A Mobile Code-driven Trust Mechanism for Detecting Internal Attacks in Sensor Node-powered IoT

Noshina Tariq[a], Muhammad Asim[a,*], Zakaria Maamar[b], M. Zubair Farooqi[a], Noura Faci[c], Thar Baker[d]

[a]*Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Pakistan*
[b]*College of Technological Innovation, Zayed University, Dubai, U.A.E*
[c]*Department of Computer Science, LIRIS, Claude Bernard Lyon 1 University, Lyon, France*
[d]*Department of Computer Science, Liverpool John Moores University, Liverpool, UK*

Abstract

The ubiquitous use of Internet-of-Things (IoT) is enabling a new era of wireless Sensor Nodes (SNs) that can be subject to attacks like any other piece of hardware and software. Unfortunately, an open and challenging issue is to what extent legitimate SNs can be trusted. This paper presents an energy-efficient, software-defined-network-based Mobile Code-driven Trust Mechanism (MCTM) for addressing this issue by assessing trust of SNs based on their forwarding behaviors. MCTM uses mobile code to visit the SNs based on pre-defined itineraries while collecting necessary details about these SNs in preparation for assessing their trust. The results gained from the experiments demonstrate a superior performance over a state-of-art technique that is energy-efficient management based on Software-Defined Network (SDN) for SNs. Message overhead is reduced by approximately 50%, which results in consuming less energy when detecting malicious SNs.

Keywords: Energy, Internet of Things, Mobile code, Sensor node, Trust.

*Muhammad Asim
Email addresses: i131502@nu.edu.pk (Noshina Tariq), muhammad.asim@nu.edu.pk (Muhammad Asim), zakaria.maamar@zu.ac.ae (Zakaria Maamar), zubair.farooqi@nu.edu.pk (M. Zubair Farooqi), noura.faci@liris.cnrs.fr (Noura Faci), t.baker@ljmu.ac.uk (Thar Baker)

1. Introduction

Internet-of-Things (IoT) is an ever-growing technology that aims at offering ubiquitous access (anytime, anywhere) to a plethora of devices (e.g., sensors, actuators, and controllers) over the Internet. IoT is the backbone of many smart applications related to process automation, traffic monitoring, unmanned vehicles, to cite just a few (e.g., [1] and [2]). CISCO anticipates that because IoT will become omnipresent through an expected number of 30 billion connected devices in 2020, there will be a lot of "disruption" in many fields like business, healthcare, and energy [3]. To respond to this predictable "tsunami" of things, IT infrastructure should support all the necessary technologies that would allow to meet the 21^{st} century applications' requirements when it comes to low latency, better privacy, and access ubiquity. Such technologies could be cloud, fog, blockchain, 5G networks, etc.

In addition to the "tsunami" of things, the trend of making unattended wireless networks accessible anywhere, anytime, and to everyone is now a reality in many cities like Tokyo in Japan [4] and Seoul in South Korea [5]. This is happening thanks to Wireless Sensor Networks (WSN) that allow running unattended operations, deploying unplanned devices, and monitoring different environments [6]. This trend is now sustained by IoT that could power a new generation of wireless Sensor Nodes (SN) in terms of autonomy, efficiency, and security [7]. However, SN-powered IoT not only bring new opportunities, but, raises many concerns that cyber-criminals are taking advantage of. Like IoT devices that can and regularly get hacked [8], SNs can be subject to the same unfortunate "fate".

Existing security solutions cannot cater to all the security needs and requirements of SN-powered IoT applications. For instance, cryptography techniques (such as data encryption, identity authenticated key-agreement, and digital signature) can prevent external attacks, but are inefficient when SNs are already embedded into an IoT application that considers these SNs as legitimate [9]. A SN that is already authenticated and is part of an IoT application could mis-

behave despite the positive authentication. Moreover, SNs are largely deployed on Low power and Lossy Networks (LLN) that have limited power, memory, and processing. This negatively impacts LLN-based applications due to high loss rates, low data rates, and instability (e.g., [10] and [11]). Besides security, another critical factor in a WSN's lifespan is energy consumption of SNs. Technical restrictions prevent the adoption of conventional security solutions that are known for their high levels of energy consumption [11, 12]. Therefore, it becomes inevitable to trade-off between energy consumption and proper security.

To address the above concerns and limitations, we resort to trust [13] and mobile code [14] to "single out" malicious SNs and reduce sensitive data transfer, respectively, while considering LLNs' characteristics. Both solutions are encompassed in an energy-efficient, software-defined-network-based Mobile Code-driven Trust Mechanism (MCTM). Trust establishes a contextual confidence level about each SN prior to including/excluding it in/from an IoT application. And, mobile code "visits" SNs so they locally (and not remotely over the network) collect necessary details that allow defining the confidence level of these SNs. We design mobile codes pre-defined itineraries that consist of visiting edge-based facilities to which SNs are connected. Upon arrival to these facilities, mobile codes collect (and sometimes pre-process) necessary details from the SNs and then, continue their roaming from one facility to another until they return back to their initial bases loaded with details. These ones will be used to develop preventive actions that would boost the security of "good" SNs and isolate the "bad" ones. Below are our main contributions:

1. An energy-efficient edge-based architecture for analyzing SNs' behaviors.
2. A novel mobile code-based mechanism for trust assessment that detects and isolates suspicious SNs.
3. A proof-of-concept of trust assessment along with some benchmark results.

The rest of the paper is organized as follows. Section 2 is an overview of security and trust in SN-powered IoT. The architecture associated with MCTM

is discussed in Section 3. The experimental set up and results are articulated in Section 4. Section 5 presents related work. Finally, Section 6 concludes the paper and identifies some future work.

2. Background

This section first, discusses the security challenges when mitigating internal attacks and then, discusses trust in a SN-powered IoT context.

2.1. Security challenges in SN-powered IoT

Data confidentiality, integrity, authenticity, and availability are mandatory security requirements in SN-powered IoT applications that are vulnerable to both external and internal attacks. Compared to external attacks, internal **(focus of this work)** are more severe/damaging since the SNs that have already acquired legal identities and possess privileged access rights, are "hijacked" and controlled by attackers making them misbehave, for example. Therefore, mandatory security requirements must be satisfied in order to provide appropriate security in SN-powered IoT applications. However, the unique characteristics of WSNs make security a real challenge when mitigating internal attacks on SN-powered IoT applications. Some challenges are discussed below:

- *SN deployment.* SNs often run in environments where an attacker can physically approach and capture them for malicious purposes. The attacker can read a SN's memory and collect all the stored credentials like cryptographic keys and identities [15]. This would further allow the attacker to control the SN in order to eavesdrop the transmitted messages or affect the network functionality in term of breaching its confidentiality, integrity, and availability.

- *Resource limitation.* Bandwidth, computing power, and battery power limitations in (mobile) SNs may lead to trade-off between security and consumption of resources like energy [12]. This leads to possible security breaches making room for potential attacks.

- *SN heterogeneity.* Since SNs range from simple sensors to sensor-embedded smart things with different power consumption and energy efficiency levels, some are expected to fulfill more responsibilities in WSN that turn out "critical" (single points of failure). Moreover, SNs are often developed with built-in security, which are designed based on SN hardware specifications [16]. Thus, the security mechanism of one type of SN may not work or be compatible with the security mechanism of another type of SN.

- *Unreliable communication medium.* Wireless networks are inherently less secure than their wired counterparts. They are open and accessible medium and hence, vulnerable to transmission interceptions, replay, and alterations [17]. Similarly, adversaries may also either inject malicious data packets or replace valid ones in wireless medium to breach data confidentiality, integrity, and availability. Even though the transmission medium is often potentially secured, an attacker can still get access to the wireless medium so, that, she captures/intercepts key messages to gather sensitive information about the SNs.

2.2. Trust in SN-powered IoT

In recent years, trust-based security (*aka* collaborative or soft security) has been the focus of IoT industry and academia in the view of the above-mentioned challenges. The purpose is to detect and isolate malicious components, SNs in our case, that are approved to participate in IoT applications based on their legitimate identities [9]. To avoid such a situation, SNs "keep an eye" on their neighbor SNs so, that, possible deviations from acceptable behaviors (e.g., safety, correctness, reliability, and availability [18]) are detected and hopefully reported to the relevant authority. Consequently, SNs' trustworthiness to handle future operations can be predicted based on past observations.

Traditionally, trust management in WSN consists of the following stages [19]: (i) *detail gathering* (i.e., how nodes collect information about relevant peers), (ii) *detail modeling* (i.e., how nodes represent direct and/or indirect opinions

about other peers (e.g., statistics or probabilities) in WSN), (iii) *detail dissemination* (i.e., how nodes share information with peers in terms of content, frequency, and locality), and (iv) *misbehavior detection and response* (i.e., what are the trust metrics used to identify misbehavior and what kind of punishment/reward mechanisms should be used). Trust can be computed into 2 ways [13]: local and global. The former is based on direct communications between 2 neighboring nodes. The latter is defined by a central entity that collects local trust information from nodes in the ecosystem. A trust value is, thus, maintained by either the neighboring devices (local trust) or the central entity (global trust) and is used to decide whether a SN is eager to perform its intended operation normally in the network [13, 20]. Some thresholds are set for the SNs to be tagged as either "good" or "bad". For instance, an already authenticated SN maliciously behaves by blocking all the packets of sensed and/or actuated data that it receives instead of forwarding them (*aka* blackhole attack). This malicious behavior can be detected by the surrounding neighboring SNs based on direct communications' observations.

In this paper we focus on malicious forwarding attacks, which can be confronted by a trust-based mechanism that would ensure communication reliability, correctness, and availability. Malicious forwarding attacks (e.g., selective and delayed forwarding) deteriorate network data delivery ratio by dropping data packets instead of passing them on and re-sending undelivered packets results in more energy consumption. Examples of such attacks include blackhole and grayhole. In the former, the malicious node drops all the packets it receives. This leads to performance degradation and excessive power drainage due to lost packet re-sending. In the latter, only few data packets are forwarded to avoid detection.

3. Trust and mobile code for safer SNs

This section defines the assumptions linked to defining MCTM, and, then, presents the proposed mobile code-driven architecture for assessing trust of SNs.

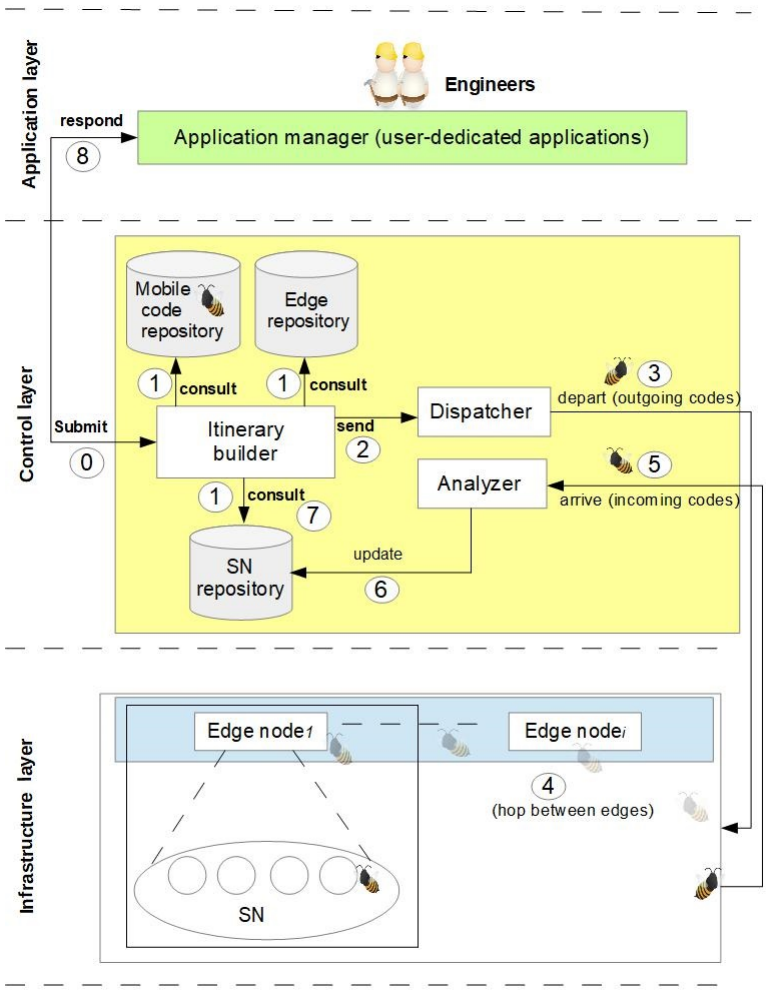

Figure 1: 3-layer architecture in support of MCTM

3.1. Assumptions

Prior to proceeding with detailing our MCTM, the following assumptions are made:

1. SNs in a network are homogeneous and stationary, associated with unique identifiers, and deployed randomly.
2. SNs have exchanged enough data packets to know each other so, that,

each SN can compute some statistics about its neighbors' behaviors using past successful and unsuccessful forwarding of data packets.

3. Mobile code can not be tampered. The afore-mentioned assumptions are deemed necessary in order to narrow down the issues to address and objectives to achieve. Indeed, a tampered mobile code would require securing this code, which does not fall into this work's scope.

3.2. Architecture

Fig. 1 illustrates the architecture that supports analyzing SNs trust behaviors and rectifying network management behaviors (by detecting and isolating the malicious ones), should suspicious signs be detected. The mechanism is built-upon 3 layers, application, control, and infrastructure, with focus on the last 2 in this paper.

Application layer targets system engineers who have needs to satisfy like configuring SNs, vetting SNs, isolating SNs, etc. The engineers submit their needs to the control layer's *itinerary builder* module that reports back to them once the needful is done.

Control layer consists of 3 modules and 3 repositories that are:

- The *itinerary builder* module identifies the different edge-based facilities (edge, for short) that mobile codes need to visit along with the SNs that these codes need to interact with when satisfying some of the engineers' afore-mentioned needs. The *itinerary builder* module uses 3 repositories: *mobile code* containing mobile codes that will be initialized (e.g., destinations and order of visits) in preparation for their departure to the infrastructure layer, *edge* containing technical details about edges like location, capabilities, and access credentials, and, finally, *SN* containing technical details about SNs like location, residual energy, coverage, and evolving trust value.
- The *dispatcher* module makes mobile codes depart from the control layer on their way to different edges along with tracking the progress

of completing the itineraries that the *itinerary builder* module has developed. Changes to mobile codes' itineraries like visiting other edges or dropping some are taken care by the *dispatcher* module in collaboration with the *itinerary builder* module, should some risks be detected at some SNs or should engineers revise their needs, for example.

- The *analyzer* module "debriefs" mobile codes upon their returns from visiting the different edges and interacting with their respective SNs. This debriefing leads to updating the *SN* repository allowing the *itinerary builder* module to include these updates when designing next mobile-code itineraries. Trust calculation also happens during the debriefing as per the details that mobile codes would have carried on the way back.

Infrastructure layer is at the lowest level of the architecture and hosts necessary equipment that are specialized into edge nodes and SNs. Some benefits of using edge-based (also referred to as fog) computing are thoroughly discussed in [21], such as minimizing data transfer to distant sites and avoiding data exposure to unnecessary risks like interception and alteration. It is recommended to have edge nodes located "close enough" to SNs to promote local *instead of* distant interactions when receiving their data for management needs. Edge nodes also act as platforms for mobile codes that arrive/depart from/to other edges after collecting SNs' details and make instructions available to mobile codes, should the *dispatcher* module decide to update these codes' itineraries. Edge nodes are also responsible for pre-processing the data collected about SNs at the infrastructure layer using edge nodes, as shown in Fig.2.

3.3. Trust assessment

We discuss the stages that guide mobile codes collect details about SNs so, that, trust assessment happens. We aligned these stages to those presented

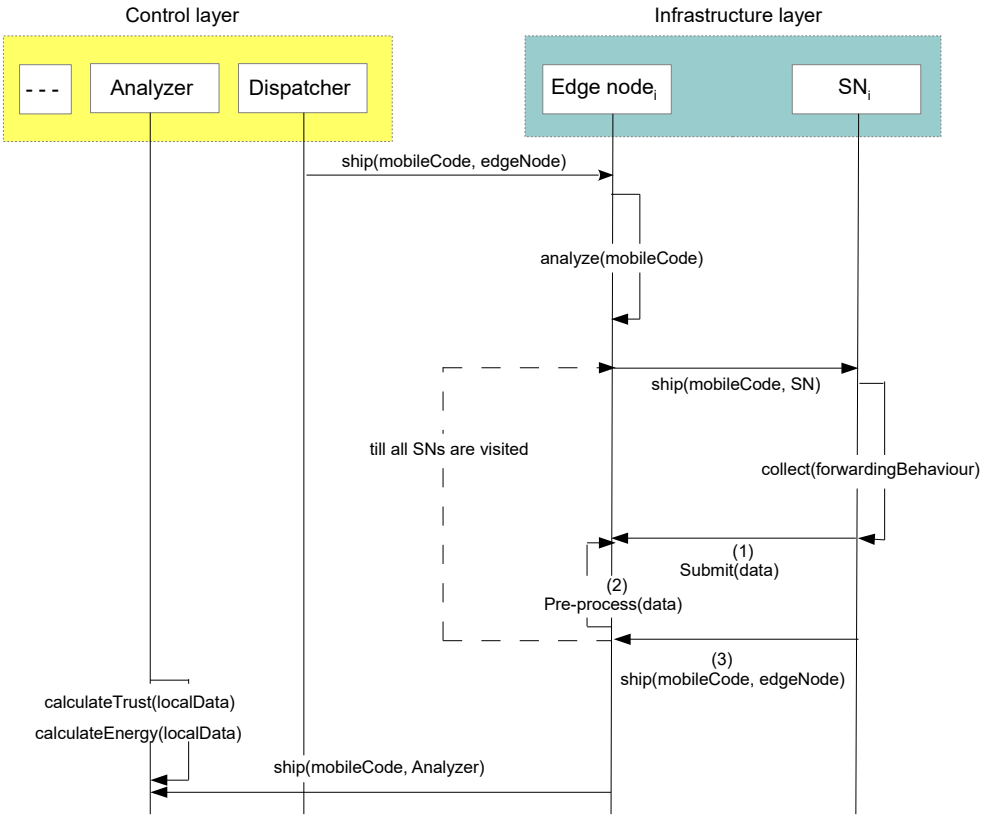

Figure 2: Detail collection for trust assessment in MCTM

in [13] (Section 2.2).

Gathering stage. To assess trust, SNs maintain up-to-date details about their neighboring nodes' forwarding behaviors. The details are required for sending and receiving data packets to/from the control layer. During node-to-node interactions, every node records about a neighbor its behavior in term of either forwarding or dropping the data packets. Thus, the node computes certain statistics like *successful communication*, *unsuccessful communication*, and *energy consumption* about its neighbors in our case. In Fig. 3, when SN_2 forwards SN_1's packet to SN_4 via SN_3, the trust mechanism of SN_2 monitors SN_3's forwarding behavior. Assumption made is that all nodes communicate via a shared wireless medium and operate in the promiscuous mode [13, 22]. Thus, if SN_2 "hears" that SN_3 has successfully forwarded the packet to SN_4, the statistics about SN_2 (e.g., standard deviation) are updated, accordingly. These statistics are also updated in the opposite case, i.e., SN_3 not forwarding data packets. When mobile codes arrive to edges and communicate with their respective SNs, they collect such statistics along with the energy parameter and submit these statistics to the *analyzer* module upon return from their visits. Equation 1 shows the calculation of energy trust assessment. It is pivotal to detect if a malicious SN intensely consumes energy compared to a benign SN.

$$E_r = \frac{E_t - E_{t_1}}{t - t_1} \tag{1}$$

Where E_r represents the residual energy of a SN, E_{t_1} is the residual energy at time t_1 and E_t is the residual energy at time t. In a scenario like DoS attack, it is expected that the value of E_r decreases eminently.

Computing stage. To calculate trust, the *analyzer* module uses Subjective Logic Framework (SLF) [23]. SLF has been extensively used in the literature like [24] and [25] to allow realistic modelling of real-world scenarios

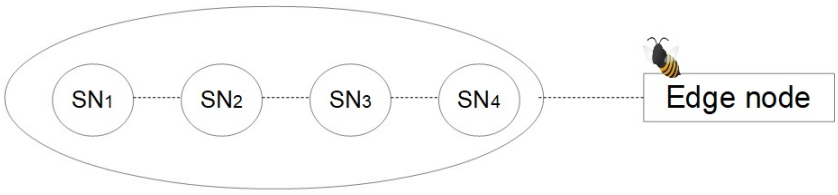

Figure 3: A topology example of connected SNs

with a better reflection of ignored and uncertain values that result from uncertain inputs. SLF uses opinions (arguments in subjective logic) instead of binary or probabilistic values for significant expressiveness. In the subjective logic, the degrees of uncertainty, ignorance, and lack of information are explicitly taken into account and can be articulated in conclusions [26].

There are 3 categories of trust in SLF [27]: belief (b), disbelief (d), and uncertain (u). Belief is the level of trust in the reliability of one entity. Disbelief is reciprocal to belief. And, uncertainty determines whether, in a given context, trust is integral or not. These values are represented in an *opinion triangle* and fall into 0 and 1 range so, that, their sum is equal to 1, i.e., b, d and $u \in [0,1]$: $b + d + u = 1$. For example, a node is believed to be trustworthy if the value of b, that is derived from the statistics received through mobile codes, is greater than the specified threshold. The computation of trust is given in Equations 2, 3, and 4 where successful/unsuccessful communication is denoted by p/n and k is a constant set to 1 to avoid division by 0 during computation.

$$b = p/(p + n + k) \tag{2}$$

$$d = n/(p + n + k) \tag{3}$$

$$u = k/(p + n + k) \tag{4}$$

For effective trust calculation, we advocate for combining trust values. Since SNs are dynamic in nature (i.e., join and leave networks without prior notice), their malicious behaviors may also fluctuate with time. For instance, at time t, a SN misbehaved and the reported trust reflected this behavior as malicious. In the next time interval *t+1*, it behaved normally. Thus, the reported trust accounts this behaviour as normal. Furthermore, in wireless medium, signals may collide and affect the data packets. Thus, the recorded behaviour based on the collected trust may fluctuate, which can be tackled by adding the history trust parameter $T_{history}$. This parameter is added to the direct trust T_{direct} calculation for the credibility and normalization of current trustworthiness of a SN. The addition of two trust values minimizes the error rate and resource consumption for effective trust calculation of SNs [28]. Based on Equation 5, the *analyzer* at the control layer calculates direct trust and makes decision about the malicious SN and may take necessary actions.

$$T_{direct} = w_1 T_{current} + w_2 T_{history} \tag{5}$$

The current trust $T_{current}$ and past trust $T_{history}$ determine T_{direct} using weights w_i, where w_1 and w_2 are the weights given to $T_{current}$ and $T_{history}$, respectively and $T_{current}$ and $T_{history}$ are the current and previously calculated values of b. For weighted values, $w_1 + w_2 = 1$. The statistics collection period among nodes is maximized in a reasonable range to minimize the node energy consumption in data transmission. In this case, the trust value may turn out to be too old to really reflect the current state of a node. So we calculate the weight of $T_{history}$ by Equation 6.

$$w_2 = r_1 * t_{network} * exp(-r_2 * t_{network}) \tag{6}$$

Where $t_{network}$ is the time interval from the last update till now. r_1 and r_2 are two real numbers that are used to simplify calculations.

4. Experiments

This section discusses our experimental setup and results with respect to message overhead, network lifetime, and energy consumption. The details are given in the following subsections.

4.1. Experimental setup

A prototype system is implemented based on SDN-WISE [29] to compare the performance of MCTM with the work presented in [13] as it also mitigates malicious forwarding attacks in Software-Defined WSNs. A light-weight SLF trust model has been used in each Software Defined Wireless Sensor Network (SD-WSN) node along with a modified Cooja platform[1] for the implementation of a data plane proposed in Contiki 2.7 [30]. A laptop computer is used as a control layer (the controller), equipped with an Intel(R) Core(TM) i5-6200U with 16GB DDR4 RAM. The performance of MCTM is evaluated with several simulation tests under both small and large-scale networks. We deployed 10 to 50 SNs in the network with the same initial energy of 100 J, where the data packets are randomly exchanged among nodes. The number of malicious nodes also varies from 5 to 20, where some nodes drop all the received packets carrying out blackhole attacks and some drop packets selectively to perpetrate grayhole attacks as well. To evaluate the performance of both MCTM and (ETMRM) [13] under various workloads, the data packet generation interval is set between 2 to 25 seconds and the simulation runs 10 to 15 times. For generating the same destination addresses, we use a pseudo-random technique for both MCTM and ETMRM.

4.2. Measuring message overhead

Message overhead is the ratio of message exchange between 2 nodes to all the messages exchanged in the network. The normalized message overhead ratios of

[1] An open source operating system that provides a controller communication interface and is used for the simulation of both IoT and WSNs.

the 2 schemes (MCTM and ETMRM) are shown in Fig. 4. MCTM reduces the message overhead approximately by 50% than ETMRM. The message exchange between nodes is fewer in MCTM when compared to ETMRM with the same number of data packets. This is due to the fact that in MCTM, mobile code hops in the WSN from edge to edge for data collection.

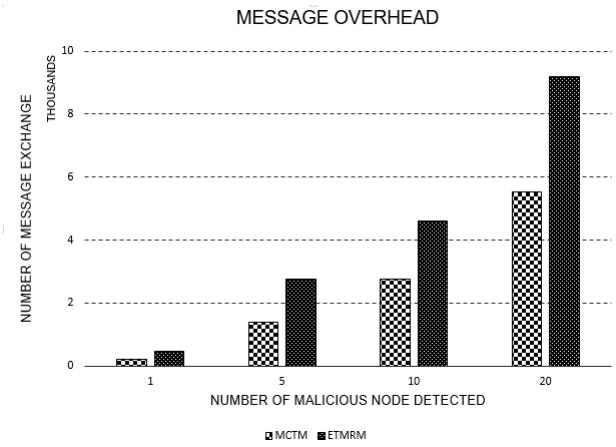

Figure 4: Message overhead in MCTM and ETMRM

Fig. 5 illustrates the time taken to detect malicious forwarding attacks; blackhole and grayhole in our case. Detection time varies in blackhole attack and grayhole attack for equal number of malicious nodes. As discussed earlier, blackhole attack drops all the packets and are somehow easy to detect. We purposely introduced a number of malicious nodes chronologically and observed the detection time. We introduced 1, 5, 10, and 20 malicious nodes progressively, after random time intervals. MCTM detected all the malicious nodes successfully in grayhole attack with a mean time of 2.45s approximately for each one of the 20 nodes, but when it comes to detection of malicious nodes in blackhgole attack, our technique took a mean detection time of only 1.4s while detecting same number of malicious node as above. However, the detection of the first node took more time due to lack of trust establishment on introduction of only 1 malicious node but after collecting enough trust details, the detection time

dropped significantly. For instance, the time taken to detect one malicious node was 8 and 10 seconds for blackhole and grayhole attack, respectively. However, it was approximately 28 and 49 seconds for detecting 20 malicious nodes.

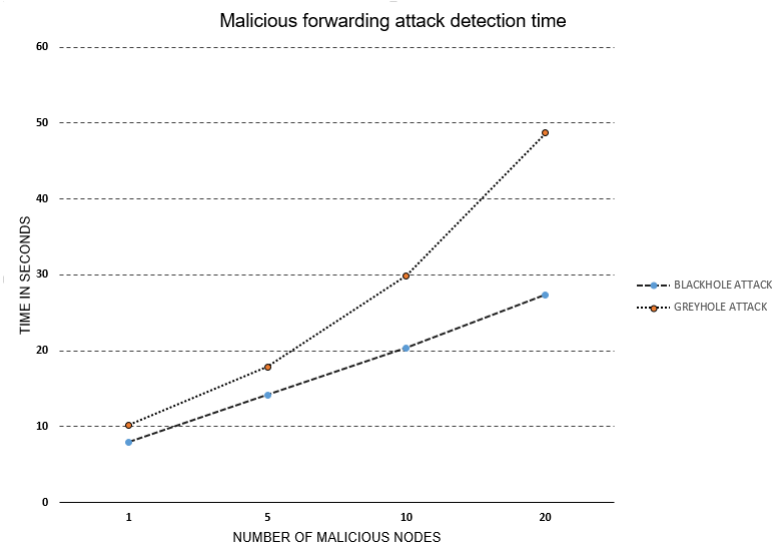

Figure 5: Detection time of malicious forwarding attacks

Table 1: Simulation parameters

Simulation environment	**Simulation details**	
	Parameter	**Value**
Network	Network Size	400m x 400m
	Number of Nodes	10 - 50
	Sink Node	[200m x 200m]
	Initial Energy	100 J
	Simulation Time	300 s
	Trust calculating Frequency	Adaptive
Energy Consumption Model [13]	E_{rx}	0.0009 mJ/bit
	E_{tx}	0.0010875 mJ/bit
	Standby Power	0.708 mJ/bit

4.3. Network lifetime and energy model

A SN whose energy drains completely is considered dead. The time when the first node dies is defined as the lifetime of the whole network. Fig. 6 shows the average network residual energy while detecting different numbers of malicious nodes. In this figure, the average remaining energy of the whole network is more than ETMRM, detecting same number of malicious nodes. Parameters associated with initial energy, simulation time, and energy consumption model, are given in Table 1. We selected these parameters in order to remain consistent with the simulation and experiment environment of [13]. While detecting less number of malicious nodes, both techniques have equal lifetime. However, on increased number of malicious node detection our technique outperforms ETMRM.

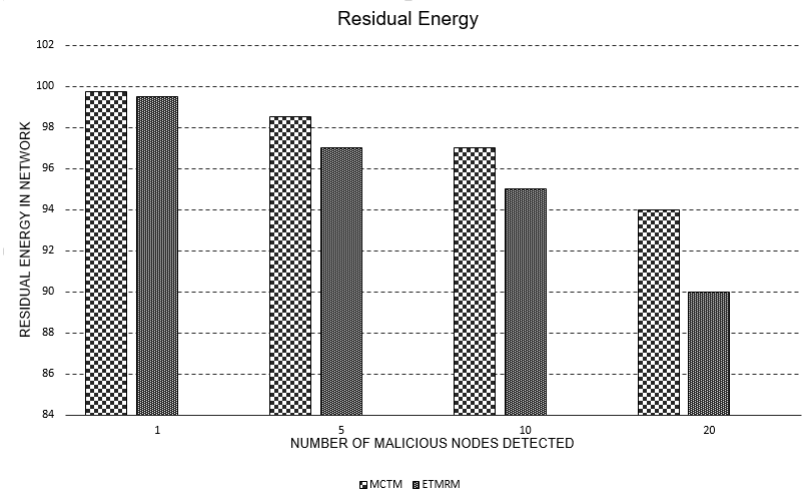

Figure 6: Residual energy in MCTM and ETMRM

Fig. 7 and Fig. 8 illustrate the 3D representation of overall node-level energy distribution of 2 schemes MCTM and ETMRM, after detecting maximum number of malicious nodes. The average residual energy of 50 SNs is 94% in MCTM and 90% in ETMRM. MCTM shows more balanced and lesser energy consumption than ETMRM due to less message overhead and computations at

the node level.

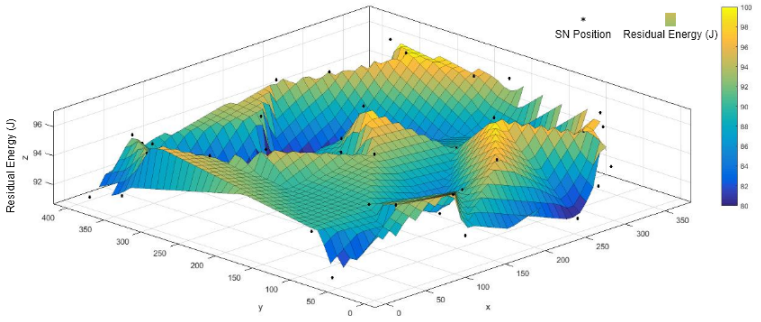

Figure 7: Energy distributions in MCTM

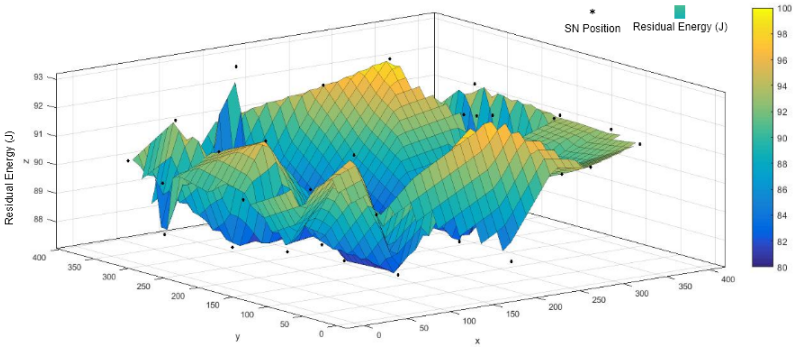

Figure 8: Energy distributions in ETMRM

The overall experimentation results affirm that the message overhead is reduced, less control energy is consumed, and network lifetime is optimized, consequently.

5. Related work

Several trust assessment techniques in the context of SNs are reported in the literature [7, 31, 32]. Tajeddin et al. [33] propose CENTERA, a centralized trust-based routing protocol with an integrated cryptographic-based authentication mechanism for SNs. CENTERA makes use of a powerful base station to

periodically gather trust information from nodes and to calculate the optimal routes after identifying and excluding malicious nodes. However, as analyzed in [13], CENTERA nodes are not "smart" enough to make decisions about identifying and isolating internal malicious nodes. Moreover, the periodic exchange of trust information with the base station, and the requirement of encryption and decryption at each intermediate node causes too much computation and message overhead.

Other works focus on trust in fog and Software-Defined Networks (SDN) for IoT applications. Galluccio et al. [34] proposed SDN-WISE to reduce the number of packets between SNs and the SDN network controller and to make SNs programmable so, that, they can be operated with the support of stateless solutions. Besides SDN-based mechanisms, Jiang et al. proposed an Efficient Distributed Trust Model (EDTM) for WSN [20]. In EDTM, the values of direct trust and recommendation trust are selectively calculated by a SN, based on the number of packets it receives. When calculating direct trust, communication-, energy-, data-related trust are all considered. Additionally, trust reliability and familiarity are defined to enhance the accuracy of recommendation trust. In [13], ETMRM is developed to detect and block malicious forwarding attacks such as grayhole, blackhole, and new-flow. At the node level, Wang t al. [13] proposed a trust monitoring and evaluation scheme to extend SensorFlow tables along with a centralized trust management for malicious node detection and isolation at the controller level. In addition, they considered residual energy and ensured control traffic transmission. In [22], suggested a light-weight trust monitoring and evaluation scheme at the node level and centralized trust management scheme at the SDN controller level. The work focuses on packet delivery-ratio and balances energy consumption. The trust is calculated both at the node level and at the controller level. This technique fails to work efficiently when the trust calculation frequency is high.

Wang et al. [9] proposed a fog-based hierarchical trust-based mechanism for SDN, which has two distinctive features: trust in network structure and trust between cloud service providers and sensor service providers. They focused

on the packet loss rate, route failure rate, and forwarding delay only. Elmisery et al. [35] suggested a fog-based middleware where trust between a fog node and the cloud is calculated in a decentralized fashion using entropy definition.

Recently, the interest in mobile agent-based WSNs is growing significantly. El Fissaoui et al. [36] proposed a novel energy-aware data aggregation itinerary planning mechanism among cluster head, based on mobile agents for WSNs. Likewise, in [37] Ioannis et al. benchmarked some renowned itinerary planning algorithms through simulations. Yuan et al. [38] proposed a mobile-agent based event-driven algorithm for gathering data in chain-based WSNs with reduced network delay. Apart from data aggregation, mobile agents are also used in detecting several attacks, such as Hada et al. [14] who proposed a mobile agent-based secure trust architecture for cloud. The agent captures data from virtual machines to provide integrity, authenticity and data security.

The afore-mentioned paragraphs discuss trust for SNs from different perspective. However, they overlook the perspective of computation and energy overhead caused due to exchange of messages between distant components for trust calculation. We also "shield" data exchange from potential attacks since edges are expected to be close to data sources (SNs in our case). Moreover, the use of mobile codes addresses issues of data aggregation in WSNs that has been a subject of interest in the recent years. Nevertheless, we argue that the potential of mobile codes has not been fully explored in term of addressing security concerns in SN-powered IoT applications. Table 2 presents a comparison between the proposed MCTM and some other state-of-the-art techniques, where $\downarrow$ presents a *'Low'* value and $\uparrow$ presents a *'High'* value.

Table 2: Comparison between MCTM and other techniques.

Reference	Message Overhead	Energy Efficient
MCTM	↓	↑
[9]	↑	↑
[13]	↑	↓
[14]	↑	↓
[20]	↑	↓
[22]	↑	↓
[33]	↑	↓
[34]	↑	↓
[35]	↑	↑
[36]	↓	↑
[38]	↓	↑

6. Conclusion

SNs are known for being resource-constrained and vulnerable to both external and internal attacks. Security needs of SN-powered IoT applications cannot be catered entirely with existing security solutions. For instance, cryptography can prevent external attacks, but are not equally useful in internal attacks when SNs use legitimate identities to engage in malicious activities. This paper proposed an energy efficient Mobile Code-driven Trust Mechanism (MCTM) for detecting and isolating malicious internal SNs in SN-powered IoT applications. Mobile codes are deployed over these applications to collect details about each SN. They crawl over the network based on pre-defined itineraries and collect necessary details about SNs that help in establishing the trust level. The proposed MCTM effectively deals with malicious forwarding attacks, such as blackhole and grayhole. The results show that MCTM improved residual energy and prolonged network lifetime when compared to state-of-art techniques like ETMRM. Further development would focus on mitigating routing attacks like Sybil, sink

hole, and wormhole for resource-constrained SNs.

References

[1] T. Baker, M. Asim, H. Tawfik, B. Aldawsari, R. Buyya, An Energy-aware Service Composition Algorithm for Multiple Cloud-based IoT Applications, Journal of Network and Computer Applications 89 (2017) 96–108.

[2] J. Ni, K. Zhang, X. Lin, X. S. Shen, Securing Fog Computing for Internet of Things Applications: Challenges and Solutions, IEEE Communications Surveys Tutorials 20 (1) (2018) 601–628.

[3] Cisco, Internet of Things (IoT) Data Continues to Explode Exponentially. Who Is Using That Data and How? (2018).

[4] K. J. Fietkiewicz, W. G. Stock, How" Smart" Are Japanese Cities? An Empirical Investigation of Infrastructures and Governmental Programs in Tokyo, Yokohama, Osaka, and Kyoto, in: 2015 48th Hawaii International Conference on System Sciences, IEEE, 2015, pp. 2345–2354.

[5] U. Gretzel, J. Ham, C. Koo, Creating the City Destination of the Future: The Case of Smart Seoul, in: Managing Asian Destinations, Springer, 2018, pp. 199–214.

[6] B. Rashid, M. H. Rehmani, Applications of Wireless Sensor Networks for Urban Areas: A survey, Journal of network and computer applications 60 (2016) 192–219.

[7] Y. Saied, A. Olivereau, D. Zeghlache, M. Laurent, Trust Management System Design for the Internet of Things: A context-aware and Multi-service Approach, Computers & Security 39 (2013) 351–365.

[8] M. Tellez, S. El-Tawab, H. M. Heydari, Improving the Security of Wireless Sensor Networks in an IoT Environmental Monitoring System, in: 2016 IEEE Systems and Information Engineering Design Symposium (SIEDS), IEEE, 2016, pp. 72–77.

[9] T. Wang, G. Zhang, M. Z. A. Bhuiyan, A. Liu, W. Jia, M. Xie, A Novel Trust Mechanism based on Fog Computing in Sensor–Cloud System, Future Generation Computer Systems.

[10] J. P. C. Agustin, J. H. Jacinto, W. J. R. Limjoco, J. R. I. Pedrasa, IPv6 Routing Protocol for Low-power and Lossy Networks Implementation in Network Simulator 3, in: TENCON 2017 - 2017 IEEE Region 10 Conference, 2017, pp. 3129–3134.

[11] W. Trappe, R. Howard, R. S. Moore, Low-Energy Security: Limits and Opportunities in the Internet of Things, IEEE Security Privacy 13 (1) (2015) 14–21.

[12] N. Tariq, M. Asim, F. Al-Obeidat, M. Z. Farooqi, T. Baker, M. Hammoudeh, I. Ghafir, The Security of Big Data in Fog-Enabled IoT Applications Including Blockchain: A Survey, Sensors 19 (8) (2019) 1788.

[13] R. Wang, Z. Zhang, Z. Zhang, Z. Jia, ETMRM: An Energy-efficient Trust Management and Routing Mechanism for SDWSNs, Computer Networks 139 (2018) 119–135.

[14] P. Hada, R. Singh, M. Manmohan, Security Agents: A Mobile Agent based Trust Model for Cloud Computing, International Journal of Computer Applications 36 (12) (2011) 12–15.

[15] V. Manjula, C. Chellappan, The Replication Attack in Wireless Sensor Networks: Analysis and Defenses, in: International Conference on Computer Science and Information Technology, Springer, 2011, pp. 169–178.

[16] G. Panić, O. Stecklina, Z. Stamenković, An Embedded Sensor Node Microcontroller with Crypto-processors, Sensors 16 (5) (2016) 607.

[17] M. R. Ahmed, Protecting Wireless Sensor Networks from Internal Attacks, Ph.D. thesis, University of Canberra, [Online; accessed 02-April-2019] (2014).

[18] S. Becker, W. Hasselbring, A. Paul, M. Boskovic, H. Koziolek, Trustworthy Software Systems: A Discussion of Basic Concepts and Terminology, ACM SIGSOFT Software Engineering Notes 31 (2006) 1–18.

[19] H. Nunoo-Mensah, K. O. Boateng, J. D. Gadze, The Adoption of Socio- and Bio-inspired Algorithms for Trust Models in Wireless Sensor Networks: A survey, International Journal of Communication Systems 31 (7) (2018) e3444.

[20] J. Jiang, G. Han, F. Wang, L. Shu, M. Guizani, An Efficient Distributed Trust Model for Wireless Sensor Networks, IEEE transactions on parallel and distributed systems 26 (5) (2015) 1228–1237.

[21] Z. Maamar, T. Baker, N. Faci, E. Ugljanin, M. Al-Khafajiy, V. A. Burégio, Towards a Seamless Coordination of Cloud and Fog: Illustration Through the Internet-of-Things, in: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC'2019), Limassol, Cyprus, 2019.

[22] X. Li, F. Zhou, J. Du, LDTS: A Lightweight and Dependable Trust System for Clustered Wireless Sensor Networks, IEEE transactions on information forensics and security 8 (6) (2013) 924–935.

[23] A. Jøsang, A Logic for Uncertain Probabilities, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 9 (3) (2001) 279–311.

[24] P. Chaturvedi, A. K. Daniel, Trust Based Energy Efficient Coverage Preserving Protocol for Wireless Sensor Networks, in: 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), IEEE, 2015, pp. 860–865.

[25] J. Jiang, G. Han, C. Zhu, S. Chan, J. P. C. Rodrigues, A Trust Cloud Model for Underwater Wireless Sensor Networks, IEEE Communications Magazine 55 (3) (2017) 110–116.

[26] A. Jøsang, Subjective Logic, Springer, 2016.

[27] F. Cerutti, L. M. Kaplan, T. J. Norman, N. Oren, A. Toniolo, Subjective Logic Operators in Trust Assessment: An Empirical Study, Information Systems Frontiers 17 (4) (2015) 743–762.

[28] X. Wu, J. Huang, J. Ling, L. Shu, BLTM: Beta and LQI based Trust Model for Wireless Sensor Networks, IEEE Access.

[29] SDN-WISE, `http://sdn-wise.dieei.unict.it/`, [Online; accessed 02-April-2019] (2019).

[30] Contiki, `http://www.contiki-os.org/`, [Online; accessed 02-April-2019] (2019).

[31] L. Gue, J. Wang, B. Sun, Trust Management Mechanism for Internet of Things, China Communications 11 (2) (2014) 148–156.

[32] Y. Yu, K. Li, W. Zhou, P. Li, Trust Mechanisms in Wireless Sensor Networks: Attack Analysis and Countermeasures, Journal of Network and Computer Applications 35 (3) (2012) 867 – 880.

[33] A. Tajeddine, A. Kayssi, A. Chehab, I. Elhajj, W. Itani, CENTERA: A Centralized Trust-based Efficient Routing Protocol with Authentication for Wireless Sensor Networks, Sensors 15 (2) (2015) 3299–3333.

[34] L. Galluccio, S. Milardo, G. Morabito, S. Palazzo, SDN-WISE: Design, Prototyping and Experimentation of A Stateful SDN Solution for Wireless Sensor Networks, in: IEEE Conference on Computer Communications (INFOCOM), 2015.

[35] A. M. Elmisery, S. Rho, D. Botvich, A Fog based Middleware for Automated Compliance with OECD Privacy Principles in Internet of Healthcare Things, IEEE Access 4 (2016) 8418–8441.

[36] M. El-Fissaoui, A. Beni-Hssane, M. Saadi, Multi-Mobile Agent Itinerary planning-based Energy and Fault Aware Data Aggregation in Wireless Sensor Networks, EURASIP Journal on Wireless Communications and Networking 2018 (1) (2018) 92.

[37] E. I. Venetis, D. Gavalas, G. E. Pantziou, C. Konstantopoulos, Mobile Agents-based Data Aggregation in WSNs: benchmarking itinerary planning approaches, Wireless Networks 24 (6) (2018) 2111–2132.

[38] L. Yuan, X.Wang, J. Gan, Y. Zhao, A Data Gathering Algorithm based on Mobile Agent and Emergent Event-driven in Cluster-based WSN, Journal of Networks 5 (10) (2010) 1160.