] Audun Jøsang and Svein J. Knapskog. 1998. A Metric for Trusted Systems. In *21st National Security Conference*. NSA, 16–29.

[13] Zeeshan Ali Khan and Peter Herrmann. 2017. A Trust Based Distributed Intrusion Detection Mechanism for Internet of Things. In *31st IEEE International Conference on Advanced Information Networking and Applications (AINA-2017)*. IEEE Computer, 1169–1176.

[14] Kevin Dominik Korte, Anuj Sehgal, and Jürgen Schönwälder. 2012. A Study of the RPL Repair Process using ContikiRPL. In *6th IFIP International Conference on Autonomous Infrastructure, Management and Security (AIMS-2012)*. Springer-Verlag, 50–61.

[15] Takumi Matsunaga, Kentaroh Toyoda, and Iwao Sasase. 2014. Low False Alarm Rate RPL Network Monitoring System by Considering Timing Inconstancy between the Rank Measurements. In *11th International Symposium on Wireless Communications Systems (ISWCS-2014)*. IEEE Computer, 427–431.

[16] Belghachi Mohamed and Feham Mohamed. 2015. QoS Routing RPL for Low Power and Lossy Networks. *International Journal of Distributed Sensor Networks* 2015 (2015), 6.

[17] Shahid Raza, Hossein Shafagh, Kasun Hewage, René Hummen, and Thiemo Voigt. 2013. Lithe: Lightweight Secure CoAP for the Internet of Things. *IEEE Sensors Journal* 13, 10 (2013), 3711–3720.

[18] Shahid Raza, Linus Wallgren, and Thiemo Voigt. 2013. SVELTE: Real-time intrusion detection in the Internet of Things. *Ad hoc networks* 11, 8 (2013), 2661–2674.

[19] Dharmini Shreenivas, Shahid Raza, and Thiemo Voigt. 2017. Intrusion Detection in the RPL-connected 6LoWPAN Networks. In *3rd International Workshop on IoT Privacy, Trust, and Security (IoTPTS 2017), held in conjunction with the 12th ACM Symposium on Information, Computer and Communications Security (ASIACCS-17)*. ACM, 31–38.

[20] Paul Smith, David Hutchison, James PG Sterbenz, Marcus Schöller, Ali Fessi, Merkouris Karaliopoulos, Chidung Lac, and Bernhard Plattner. 2011. Network Resilience: A Systematic Approach. *IEEE Communications Magazine* 49, 7 (2011), 88–97.

[21] Alfonso Velosa, James F. Hines, Hung LeHong, Earl Perkins, and Satish R.M. 2014. Predicts 2015: The Internet of Things. https://www.gartner.com/doc/2952822/predicts--internet-things. (2014). Accessed: 2017-03-23.