Data Trustworthiness in IoT

Sabah Suhail, *Choong Seon Hong
Department of Computer Engineering
Kyung Hee University
Yongin, Korea
sabah,cshong@khu.ac.kr

M. Ali Lodhi
Department of Computer Science
COMSATS Institute of Information and Technology
Sahiwal, Pakistan
alilodhi30@googlemail.com

Faheem Zafar, Abid Khan
Department of Computer Science
COMSATS Institute of Information and Technology
Islamabad, Pakistan
faheemiiui@gmail.com, abidkhan@comsats.edu.pk

Faisal Bashir
Department of Computer Science
Bahria University
Islamabad, Pakistan
faisalwn@yahoo.com

***Abstract*—Internet of Things (IoT) is deployed in numerous pervasive application areas, for instance, environment monitoring, energy management, health-care system and industrial automation. Data are streamed from multiple sources and is traversed through intermediate nodes until it arrives at sink node which performs decision-making for critical infrastructures. Malicious or compromised nodes may forge data or inject false data. Therefore, assuring high data trustworthiness is crucial for precise decision-making. Data provenance play an important role in evaluating the trustworthiness of data. However, provenance management for resource-constrained devices introduces several challenging requirements, such as network overhead, energy consumption, and efficient storage. In this paper, we formulate the problem of binding IoT with a provenance-aware system to enable it to track the data flow across the networked entities and data transformations applied to data by nodes. We propose a lightweight scheme to transmit provenance for IoT sensor data. The proposed technique relies on the hash chain scheme to encode provenance as the packet traversed from each participating node while the provenance verification is done at the sink node. Furthermore, we evaluate our technique with respect to energy consumption by the constrained nodes.**

***Index Terms*—provenance, trustworthy data, integrity, RPL**

I. Introduction

The idea of deploying Internet-connected-things (sensor devices) to monitor and analyze physical world objects in real time has become an intelligent source of disseminating services in various critical cyber-physical infrastructure. However, due to the interconnection of things with untrusted Internet, the data generated by sensor devices are vulnerable to attacks including data forgery, data disruption or false data injection and therefore, the trustworthy information cannot be assured in the decision-making process.

We consider the following case studies to illustrate the significance of data trustworthiness in IoT. Consider a personalized wellness recommender system that collects sensor data from Alice's wearable devices, process sensor data to curate information and provides context-aware personalized recommendations to users. Suppose a case of false data propagation in which the adversary, Mallory is able to perform malicious activities (for instance, eavesdropping or packet injection or drop) on any of Alice's wearable devices. Then, *how to ensure that the collected data from multimodal data sources arriving at recommender system does not meddle with false data and ultimately the recommendations are not based on fictitious data values?* In additional to wearable devices other implantable medical devices (IMD) are prone to attack due to their connectivity and remote-communication capabilities as they are operating in the untrusted Internet. According to 2016 Internet Security Threat Report (ISTR), many medical devices (such as insulin pumps, x-ray systems, CT-scanners, medical refrigerators, and implantable defibrillators) are deadly vulnerable to cyber threats [1]. One such threat was reported by Reuters about Medtronic insulin pump that can be hacked to dose fatal amount of insulin to diabetes patients [2].

Another example is related to smart grids that operate by facilitating consumers through collecting their power consumption information and providing them value-added services including billing information, controlling energy conservation, usage patterns etc. However, such a two-way communication between consumer and control center widens the likelihood of cyber-crimes, for instance, manipulation of meter reading. Suppose a greedy customer, Mallory attempted to perform energy theft by forging his meter reading and event logs thereby depicting false usage patterns. Since such kind of services are usually charged and the revenue is based on data or services used hence, the *integrity* of the data is highly pivotal to prevent energy fraud issues. In such scenario, *how to ensure that information flow between the smart meter and utility company is reliable or fraudulent and*

*Dr. CS Hong is the corresponding author.

978-1-5386-2290-2/18/$31.00 ©2018 IEEE

how to detect vulnerability causing anomaly.

Similarly, consider a nuclear reactor where sensors (for instance, temperature, flow, pressure or level) are deployed in order to monitor (heating system or water pressure or level) and notify sensor readings (temperature measurements or water level) to a control room. The control room is responsible for executing critical decisions (to turn off/on any valve or to adjust any values) based on the sensors readings. A traitor eavesdrops on the sensor data and forged the sensor values (increase or decrease the temperature values) for instance, the destruction caused by Stuxnet worm [3] in Iran's nuclear plant. *How the control room can validate the data values from sensors?*

Under such circumstances, how to ensure:

a) trustworthiness of data generated by sensor nodes.
b) authenticity of the decisions or recommendations communicated to users or system by IoT sensor network.
c) data modifications are either caused by device malfunction or malicious user.
d) main source of disruption in the system.
f) sensor system is working accurately or not.

It is evident from the above examples that there must be an intelligent tracking mechanism that can ensure the integrity of sensor data during data debugging, reconciliation, replication, decision making, performance tuning, auditing, forensic analysis or other conflicting issues. One such mechanism is termed as provenance. Provenance is a meta-data describing the complete lineage of data and set of actions perform on data [4].

Our specific contributions with reference to verifying the integrity of provenance data are:

- We formulate the problem of providing data trustworthiness in IoT;
- We propose a lightweight technique by employing hash chain for provenance encoding;
- We implement the proposed technique for IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [5] in Contiki OS [6];
- We evaluate the proposed technique with respect to energy consumption by the constrained nodes.

The rest of the paper is organized as follows: Section II presents the related work. Section III sets the challenges and constraints for provenance-aware IoT system. Section IV introduces the system model whereas problem formulation is defined in Section V. Section VI explains the working of the provenance scheme. Section VII presents the experimental evaluation results. We conclude the paper with future research directions in Section VIII.

II. Related Work

The interconnectivity of 6LoWPAN networks with the Internet raises serious security concerns, as resource-constrained things are globally accessible anywhere from the untrusted Internet [7]. An attacker may attempt to disrupt the data generated by nodes either by inserting malicious nodes or by compromising benign nodes. Such forging or alteration of data produces catastrophic results especially for the applications relying on things data for critical decision-making processes, risk assessment, and performance evaluation. To enable the data trustworthiness among IoT devices, provenance can be deployed.

Provenance has been extensively studied in a variety of application areas including databases, scientific workflows, distributed systems, and networks and has many ingredients (for instance, confidentiality, integrity, availability, privacy, and non-repudiation) that are discussed in detail in [8]. Considering the domain of sensor networks, we focus particularly on the integrity of provenance data.

In sensor networks, the idea of provenance with the goal to ensure data trustworthiness has been addressed by [9]–[12]. [9] and [10] introduce the concept of employing in-packet bloom filter whereas [12] introduces watermarking based on inter-packet delay as provenance information in WSN.

The integration of provenance with resource-constrained devices employing Routing Protocol for Low-Power and Lossy Networks (RPL) is a challenging task as discussed in next section. To the best of our knowledge, the framework for provenance in IoT has not been worked out so far.

III. Challenges and Constraints of Provenance-Aware IoT system

In [13] we have identified requirements and challenges for integration of secure provenance with IoT. However, here we address the following problems (P) keeping in view the provenance as overhead for resource-constrained devices.

P1: Provenance collection: The first issue is related to provenance collection which arises the following questions:

i) What type of data should be collected that can help to figure out the data inconsistencies or malicious activities? for instance, the id of participating nodes, timestamps, and data flow (either involving data modification or packet forwarding.
ii) From where to collect the provenance? for instance, collect meta-data from each node and then disseminate the aggregated data at sink node or root node?
iii) What will be the network overhead in comparison to the granularity level of data being collected? for instance, how many extra bytes in a packet are required to be sent along with data packet as provenance?

P2: Resource-constrained devices: The above questions themselves raise several other questions in relation to the resource-constrained devices, for instance, nodes energy constraints (battery-powered or battery-less), nodes storage capacity (flash memory), and nodes processing power etc.

P3: Network overhead: To illustrate the network overhead we need to identify those additional bytes to be added in the

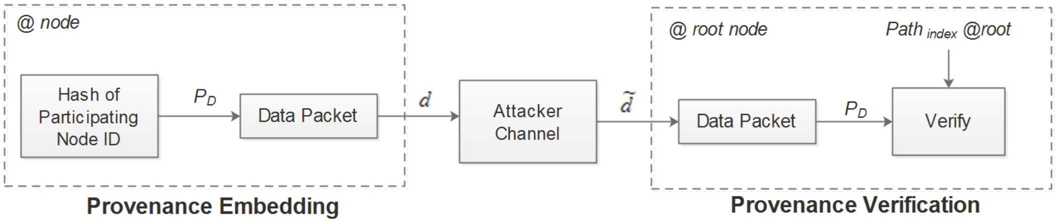

Fig. 1. Stages of Provenance Embedding and Verification

packet to support the idea of provenance. For instance, data flow (node id of all participating nodes starting from the source node to root node) and other related attributes like sequence number etc.

P4: Performance overhead: As the extra meta-data may lead to network overhead which results in another problem related to performance that is critical for many applications. In other words, we expect trustworthy data at the cost of system performance degradation. So a trade-off is required to establish a well-balanced performance-oriented secure solution for IoT devices.

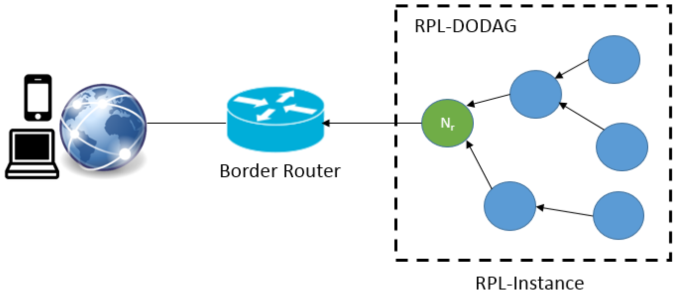

Fig. 2. System Model

IV. System Model

A. Network Model

We consider a RPL network. The border router (BR) connects the RPL network with Internet. In RPL-DODAG, root node (sink node) collects data from nodes of a sub-DODAG and sends it to the BR that forwards it to the Internet host to interpret and analyze the sensor data.

The network is modeled as a graph $G(N, E)$ such that:

$$N = N_s, N_{i1}, \ldots, N_{in}, N_r$$

where N is the set of nodes consisting of source node N_s, intermediate nodes $N_{i1}, \ldots, N_{in}$ and root node N_r while E is the set of edges representing the path between two nodes traversed by a packet.

B. Provenance-Aware Data Model

We consider a provenance-aware data model that keeps track of:

a) List of participating Nodes (N_ω)

b) List of actions $(\mathcal{A})$ performed by participating Nodes Hence, the provenance data P_D track the actions $\mathcal{A}$ performed by set of participating nodes N_ω as:

$$P_D = \mathcal{A}(N_\omega) \tag{1}$$

Actions in terms of provenance data can be divided as:

i) Packet forwarding
ii) Data aggregation

$$p_d = d_f(N_{id_\omega}) \tag{1a}$$

i.e. the data flow d_f represents the identity of the participating nodes N_ω.

$$p_d = d_f(N_{id_\omega} + d^{'}) \tag{1b}$$

i.e. the data flow d_f represents the identity of the participating nodes N_ω along with the aggregated data $d^{'}$. We have explained more about the collection of data provenance in the next section.

V. Problem Formulation

We have considered the following assumptions:

Assumptions:

a) a whitelist W_c of legitimate child nodes maintained by parent nodes n_p.

b) a blacklist of B of malicious nodes maintained by root node N_r.

c) a symmetric cryptographic key K_i.

d) a set of hash functions $\mathcal{H}$.

We may divide the provenance embedding mechanism in 2 sub-cases.

a) Case I: Packet forwarding

b) Case II: Aggregated data forwarding

A. Case I: Packet forwarding

Packet forwarding involves the intermediate participating nodes forwarding data packet between source N_s and destination N_d in a multi-hop network. We need the following requirements (R) mentioned below:

RI(a) :Whitelist W_C

- Each parent node has a whitelist W_C of all of its legitimate child nodes (*Check-I*).

$$N_p \leftarrow W_C$$

- Since in IoT, every node has a global identification or IP Address hence we can say that the whitelist W consist of IP address of all the legitimate nodes in the network.
- Making a whitelist at parent nodes will allow the parent node to dispose of the received data if its not from one of its legitimate child node(s).
- The parent node maintains a table [1] to keep track of this child nodes.

RI(b) :Whitelist $W_{P:C}$

- The root node (special parent node) has a whitelist $W_{P:C}$ of parent and its corresponding child nodes (*Check-II*)
$$N_r \leftarrow W_{P:C}$$
- Following this tradition may allow the root node to cross-check for legitimate data from legitimate nodes such that:
  –if the parent node could not classify a malicious child node as compromised node or
  –if a compromised node is acting as the parent node.
- To facilitate a dynamic setting, the root node is responsible for keep on updating the W_C at parent nodes in either case:
  –adding new legitimate node(s)
  –removing old comprised node(s)

RII :Blacklist B

- The root node also maintains a list of compromised nodes.

RIII :Dataflow integrity: P_{chain}

- To ensure the integrity of dataflow the node compute the hash of node ID's.

Definition 1: The Proof $\mathcal{P}$ represents the presence of a node id in provenance data p_d for forwarding a data packet d such that:

$$\mathcal{P} = \mathcal{H}(N_w)$$

where $\mathcal{H}$ is a set of hash functions and N_W is the node id of participating nodes.

VI. Provenance Scheme

Fig. 1 shows the overview of the proposed provenance scheme. The detail working of each stage is as follows:

1) Provenance Embedding: Consider the data flow from source node 4 to destination node N_r (see Fig. 3). The node 4 generates a data packet including fields: packet sequence number, data and provenance data. Node 4 computes $\mathcal{P}(4)$ concatenated with initialization vector IV and passes it to node 2. The node 2 first checks its W_C list to assure that the received packet is from a legitimate child (Check-I) and then it computes the hash as the proof of the presence of previous node 1 along with the proof of its own presence in P_{chain}. Thus, each subsequent node in the data path will keep on aggregating the provenance data p_d in provenance chain P_{chain} as:

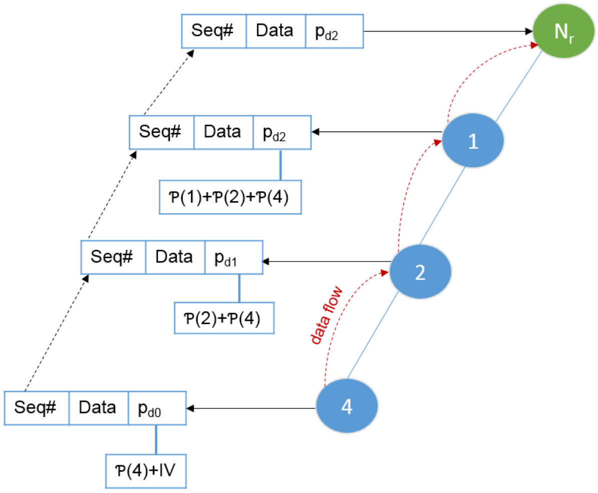

Fig. 3. Provenance Embedding

$$\mathcal{P}(currentNode) + \mathcal{P}(previousNode(s))$$

where $\mathcal{P} = \mathcal{H}(N_w)$ according to def. 1. This process continues until the data packet holding the provenance reaches the root node. Since we consider integrity as the main security objective of our technique, we do not emphasize on the encryption of provenance data. However, confidentiality can be achieved by encrypting the provenance with the secret keys of the respective nodes. Moreover, we consider the challenges mentioned in section III to make provenance workable in resource-constrained devices.

2) Provenance Verification: The root node has already acquired the knowledge of packet's path $(Path_{index})$ via DIO (DODAG Information Object). In order to verify the provenance data, the root node N_r computes the hash $\mathcal{H}$ for $\mathcal{P}(N_i \rightarrow N_j)$ where N_i represents the source node and N_j represents the destination node. The root node then compares the calculated value with the value in the P_{chain} received along with data packet d from the $(Path_{index})$ maintained at root node in order to verify the path (Check-II). If the path is not verified, then root node discards the packet.

A. *Case II: Aggregated data forwarding*

Consider an application in industrial process monitoring and control where sensors nodes collect, store and transmit processed data over specific time periods (for instance, average sensor data or sampling sensor data points) to a sink node. Such specific data requests may be necessary for error diagnosis or to calibrate the overall production processes [14]. Hence, the provenance data must ensure not only the node ids of participating nodes but also the data being accumulated from them at the aggregator node(s) N_g. Implementation of aggregated data forwarding will be part of our future work.

[1] Nodes may use DODAG Information Solicitation (DIS) messages to request graph related information from the neighboring nodes.

VII. Experimental Evaluation

We run our experiments on Cooja [15] which is Contiki based simulator. Cooja is developed for Windows and Linux platform [16]. For our simulations, we use Tmote sky as things. A Tmote sky has a CCC2420 transceiver and 48kB of ROM. Simulations are executed for 10 min and the reporting starts after 1 min because initial time is required to converge topology. All simulations are run 5 times and average results are presented. We have considered the following 2 scenarios for simulation:

A. Scenario I: Benign Node(s)

To evaluate the working of the proposed technique, we consider 10 source nodes from 2 to 11 where 1 is sink node (see Fig. 4). For example, the data packet traverses the path as follows: 6→10→4→1. The node 6 being the source node attach the hash of its node id. [2] The source node after adding the hash of its node id forward the packet to its preferred parent node 10 which first evaluate node 6 as legitimate or not from its whitelist of legitimate of nodes. Node 6 add the hash of its node id as proof and forward it to its preferred parent. Then the subsequent nodes keep on following the procedure discussed in SectionVI-1.

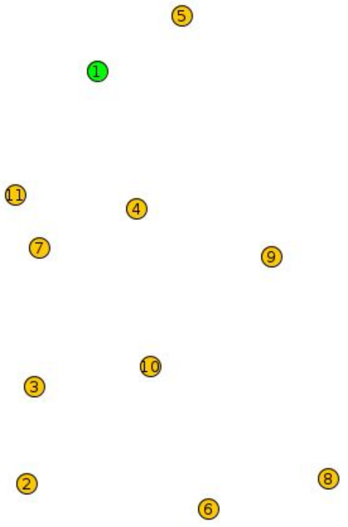

Fig. 4. Network configurations: Benign Nodes

B. Scenario II: Malicious Node

To justify and evaluate the trustworthiness of provenance data, we have introduced a malicious node in the path as follows: 6→12→10→4→1 (see Fig. 5). Node 12 may intent to perform any attack, for example, replay attack or data forge attack. When the packet along with provenance data arrives at sink node 1, it verifies the packet path as discussed in section VI-2.

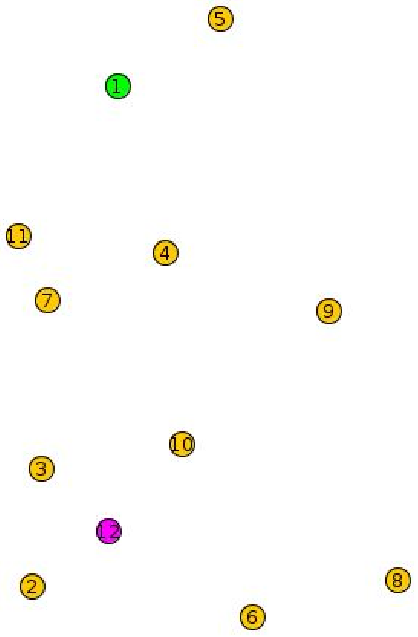

Fig. 5. Network configurations: Attacker Node

C. Energy Consumption

The nodes in the IoT are usually battery powered and hence energy is a scarce resource. Therefore, we measure energy consumption for both RPL without Hash Chain Provenance and RPL with Hash Chain Provenance technique at system-level by using the nominal values of the Tmote sky [17]. In order to compute the energy, we have use the following equation [18]

$$\begin{aligned} Energy(mJ) = (&Tx * 19.5mA + Rx * 21.8mA \\ &+ CPU * 1.8mA + LPM * 0.0545) \\ &* 3V/4096 * 8 \end{aligned} \tag{2}$$

where as Tx and Rx represent the values for transmitter and receiver respectively.

Fig. 6 shows that energy consumption in case of Hash Chain Provenance is almost tolearable when compared with the RPL without Hash Chain Provenance.

From the system-level energy usage, we calculate the average power as:

$$Power(mW) = \frac{Energy(mJ)}{Time(s)} \tag{3}$$

Fig. 7 shows that average power consumption per node in case of Hash Chain Provenance is almost negligible when compared with the RPL without Hash Chain Provenance. Hence we can achieve data trustworthiness at cost of extra power consumption.

[2] For node identification, Cooja emulator provides the facility of using compressed 8-bit address instead of 128-bit IPv6 address, hence, we consider compressed 8-bit address for hash computation.

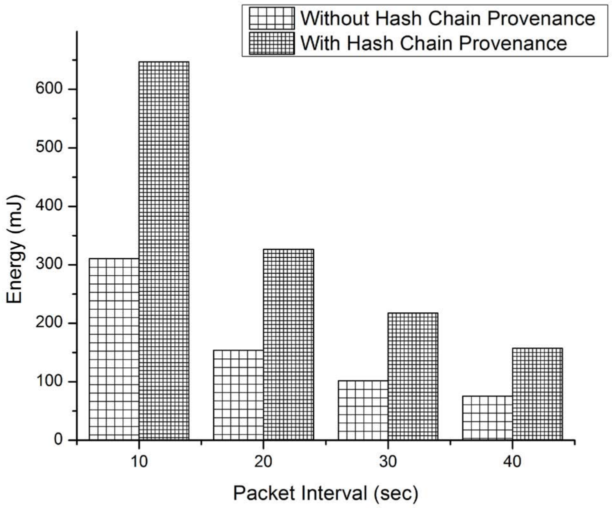

Fig. 6. Comparison of Energy Consumption between without hash chain provenance and with hash chain provenance

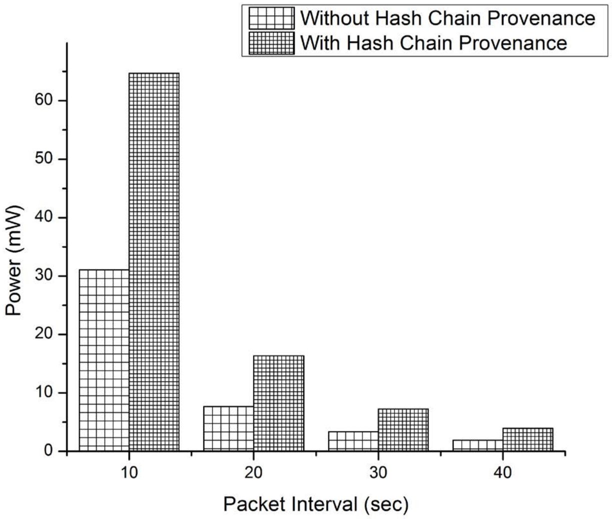

Fig. 7. Average Power Consumption per Node

VIII. Conclusion

We addressed the problem of data trustworthiness for IoT. We proposed a light-weight provenance embedding scheme that keeps on tracking the data packet by attaching the hash of traversed node id. The sink node verifies the data packet path to ensure the integrity of provenance data. Experimental results show that the proposed scheme is effective and light-weight for resource-constrained IoT. In future work, we plan to extend our work for case II where the nodes have to forward aggregated data as provenance information.

Acknowledgment

This work was supported by the Industrial Core Technology Development Program(10049079, Development of Mining core technology exploiting personal big data) funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea).

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2015-0-00274, Development of Smart Mediator for Mashup Service and Information Sharing among ICBMS Platform).

References

[1] https://www.symantec.com/security-center/threat-report

[2] http://www.reuters.com/article/us-medtronic-cybersecurityidUSTRE79O8EP20111025

[3] Kushner, David. ”The real story of stuxnet.” ieee Spectrum 3.50 (2013): 48-53.

[4] Hasan, Ragib, Radu Sion, and Marianne Winslett. ”The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance.” FAST. Vol. 9. 2009.

[5] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R.Struik, J. Vasseur, R. Alexander, ”RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks”, RFC 6550, March 2012.

[6] A. Dunkels, B. Grnvall, T. Voigt, ”Contiki a lightweight and flexible operating system for tiny networked sensors”, in: EMNets04, Tampa, USA, 2004, pp. 455462.

[7] Shreenivas, Dharmini, Shahid Raza, and Thiemo Voigt. ”Intrusion Detection in the RPL-connected 6LoWPAN Networks.” Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security. ACM, 2017.

[8] Zafar, Faheem, et al., ”Trustworthy Data: A Survey, Taxonomy and future trends of Secure Provenance Schemes.” Journal of Network and Computer Applications (2017).

[9] B. Shebaro, S. Sultana, S. R. Gopavaram, and E. Bertino, Demonstrating a lightweight data provenance for sensor networks, in Proc. ACM Conf. Comput. Commun. Security, 2012, pp. 10221024.

[10] Sultana, Salmin, et al., ” A lightweight secure scheme for detecting provenance forgery and packet dropattacks in wireless sensor networks”,IEEE transactions on dependable and secure computing 12.3 (2015): 256-269.

[11] Wang, Changda, Syed Rafiul Hussain, and Elisa Bertino. ”Dictionary based secure provenance compression for wireless sensor networks.” IEEE transactions on parallel and distributed systems 27.2 (2016): 405-418.

[12] Sultana, Shabana, Mohamed Shehab, and Elisa Bertino. ”Secure provenance transmission for streaming data.”Knowledge and Data Engineering, IEEE Transactions on 25.8 (2013): 1890-1903.

[13] Suhail, Sabah, et al., ”Introducing Secure Provenance in IoT: Requirements and Challenges.” Secure Internet of Things (SIoT), 2016 International Workshop on. IEEE, 2016.

[14] Bagci, Ibrahim Ethem, et al. ”Codo: Confidential data storage for wireless sensor networks.” Mobile Adhoc and Sensor Systems (MASS), 2012 IEEE 9th International Conference On. IEEE, 2012.

[15] Osterlind, Fredrik, et al. ”Cross-level sensor network simulation with cooja.” Local computer networks, proceedings 2006 31st IEEE conference on. IEEE, 2006.

[16] Lodhi, M. Ali, et al. ”Transient Multipath routing protocol for low power and lossy networks.” KSII Transactions on Internet and Information Systems (TIIS) 11.4 (2017): 2002-2019.

[17] Tmote Sky Datasheet http://www.sentilla.com/pdf/eol/tmote-sky-datasheet.pdf.

[18] Raza, Shahid, Linus Wallgren, and Thiemo Voigt. ”SVELTE: Real-time intrusion detection in the Internet of Things.” Ad hoc networks 11.8 (2013): 2661-2674.