Simulation Modelling Practice and Theory 101 (2020) 102041

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

Simulation of watchdog placement for cooperative anomaly detection in Bluetooth Mesh Intrusion Detection System☆

Mateusz Krzysztoń*, Michał Marks

NASK PIB, Kolska 12, 01–045 Warsaw, Poland

ARTICLE INFO

Keywords:
Internet of Things
Bluetooth Mesh
Simulation
Intrusion Detection System
Security
Watchdog
Placement
Cooperation
Fog computing

ABSTRACT

Cyber-attacks on the Internet of Things (IoT) are growing at an alarming rate as IoT technologies are literally connecting everything into networks. In 2020 more than 25% of identified attacks in enterprises will involve the IoT. Hence, it is very important to treat IoT security as a mandatory design factor. Moreover the deployed IoT system should be continuously monitored to detect malicious behaviour such as packet dropping, worm propagation or jammer attacks. In the paper we propose the anomaly based Intrusion Detection System dedicated to Bluetooth Mesh networks. The machine learning algorithm is used to classify traffic and detect malicious behaviour in IoT networks. The proposed solution involve cooperative decision making which is done by multiple watchdogs distributed in different regions of the considered network – which are responsible for processing of mostly local traffic. The optimal placement of watchdogs is proposed based on simulations done by *BMWatchSim* software. The experimental results coming from our testbed confirm that the watchdog placement proposed by simulator allow on effective detection of real-world intrusions.

1. Introduction

In recent years we have seen an explosion of Internet of Things (IoT) applications. From typical home automation through beacons in shopping centers up to automatically communicating cars. Diversities of IoT devices installed around us, such as smart doors, shared bikes and electric scooters or sweeping robots, gives us a feeling that IoT technologies are literally connecting everything into networks. And from the numbers, we can see that this feeling is becoming more and more reality. Gartner estimates that there will be over 20 billion connected things in use by 2020 [1]. Moreover machine to machine traffic is going to reach 45% of all Internet traffic by the end of 2022 [2].

On the other hand the same Gartner report [1] estimates that in 2020 more than 25% of identified attacks in enterprises will involve the IoT. What is interesting the Gartner's 2016 IoT Backbone Survey showed that 32% of IT leaders indicate security as a top barrier to IoT success. The IoT creates new security challenges for enterprises in both scope and scale. In the IoT world everyone can experience the attacks which are typical for IP networks or cloud systems. However the spectrum of possible attacks is much wider as the attacks may concentrate, for example, on perception layer where wireless sensors are used. The signals transmitted between IoT nodes using wireless technologies may be compromised by disturbing waves. Moreover, the sensor/IoT node can be intercepted not only by the owner but also by the attackers as the IoT nodes usually operate in external and outdoor environments, leading to

☆ Work done as part of the CYBERSECIDENT/369195/I/NCBR/2017 project supported by the National Centre of Research and Development in the frame of CyberSecIdent Programme.

* Corresponding author.
E-mail addresses: mateusz.krzyszton@nask.pl (M. Krzysztoń), michal.marks@nask.pl (M. Marks).

https://doi.org/10.1016/j.simpat.2019.102041
Received 22 July 2019; Received in revised form 9 December 2019; Accepted 10 December 2019
Available online 11 December 2019
1569-190X/ © 2019 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/BY-NC-ND/4.0/).

physical attacks on IoT sensors and devices in which an attacker can tamper the hardware components of the device [3].

Hence, it is very important to treat IoT security as a mandatory design factor. Moreover the deployed IoT system should be continuously monitored to detect malicious behaviour such as packet dropping, worm propagation or jammer attacks. It is especially important in the applications where the IoT devices are installed in groups communicating with each other trough varieties of wireless technologies (e.g., RFID, Wi-Fi, Bluetooth, ZigBee, LoRa, NB-IoT). In the paper we concentrate on simulation and experimental validation of monitoring solution for one of the latest wireless technologies i.e. Bluetooth Mesh (BM). The proposed watchdog based Intrusion Detection System (IDS) provides the means for detecting various kind of attacks e.g. gray hole, delay, injection or flooding attacks in the Bluetooth Mesh networks like office light management systems or patient monitoring platforms.

The contributions of this paper are threefold. First, novel IDS for detecting unknown attacks in BM networks by one or multiple watchdogs is proposed. Second, method for optimal placement of those watchdogs within the network is described. Third, new BM simulator, that enables quick optimal watchdogs positions determination in accordance with the aforementioned method, is presented. Quality of the IDS, the method for optimal placement and the simulator is verified with experiments in real BM network. Best to our knowledge this is the first work that investigates problem of efficient watchdog placement so thoroughly.

The outline of this paper is summarized as follows. The overview of Bluetooth technology including both Bluetooth 5.0, Bluetooth Mesh, Bluetooth security as well as Intrusion Detection Systems is presented in Section 2. In Section 3, the anomaly detection in Bluetooth Mesh, as a counteract to typical attacks is described. In Section 4, the application of multiple watchdogs to cooperatively detect anomalies is presented. The problem of optimal watchdogs placement is solved in Sections 5 and 6 where both the simulation software and experimental verification in laboratory is described. In Section 7, we discuss possible research directions and summarize the paper.

2. Overview of Bluetooth technology and its security

As it was mentioned in the introduction section there is a huge variety of wireless technologies (e.g., RFID, Wi-Fi, Bluetooth, ZigBee, LoRa, NB-IoT) which have been developed in the past decades and can be used to create the IoT solutions. However, since the idea of IoT covers so many different kinds of application scenarios, none of the existing technologies can fulfill the requirements of all of them. Typically, certain technologies are designed for meeting certain demands – such as low power consumption with long range communication or short range communication with minimal energy consumption. Moreover the IoT technologies do not remain the same forever. Instead, they continue to evolve and implement new functions to expand their application fields and properties. The great example how the wireless technology evolves can be the well known Wi-Fi standard – IEEE 802.11 and it's sub 1Ghz low-power version 802.11 ah [4] or a cellular technology 5G, which aims to revolutionize the IoT market [5].

Another technology which has changed a lot is Bluetooth. In 2003 the Bluetooth – at that time version 1.0 – was considered to be dead [6]. However, the wide adoption in the smartphones and the changes introduced in next generations turned away the spectre of death and Bluetooth is nowadays the most popular protocol for building Wireless Personal Area Networks (WPANs). The most recent evolution drawing our attention is the latest generation of Bluetooth. In 2016, the Bluetooth Special Interest Group (SIG) released Bluetooth 5.0, which was in 2019 replaced by version 5.1 [7], together with its first official specification for mesh topology, Bluetooth Mesh [8].

The new Bluetooth generation (5.x) introduced a lot of changes like higher data rate, which is 2.1 Mb/s in the Enhanced Data Rate (EDR). However the most important changes from the IoT point of view focus on the Bluetooth Low Energy (BLE) version. Bluetooth 4.x has a range between 50 and 100 m in the free space and 10–20 m in indoor environments. Bluetooth 5.x aims to quadruple the range of BLE devices. Another very significant enhancement which created the basis for more complicated algorithms and applications in the extended size of advertising packets, which can contain up to 255 bytes, instead of 31 as it was with BLE 4.x.

2.1. Bluetooth Mesh

As Bluetooth is a low-power wireless technology working in a 2.4 GHz ISM band, it is physically impossible to send signals for a long-range without relaying, which limits usage of Bluetooth in applications requiring wide coverage, such as agricultural or environmental monitoring scenarios. BLE was designed to follow the star network topology and there is no way to use relaying according to core specification. Therefore, plenty of research has been done to realize mesh functionality for Bluetooth [9], and products using BLE-based mesh networks (e.g., CSRmesh or Nordic nRF OpenMesh) have appeared in application scenarios like home automation. However the absence of official support on mesh topology have been a serious drawback till the Bluetooth Mesh specification appeared [8].

Generally, mesh networks are made up of devices that are capable of communicating with each other besides their basic functionalities. Each device operating in Bluetooth Mesh network is called "node" – and it works in the same manner as nodes in IEEE 802.15.4 networks. BM network uses typical client-server architecture, where each node plays role of client, server or both. Simplifying, server node usually is related to some end device that contains state (e.g. bulb with level of dimming), while client node is related to controlling device (e.g. light switch).

Nodes in Bluetooth mesh networks communicate with each other by sending messages in an advertising manner. Each message contain the addresses of the sender and receiver nodes (SRC and DST, accordingly), time to live counter used in flooding mechanism (TTL), sequential number of message (SEQ) and initialization vector index (IV INDEX), used to guarantee uniqueness of the message, and the basic content (payload). What is important from the perspective of this paper, message does not contain address of the relying node.

The relaying of messages is handled in a flooding manner, where the range of message (in number of hops) is defined by well known TTL field. Message is not relied further by the node if its TTL field value is equal to one, it has been already relied (identification based on cache of relied messages) or the node's address is equal to DST field (in case of unicast address). Another relevant element of the BM standard from the perspective of traffic analysis is Heartbeat message, sent periodically by each node. The aim of this message is to indicate activity of the node. Based on the content of the Heartbeat message other nodes calculate initial TTL value for the given unicast address. More information about Bluetooth Mesh can be found in a paper [10].

2.2. Bluetooth Mesh security

The first important note regarding security in Bluetooth mesh is that it is mandatory. In Bluetooth Mesh all mesh messages have to be encrypted and authenticated. There are three security levels which are handled independently: network security, application security and device security. Each mechanism is protected by a different security key. The security keys can be changed during the life of the mesh network. Each key addresses a specific concern:

1. **Device Key (DevKey)**: Every device owns a unique DevKey which is only known to the network configuration device and the device itself. Device-specific key is used during the provisioning process for securing communication between the device which is added to the network and the network configuration device.
2. **Network Key (NetKey)**: A Network Key is shared across all nodes in the network. NetKey allows a node to decrypt and authenticate up to the network layer. It means that node can send and relay the messages, but it cannot decrypt application data.
3. **Application Key (AppKey)**: An Application Key is used to decrypt and authenticate messages at the application level and is shared across all nodes which participate in a given mesh application. It's role is to provide confidentiality and integrity at the application level.

Taking into account the designing IDS systems, it is very important which keys might be available to the IDS system in order to detect attacks. Providing the application key to the IDS is reckless as this key can be used to take control of the system. Therefore AppKey availability for IDS can result in creating an additional opportunity to compromise the network. On the other hand, the potential leak of the network key does not pose a critical threat to the system – it is not possible to generate control messages or to preview the transmitted data.

2.3. Types of attacks in Bluetooth Mesh networks

Despite the proposed security mechanisms, BM networks are still vulnerable to attacks. According to the work [11] recovering both device and application keys by attacker is theoretically possible. Recovering device key is especially dangerous as it allows to take full control of the attacked node. The attacks can be divided due to the purpose of the attack on: denial of service attacks, information acquisition attacks, attacks reducing quality of service or weakening IDS attacks. In this section we discuss mostly denial of service attacks which can be potentially carried out in BM network and which can be detected by the IDS system proposed in the paper.

A very common target of the attacker is to prevent proper network operation. This goal can be achieved through i.a. media or devices occupation, deleting, delaying and redirecting messages or destroying devices. In jamming attack attacking device tries to change the transmitted signal or completely jam it by sending a signal causing interference [12]. The following types of jamming can be distinguished [13]: continuous (the attacker occupies the channel without any brake), deceptive (the attacker repeats correct signal), random (the attacker starts broadcasting at random times and transmits for a random period of time), reactive (the attacker listens to the activity in the given channel and starts broadcasting when it registers a signal). The problem from the perspective of the attacker is high energy consumption, due to the necessity of frequent jamming of the channel in order to obtain adequate attack efficiency.

Next way to decrease quality of service is dropping packets [14]. Within this group the most important in case of BM networks is selective forwarding attack. In this attack, a malicious relying node does not forward selected messages, so the message is not propagated further. If the node drops only part of messages, the attack is called a gray hole attack. In the extreme case, when the malicious node does not rely any message, the attack is called a black hole attack. Due to flooding approach for implementing many-to-many communication in BM networks, the effectiveness of attack depends on both number of infected nodes and density of the network. Another way to remove packets from the network before they arrive to the destination node could take advantage on the algorithm for calculating initial TTL in BM networks. Initial TTL value for sending the given message to the destination node with the given unicast address is based on Heartbeat message sent by that node. Thus, if payload of the Heartbeat message created by destination node is malformed, then TTL value required to deliver message to the destination node may be underestimated and rejected by relying nodes before arriving to the target.

2.4. Intrusion Detection Systems in IoT systems

The idea of intrusion detection appeared in the 1980s and is usually connected with Jim Anderson and his report "Computer Threat Monitoring and Surveillance" providing means for improving security by logs files analysis and differentiation of normal use from anomalies. Nowadays, IDS is one of the primary tools used for protection of traditional IP networks and information systems.

The IDS monitors the operations of a host and a network, alerting the system administrator when it detects a security violation.

Despite the maturity of IDS technology, which is very widely adapted in traditional IP networks, current solutions are inadequate for IoT systems, because of IoT particular characteristics that affect IDS development. First of all in IoT networks one can experience very limited processing and storage capacity of network nodes that host IDS agents, specific protocol stacks, multi-hop communication and huge diversity of standards and proprietary solutions applied in particular applications. As a consequence, in case of IoT it is rather difficult to mention commercial products which are the leaders in the market. However there is a plenty of research efforts which appeared in the last ten years. An interesting overview of existing solutions are presented in survey papers written by Zarpelao et al. [15] or Gendreau and Moorman [16].

Zarpelao et al. [15] propose an IDS for IoT classification according to four criteria: intrusion detection methods, placement strategy, security threat and validation strategy. In this paper, we concentrate only on the first two criteria which determines the intrusion detection systems architectures, as security threats have been already described in previous section and validation strategy characterize the way how the IDS was tested, not the way how it operates.

2.4.1. Intrusion detection methods

Intrusion detection techniques can be classified into four categories depending on the detection mechanism used in the system: signature-based, specification-based, anomaly-based and hybrid. In signature-based approaches, Intrusion Detection Systems detect attacks when system behavior matches an attack signature which is stored in the IDS databases. If any system or network activity matches with stored patterns, then an alert will be triggered. Signature-based IDSs are popular as they are easy to understand, effective and accurate at detecting known threats. Their main drawback is ineffectiveness in detecting new attacks and variants of known attacks, because a matching signature for these attacks is still unknown.

Anomaly-based IDSs compare the activities of a system in a particular time to a normal behavior profile, which was prepared earlier based on system observations. Anomaly-based IDS generates the alert whenever a deviation from normal behavior exceeds a threshold. This technique is efficient to detect new attacks, in particular, the attacks related to tempering the IoT nodes. However, anything that does not match to a normal behavior is considered an intrusion and learning the entire scope of the normal behavior is not a simple task. Hence, this approach is characterized by a high false positive rate [17].

Specification is a set of rules and thresholds that define the expected behavior for network components such as nodes, protocols, and routing tables. Specification-based approaches detect intrusions when network behavior deviates from specification definitions. Therefore, specification-based IDS is very similar to anomaly-based IDS. The main difference between these approaches is the way how the rules are build. In anomaly-based IDS the model of normal behaviour is build automatically based on network observation, while in specification-based approaches, a human expert should manually define the rules of each specification. Manually defined specifications usually provide lower false positive rates in comparison with the anomaly-based detection [17,18].

Finally hybrid approaches use all previous concepts to maximize their advantages and minimize the impact of their drawbacks.

2.4.2. IDS placement strategies

In IoT networks, the IDS can be placed in the border router, in one or more dedicated hosts, or in every physical node. All of the mentioned strategies involve some advantages and some drawbacks.

In the centralized IDS placement, the IDS is placed in a centralized component, for example, in the border router. It allows on inspecting all the data which is transmitted between low power nodes and Internet as well as all the request coming from the Internet clients. Hence such type of IDS should be very effective in detecting all intrusion attacks coming from Internet. However, analyzing the traffic that traverses the border router is not enough to detect attacks that involve only low power nodes, for example it is impossible to detect the there is a gray hole attack somewhere in the low power (eg. Bluetooth Mesh) network and some messages are removed or delayed. The examples of centralized intrusion detection can be found in [19,20].

Distributed IDS placement assumes that IDSs are placed in every physical object of the low power network – for example in every Bluetooth controlled lamp. Placing the IDS in every node might decrease the communication overhead associated with network monitoring, but requires more resources (processing, storage, and energy) from the nodes. This might be a problem due to resource limitations of low power nodes. To address this issue some lightweight IDSs where proposed. For example Oh et al. [21] defined a lightweight algorithm to match attack signatures and packet payloads, while Lee et al. [22] proposed an anomaly-based method to detect intrusion detection based on energy consumption.

Hybrid IDS placement mixes ideas of centralized and distributed placement to take advantage of the strong points of both solutions. Hybrid solutions realize the idea of fog computing architecture as the decision whether there is an intrusion attack or not is taken in a distributed manner based on monitoring the neighbors behaviour. Nodes that monitor their neighbors are referred to as watchdogs. Good examples of fog IDSs can be found in [18,23,24]. Amaral et al. proposed a system where selected nodes (watchdogs) in the network host an IDS. Their aim to identify intrusions by eavesdropping the exchanged packets in their neighborhood. The watchdog decides whether a node is compromised according to a set of rules.

The Bluetooth Mesh Intrusion Detection System presented in this paper is an example of anomaly-based IDS working in a fog (hybrid) manner. The final decision is taken collaboratively by a set of nodes in comparison to some fog systems supported by the cloud layer like [25].

3. Anomaly detection in Bluetooth Mesh networks

In this paper, a novel approach for anomaly detection in Bluetooth Mesh networks is proposed. In our approach network traffic is

monitored by at least one passive node (*watchdog*). The watchdog node records every packet that it receives only, it does not create any new message nor it relies the received packets. In this work, it is assumed that during the configuration process each watchdog node obtains network key, which allows to decode following fields of each packet: SRC, DST, TTL, SEQ number and IV INDEX. Additionally, watchdog node can calculate payload length (LEN) and received signal strength indicator (RSSI). Moreover each watchdog is also capable to verify if two received packets represent the same message by comparing SRC and SEQ fields values. In order to limit an access to sensitive data the watchdog node does not store application key, thus decryption of payload data is not possible.

3.1. Learning phase

Watchdog node observes network traffic continuously and processes each received packet p_j. Observation window $T_i = [i \cdot t_{win}, (i+1) \cdot t_{win}]$ of length t_{win} is introduced to divide all registered packets into disjunctive sets $X_i = \{p_j | t(p_j) \in T_i\}$, where $t(p_j)$ is time of receiving packet p_j, $i \in \{1, \cdots, N_L\}$ and N_L is number of observation windows in the learning phase. For each set X_i values of following features are calculated: n^i — size of the set, μ^i_{SRC} — average number of packets per source address, σ^i_{SRC} — standard deviation of number packets per source number, μ^i_{DST} — average number of packets per destination address, σ^i_{DST} — standard deviation of number packets per destination number, μ^i_{TTL} — average value of TTL values, σ^i_{TTL} — standard deviation of TTL values, μ^i_{RSSI} — average value of RSSI values, σ^i_{RSSI} — standard deviation of RSSI values, μ^i_{LEN} — average size of packet, σ^i_{LEN} — standard deviation of size of packets, μ^i_p — average message processing time by neighbours of watchdog node[1], σ^i_P — standard deviation of message processing time and μ^i_{SEQ} — average value change of SEQ field of packets with the same value of SRC field.

Features values for set X_i create tuple o_i:

$$o_i = (n^i, \mu^i_{SRC}, \sigma^i_{SRC}, \mu^i_{DST}, \sigma^i_{DST}, \mu^i_{TTL}, \sigma^i_{TTL}, \mu^i_{RSSI}, \sigma^i_{RSSI}, \mu^i_{LEN}, \sigma^i_{LEN}, \mu^i_P, \sigma^i_P, \mu^i_{SEQ}), \tag{1}$$

that describes traffic in observation window T_i. As traffic characteristic may vary in time (e.g. due to time of day) network traffic needs to be observed for significant time, i.e. N observation windows. Hence, as the result of learning phase N tuples representing correct traffic characteristic are created. Additionally, to simplify further consideration, each feature is normalized according to min-max normalization formula:

$$\alpha_{norm} = \frac{\alpha - \alpha_{min}}{\alpha_{max} - \alpha_{min}}, \tag{2}$$

where α is a value of feature before normalization, α_{norm} is a value after normalization, $\alpha_{\max}$ and $\alpha_{\min}$ are the greatest and smallest values of normalized feature among all N tuples in set O, respectively.

To detect anomaly in network traffic some general model of correct traffic needs to be created first. In our approach clusterization is used to divide set $O = \{o_1, o_2, \cdots, o_N\}$ into K disjunctive subsets (clusters) C_i, $i = 1, 2, \cdots, K$, representing similar traffic characteristic (e.g. if traffic characteristic depends on time of day each cluster represents traffic in different time of day). In this work **mclust** R package [26] for model-based clustering based on finite Gaussian mixture modelling was used to perform clusteriztion of set O. Thus, the number of clusters (mixture components) is calculated and does not need to be given a priori. If the traffic characteristic does not vary significantly in time only one cluster should be detected. Each cluster C_j is described by center θ_j and radius r_j:

$$\theta_j = (\overline{n^j}, \overline{\mu^j_{SRC}}, \overline{\sigma^j_{SRC}}, \overline{\mu^j_{DST}}, \overline{\sigma^j_{DST}}, \overline{\mu^j_{TTL}}, \overline{\sigma^j_{TTL}}, \overline{\mu^j_{RSSI}}, \overline{\sigma^j_{RSSI}}, \overline{\mu^j_{LEN}}, \overline{\sigma^j_{LEN}}, \overline{\mu^j_P}, \overline{\sigma^j_P}, \overline{\mu^j_{SEQ}}), \quad \overline{x^j} = \frac{\sum_{o_i \in C_j} x^i}{|C_j|}, \tag{3}$$

$$r_j = \max_{o_i \in C_j} d(o_i, \theta_j), \tag{4}$$

where d is Euclidean distance between two tuples.

The above considerations apply to single watchdog node. Of course multiple watchdog nodes should learn network traffic independently — if location of nodes differs even slightly the observed traffic may differ significantly.

3.2. Anomaly detection by a single watchdog

When the model of traffic (in properly working network) is created, anomalies that can indicate attack occurrence, can be detected. To detect anomalies watchdog node observes traffic constantly, just like in the learning phase. After each t_{win} period new set X_i is created and tuple o_i is calculated[2] Then, for each cluster C_j distance $d_{ij} = d(\theta_j, o_i)$ between its center θ_j and tuple o_i is calculated. If condition:

[1] Processing time is estimated with the following approach: all packets in set X_i are divided into those with the same value of field SRC and SEQ (packets representing the same message, repeated by various nodes). Packets in subsets differ only with TTL field value. Then, average receiving time t_{TTL} is calculated for all packets with the same value *ttl* of TTL field in those subsets. It is assumed that message processing time is average value of values $t_{ttl+1} - t_{ttl}$, for all observed values of TTL, but the biggest. Hence, the average message processing time is average value of all estimated message processing times.

[2] To normalize each feature values greatest and smallest values of feature ($\alpha_{\min}$ and $\alpha_{\max}$) obtained in learning phase are used.

$$\exists_{C_j} \; d_{ij} > r_j \cdot \gamma_j, \; \gamma_j > 1, \tag{5}$$

where γ_j is coefficient that relaxes requirement for similarity of analyzed traffic to the one observed in learning phase, the anomaly is detected by watchdog. The role of coefficient is to prevent false positive detection in case of traffic slightly different than one observed in learning phase. Adjusting value of coefficient γ_j is non trivial task and is not subject of this research. In this work, value of coefficient is calculated with following formula: $\gamma_j = 1 + \frac{\sigma_d}{r_j}$, where σ_d is standard deviation of distances between all tuples in the cluster C_j and center θ_j of this cluster.

4. Cooperative anomaly detection by multiple watchdogs

4.1. Local and global impact of attack on Bluetooth Mesh network

When one of the network node is attacked the change in the traffic characteristic may have mostly local character — e.g. in case of delay attack only neighbouring nodes (i.e. within transmission range) can detect change of processing time of packets. As well other attacks, e.g. gray hole attack, will cause the biggest change in traffic characteristic in the neighbourhood of the infected node. Hence, introducing multiple watchdog can increase quality of anomaly detection, especially in case of networks with big number of nodes.

Introducing multiple watchdogs requires designing an algorithm for cooperative decision making. In literature multiple cooperative approaches were proposed. In this work, the following technique is proposed – alarm is triggered if at least one watchdog detects anomaly in network traffic. Lets assume that in the network set S composed of M watchdogs w_i, $i = 1, \cdots, M$ is deployed. Watchdog w_i is placed in location $l_i = [x_i, y_i]$. Each watchdog monitors network traffic independently, according to the technique described in the previous section. At the beginning, each watchdog constructs own model of typical network traffic. Next, in every observation window T_i each watchdog w_j assesses, according to the proposed condition (5), if there is any anomaly in observed network traffic. If at least one of the watchdogs detects anomaly alert is raised. The following anomaly detection indicator ψ by the set of watchdogs S in the observation window T_i can be introduced:

$$\psi^i(S) = \max_{j=1,\cdots,M} \min_{C_k \in C(w_j)} \frac{d(\theta_k, o_i^k)}{r_k \cdot \gamma_k}, \tag{6}$$

where $C(w_j)$ is set of clusters created for watchdog w_j and tuple o_i^k is calculated by watchdog w_j in observation window T_i. If $\psi^i(S) > 1$ then anomaly is detected by set of watchdogs S.

The presented aggregation model of watchdogs votes arise from the fact that the change in network behaviour is in many cases observable only in a small neighbourhood of the infected node. Such approach will cause that the attack detection scheme has tendency to be sensitive (potential high true positive rate), but not specific (low true negative rate, as observation of one watchdog does not need to be confirmed by another watchdog).

4.2. Optimal placement of multiple watchdogs

It is clear that the more watchdogs is installed the better accuracy of intrusion detection can be obtained. However each watchdog is connected with particular cost. For example in our laboratory the light control network is composed of Nordic boards, which are not sufficient to realize the IDS functions. Hence the fog computing solution is applied. In order to realize intrusion detection the more powerful Raspberry Pi devices are added and connected to Nordic nodes. Raspberry Pi nodes are responsible for watchdog implementation based on data received from Nordic nodes. Each Raspberry Pi detects attacks separately close to the edge and individual decisions are sent as an extra packets to the border router.

As the number of watchdog nodes is limited, the question where to locate those nodes to ensure the highest efficiency of the IDS arises. Intuitively, watchdogs should be placed far from each other to monitor various parts of the network. However, deciding if watchdog nodes should be placed in dense rather than sparse regions of network is more difficult — on the one hand in dense regions more nodes are monitored directly, on the other in sparse regions detecting anomalies is easier due to more significant change in traffic characteristic in case of attack on a single node.

To deploy set of watchdog nodes S in the network the M locations need to be determined. Placing watchdogs in those locations should guarantee detection of each anomaly in the network traffic, but at the same time anomalies should not be detected when network works properly. Lets consider BM network composed of R nodes r_i, $i \in \{1, \cdots, R\}$ deployed in two dimensional area A. Each node r_i is at risk of being infected with probability a_i (if node cannot be infected then $a_i = 0$). Infected node misbehave, e.g. does not rely some packets in case of gray hole attack. To simplify considerations, let assume that network operates for N observation windows, out of which network operates at the beginning properly for $N_P > N_L$ observation windows, so learning phase can be completed. To make task of detecting anomalies more difficult, let assume, that only one node can be infected and misbehave at one observation window T_i.

To evaluate given set of watchdogs S first learning phase needs to be performed. Later, during the rest $N - N_L$ observation windows watchdogs observe traffic and detect anomalies. Each observation window T_i is associated with label b_i that indicates if any node is infected in this window (if anomaly should be detected). Label b_i takes value 1 if at least one node is infected and -1 otherwise. Let define following criterion Ψ_S for set S evaluation:

$$\Psi(S) = \sum_{i=N_L+1}^{N} \psi^i(S) \cdot b_i. \tag{7}$$

The proposed criterion $\Psi(S)$ has one drawback, which is very wide range of possible values of element ψ_S^i. This issue may cause uneven impact of detection quality in individual time windows on the final evaluation. For example if one set of watchdogs performs very well in detecting anomalies in few windows (large value of ψ_S^i) and in all others does not detect anomaly at all, than, according to the proposed criterion, it may outperform the set of watchdogs that detects anomalies correctly in all observation windows, but not so strongly (value ψ_S^i passes condition (5), but it is not so clearly bigger). Thus, in case of comparing quality of multiple set of watchdog locations, normalization of value $\psi^i(S)$ among all sets is introduced.

Let assume that T sets of watchdogs S_i creates set $\mathcal{S} = \{S_1, S_2, \cdots, S_T\}$ and these sets should be compared with each other using criterion Ψ to recommend the most efficient one. Let define values $\psi_{\max}^i$ and $\psi_{\min}^i$ — respectively maximal and minimal value of criterion $\psi^i(S)$ among all sets $S \in \mathcal{S}$:

$$\psi_{min}^i = \min_{S_j \in \mathcal{S}} \psi^i(S_j) \tag{8}$$

$$\psi_{max}^i = \max_{S_j \in \mathcal{S}} \psi^i(S_j) \tag{9}$$

Then the normalized version of criterion Ψ is:

$$\hat{\Psi}(S) = \sum_{i=N_L+1}^{N} \frac{\psi^i(S) - \psi_{min}^i}{\psi_{max}^i - \psi_{min}^i} \cdot b_i = \sum_{i=N_L+1}^{N} \hat{\psi}^i(S) \cdot b_i \tag{10}$$

The most straightforward way to find set S with high value of criterion $\hat{\Psi}$ in the existing, functioning network is performing a series of experiments. Each experiment should involve deploying watchdog nodes in some promising locations, learning characteristics of correct traffic, performing exemplary attacks and verifying quality of anomaly detection. If the results are unsatisfactory another set of locations needs to be chosen and experiment repeated to check if better results can be obtained. In real network such approach is very time consuming and in practice allow to test only few sets of watchdogs locations. Additionally, performing experiments that involve manipulating nodes in existing network is usually impossible.

Hence, the best option to determine the optimal placement of watchdogs is to apply simulation methods of determining recommended set of watchdogs locations. However, as Bluetooth Mesh is novel technology, best to our knowledge, no simulator of BM is publicly available yet. Therefore, we decided to create a simulation software for Bluetooth Mesh networks which can support engineers in deploying watchdog based IDS system.

5. Simulation of watchdogs' placement

The simulator *BMWatchSim* (Bluetooth Mesh Watchdog Simulator) was created to support determining the positions of the watchdog nodes. Therefore, only selected elements of the BM protocol relevant for this task were implemented. The *BMWatchSim* allows to define any number of nodes of client and server type that create the lighting control system. The simulator implements, in accordance with the BM specification, the network flooding protocol, that allows many to many communication as well as the network management protocol, the effect of which is broadcasting Heartbeat messages by nodes. In the context of the research on the security of the BM network, an important element of the simulator is the possibility to configure any number of attacks on any network nodes that start and end at any moment of the simulation. This feature allows verification of the effectiveness of IDS methods in numerous, also complex, attack scenarios.

Particular attention was paid to two another aspects of simulation, that have a significant impact in the context of research on the detection of network traffic anomalies. The first one is randomness in the functioning of the nodes: packets are sent in random intervals and packets processing takes random amount of time. Additionally, all random variables have normal distribution with configurable parameters. The second important aspect is simulation of received signal strength indicator value, which, according to the conducted research and the literature, shows significant variability despite the constant distance between the transmitting and the receiving device. In order to improve the quality of simulation in this area, experiments with two real BM devices were conducted. The received signal strength indicator was measured for multiple sent packets (30 seconds of transmission) for every possible distance (with step equal to 10 cm). This way for each considered distance a list of the RSSI values was obtained. This list is used by the simulator to calculate the RSSI value for each received packet — the value is drawn with uniform distribution from the list for the appropriate distance between the transmitter and the receiver. Taking into account the fluctuation of the RSSI value is particularly important in the simulation, as the RSSI is one of the features used in the anomaly detection scheme.

The *BMWatchSim* simulator can be used to determine efficient placement of watchdogs in the existing network with the following assumptions. First, locations and roles (client or server) of all network nodes must be known, as they have crucial influence on observed traffic. Additionally, transmission range r_t and tables of RSSI for all possible distances are available. The last requirement can be easily fulfilled having two devices of the type the network is composed of (with the same radio board and configuration of transmission signal strength). Last element of simulator configuration is the vector of normal distribution parameters for simulating packet processing intervals and intervals between sending two packets. Those parameters can be estimated based on real traffic, that can be sniffed e.g. with the watchdog node.

The last element of the simulation configuration is attacks definition. The first attack starts in $N_P \cdot t_{win}$ time of simulation. Next attacks can overlap with previous or start one by one. The result of each attack is disturbance of the single network node what should be reflected in the network traffic.

The result of performing simulation is flow (list of all registered packets with timestamp) in every potential location of watchdog[3] Such log is an input for the algorithm choosing set of the watchdogs locations. First, for each potential location learning phase is performed. The result is a characteristic of correct traffic (clusters described with center θ_j, radius r_j and relaxation coefficient γ_j) for each considered location. Then, T random sets (M potential locations each) are created. Then, for each set of potential locations S_j value $\psi^j(S)$ in each observation window T_i is calculated. Finally, normalized version of criterion $\hat{\Psi}(S)$ is computed — set with the highest value of criterion contains recommended locations of watchdogs to apply in real network.

6. Experimental verification in laboratory

To verify the proposed technique for optimal watchdogs placement the following approach was applied. First, the network, which should be monitored, was recognized. The important information about network was quantity, type and location of nodes, physical devices used to implement network (to determine transmission range and RSSI characteristic) and data which nodes were at risk of being infected (with probability). This information allowed us to configure simulation. For each node at risk of being infected an attack was added to configuration. The length of the attack depended on probability of the node infection in real network. Based on the simulation results recommended sets of watchdog locations were chosen. Then, the watchdogs were deployed in the chosen locations in the real network to verify quality of recommended set of locations.

The laboratory involved in the research is composed of 15 Nordic development kit boards (3 DK NRF52832 and 12 DK NRF52840) and 3 Raspberry Pi boards. To create watchdog node, which was able to sniff traffic in monitored network and had enough resources to perform traffic analysis in real time, the Raspberry Pi devices were connected with the DK boards. The laboratory network during one of the experiments is presented in Fig. 1.

6.1. Real-life use case

Increased coverage of network thanks to the introduction of many-to-many communication allows to extend the current area of applications of Bluetooth technology. As pointed in article [10] the most promising areas for BM technology are smart homes/offices and industrial control. Within smart home area one of the first applications of BM, available already on market, are systems for smart lighting [27]. Thus, in the research presented in this article the network for lighting control is investigated as an exemplary network endangered of being attacked.

To implement system the generic model level from standard Bluetooth Mesh models [8] was used. This model allows to set and read level of the given feature (in this case dimming) of the server node. To set level state one of three following messages can be used:

- Generic Level Set — changes the level state to an absolute value,
- Generic Delta Set — changes the level state by a relative value,
- Generic Move Set — starts changing the level state at a given speed (in positive or negative direction). The speed is calculated based on two fields included in message: delta value and transition time.

Additionally, each server node periodically sends Generic Level Status message with current value of the level state.

Due to the fact, that watchdog node cannot decrypt messages' payload the main difference between messages from the perspective of this work is their size. A seemingly insignificant value is an important feature of network traffic, since messages with malformed payload (e.g. code injection) usually differ in size or filtering one type of messages in network (i.e. gray hole attack) results in change of observed distribution of message size.

6.2. The test network topology and attacks scenario

The following verification scenario was proposed. The 12 out of 15 DK devices were used to create BM network. The remaining devices acted together with Raspberry Pi as watchdog nodes. The network implemented the lighting control system with single client node and 11 server nodes. The topology of the network is presented in Fig. 2. The network was deployed in obstacle-less environment. In the considered scenario it was assumed that only edge nodes were at risk of being attacked with the same probability. The client node was responsible for controlling dimming level of each server node (each server node can be identified with single bulb).

The major issue in the research on detecting attacks in networks implemented in novel technology is lack of documented exemplary attacks. Hence, for the sake of this research the following version of gray hole attack was implemented, both in simulator and real network. As the result of the gray hole attack the infected node stops broadcasting Generic Level Status message and relying both Generic Level Set and Generic Level Status messages.

[3] for a set of potential locations of the watchdog node, all locations (with some density) within the range r_t of at least one network node are accepted.

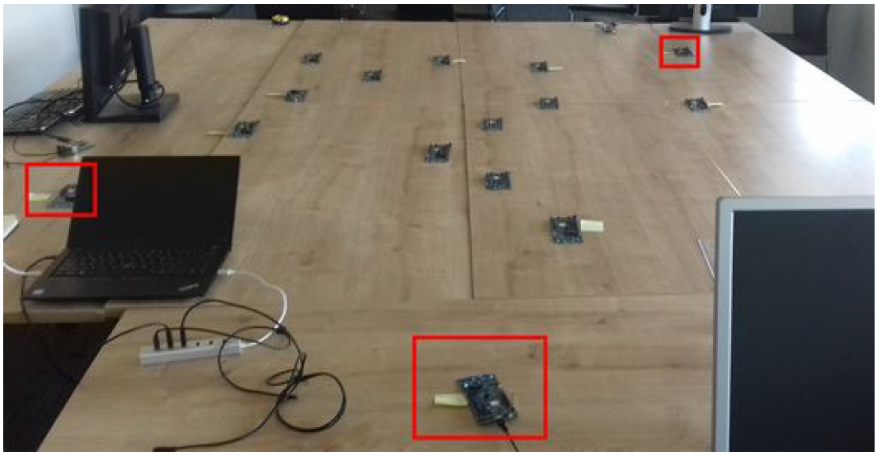

Fig. 1. Bluetooth Mesh network in laboratory. Watchdog nodes marked with red frames.

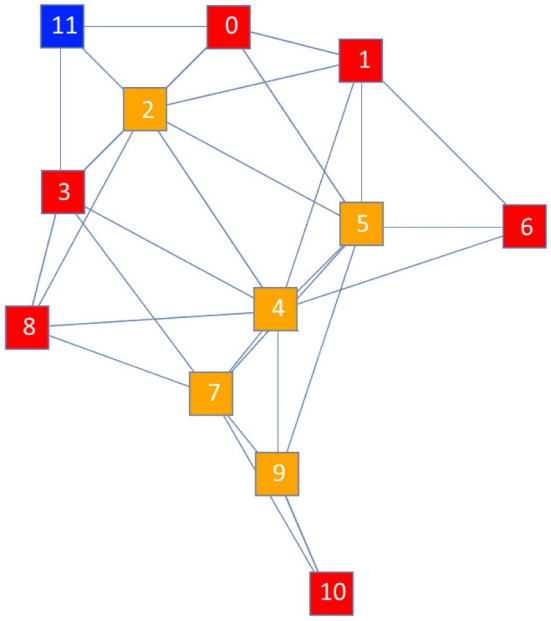

Fig. 2. Network topology in verification scenario. With blue color client node is marked, with orange and red server nodes are marked. The red color indicates nodes that are subject of the attack.

Each experiment was composed of $N = 45$ observation windows, each of length $t_{win} = 2min$. The network operated correctly for $N_P = 15$ observation windows, then 6 nodes (located on the edge of the network) were infected one by one, each for 5 observation windows. In each experiment location of watchdogs is different. Each watchdog records every packet that it receives during given observation window and calculates traffic features in that window on this basis.

6.3. Simulation results

First, based on the simulation results, influence of the single watchdog location on criterion $\hat{\Psi}$ was investigated. The criterion value was calculated for a set $\mathcal{S}$, composed of single element sets S_i, one for each potential location of watchdog in the considered network. The obtained values of the criterion are presented in Fig. 3. The lighter color denotes the higher value of criterion. The highest quality of anomaly detection should be obtained if watchdog is placed in location where only infected nodes are in range. The dark area in top left corner depicts that in the proposed scenario traffic generated by client node does not contain much information about the anomalies appearance.

However, the results for single watchdog placement gives only some intuition how to deploy few watchdogs to get the best anomaly detection quality. To determine efficient location for M watchdogs $T = 1000$ random sets of M locations were generated and compared with criterion $\hat{\Psi}$. The highest value was obtained for locations marked with red color (S_{best}) in Fig. 3. As expected, each of

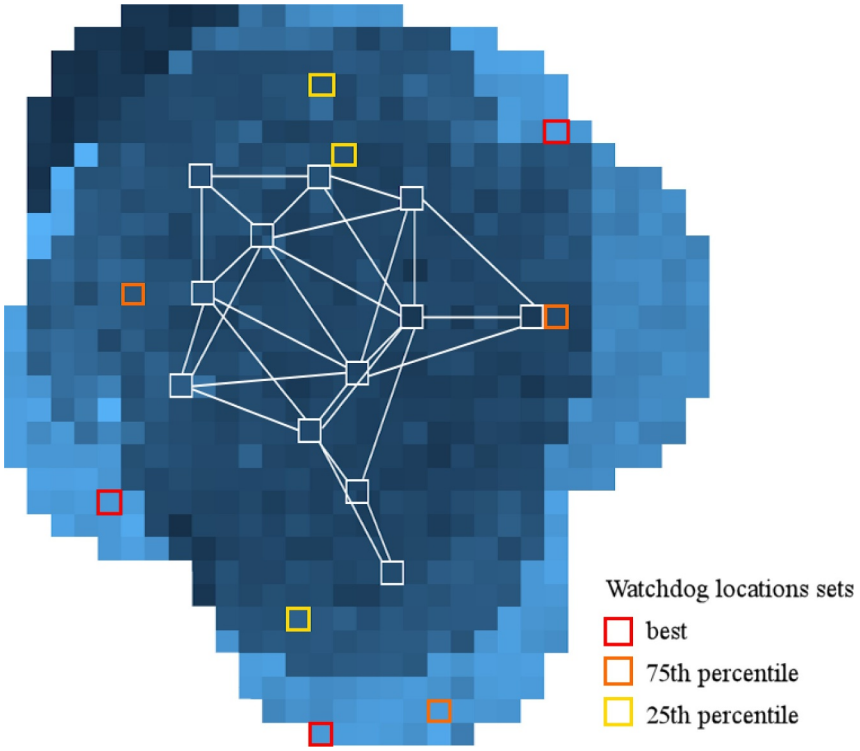

Fig. 3. Value of criterion $\hat{\Psi}$ for all potential locations of watchdog in case of single watchdog deployment. The lighter color denotes the higher value of criterion. The best out of considered sets and sets 75th and 25th percentile are marked with colors red, orange and yellow colors, appropriately.

these locations is placed in lighter area (with higher value of $\hat{\Psi}$ criterion). Locations are distant from each other, thus each watchdog is responsible for detecting attack on another part of the network. In the Fig. 3 sets of locations that are on 25th (S_{25}) and 75th (S_{75}) percentile are presented too. Those sets were used in further research in laboratory as well.

6.4. Verification in laboratory

To verify quality of anomaly detection by set of watchdogs deployed in locations S_{best}, S_{75} and S_{25} the experiments in laboratory network were conducted. The value of the normalized anomaly indicator $\hat{\psi}^i(S)$ obtained by the watchdogs deployed in the three considered scenarios are presented in the Table 1. The very positive result is that for all watchdog sets all attacks were detected – windows from 6 to 35 are assessed correctly. However the watchdogs deployed in locations from sets S_{75} and S_{25} had a tendency to be too sensitive. For the locations in set S_{75} one observation window was misjudged ($TNR = 0.8$/specificity equal to 80%), for the set S_{25} even two false positive assessments were done ($TNR = 0.6$). Hence the proposed solutions differ not only in the values of normalized anomaly detection indicator but also in the number of correctly classified examples. Confusion matrixes illustrating the classification accuracy (for locations S_{best}, S_{75} and S_{25}) in laboratory are presented in Table 2.

The aggregated values of anomaly detection quality measured with criterion $\hat{\Psi}$ are presented in Table 3, which compares value of $\hat{\Psi}$ for all three sets obtained both in the simulation and in the laboratory. As expected watchdogs deployed in locations from S_{75} perform worse than recommended solution (S_{best}) and slightly better that watchdogs deployed in locations from S_{25}.

The presented results show that simulation can be successfully used to obtain deployment locations for set of watchdogs that ensure high performance according to some given criterion. However, obviously, results obtained with simulation are flawed. This is especially true in case of simulating wireless communication due to low level of measurement accuracy caused by high variability of the RSSI value [28,29].

The inaccuracy of simulation is visible also in the presented results. The difference between value of criterium $\hat{\Psi}$ obtained for sets S_{75} and S_{25} in simulation experiment is much bigger than the one obtained in laboratory (27% of $\hat{\Psi}$ in simulation versus 6% of $\hat{\Psi}$ in laboratory). Inaccuracy of simulation in the presented research is mainly caused by the lack of uniformity of radio transmission. In Fig. 4 the relations between a distance and RSSI readings are compared for the few exemplary mutual arrangements of sender and receiver. It can be observed, that mutual arrangement has significant influence on the transmission range and variability of RSSI. Quantity of possible mutual is relatively large, thus accurate modelling of those aspects is a difficult task. Additionally, in most real networks mutual arrangement of devices is unknown (e.g. devices are in casing, that prevent identifying orientation of internal components), so even the most accurate modelling of transmission range would be useless.

Table 1
The values of anomaly indicator $\hat{\psi}^i(S)$ for three watchdogs' placement sets: S_{best}, S_{75} and S_{25}. For each window the values are annotated by the flag if the infection was detected or not.

Infected node	———					8				
T_i	1	2	3	4	5	6	7	8	9	10
$\hat{\psi}^i(S_{best})$	0.77	0.49	0.84	0.54	0.53	1.00	1.00	1.00	1.00	1.00
infection detected?	N	N	N	N	N	Y	Y	Y	Y	Y
$\hat{\psi}^i(S_{75})$	0.68	1.00	1.00	0.88	0.56	0.96	0.84	0.82	0.77	0.63
infection detected?	N	**Y**	N	N	N	Y	Y	Y	Y	Y
$\hat{\psi}^i(S_{25})$	1.00	0.39	0.80	1.00	1.00	0.41	0.57	0.42	0.44	0.34
infection detected?	N	N	N	**Y**	**Y**	Y	Y	Y	Y	Y
Infected node	0					10				
T_i	11	12	13	14	15	16	17	18	19	20
$\hat{\psi}^i(S_{best})$	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
infection detected?	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
$\hat{\psi}^i(S_{75})$	0.55	0.42	0.48	0.39	0.39	0.38	0.56	0.49	0.58	0.63
infection detected?	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
$\hat{\psi}^i(S_{25})$	0.39	0.59	0.70	0.60	0.66	0.76	0.68	0.82	0.81	0.92
infection detected?	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Infected node	3					1				
T_i	21	22	23	24	25	26	27	28	29	30
$\hat{\psi}^i(S_{best})$	0.77	0.83	0.54	0.63	0.90	1.00	0.74	1.00	1.00	0.89
infection detected?	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
$\hat{\psi}^i(S_{75})$	0.61	0.77	0.52	1.00	0.50	0.87	1.00	0.88	1.00	1.00
infection detected?	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
$\hat{\psi}^i(S_{25})$	1.00	1.00	1.00	1.00	1.00	0.63	0.61	0.83	0.71	0.64
infection detected?	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Infected node	6									
T_i	31	32	33	34	35					
$\hat{\psi}^i(S_{best})$	1.00	0.80	0.88	0.76	0.92					
infection detected?	Y	Y	Y	Y	Y					
$\hat{\psi}^i(S_{75})$	0.99	1.00	1.00	1.00	1.00					
infection detected?	Y	Y	Y	Y	Y					
$\hat{\psi}^i(S_{25})$	0.64	0.66	0.67	0.53	0.70					
infection detected?	Y	Y	Y	Y	Y					

Table 2
The confusion matrixes for sets S_{best}, S_{75} and S_{25} .

S_{best}		Attack detected		S_{75}		Attack detected		S_{25}		Attack detected	
		Y	N			Y	N			Y	N
Attack	Y	30	0	Attack	Y	30	0	Attack	Y	30	0
Conducted	N	0	5	Conducted	N	1	4	Conducted	N	2	3
Classification accuracy: 100%				Classification accuracy: 97%				Classification accuracy: 94%			

The inaccuracy of the transmission range modeling may cause that in real network watchdog observes traffic generated by other set of nodes than results from simulation. This problem is responsible for the fact, that results obtained for set S_{best} in laboratory is not that much better than for sets S_{25} and S_{75} as it was expected to be according to simulation. More detailed analysis of the traffic registered by watchdogs located in set S_{best} shows that those watchdogs was registering traffic from more nodes than they should according to the simulation.

Table 3
The comparison of values of criterion $\hat{\Psi}$ for sets S_{best}, S_{75} and S_{25} in simulation and laboratory.

	Simulation		Laboratory	
	$\hat{\Psi}$	% of $\hat{\Psi}(S_{best})$	$\hat{\Psi}$	% of $\hat{\Psi}(S_{best})$
S_{best}	18.19	100	24.50	100
S_{75}	10.02	55	17.91	73
S_{25}	5.16	28	16.51	67

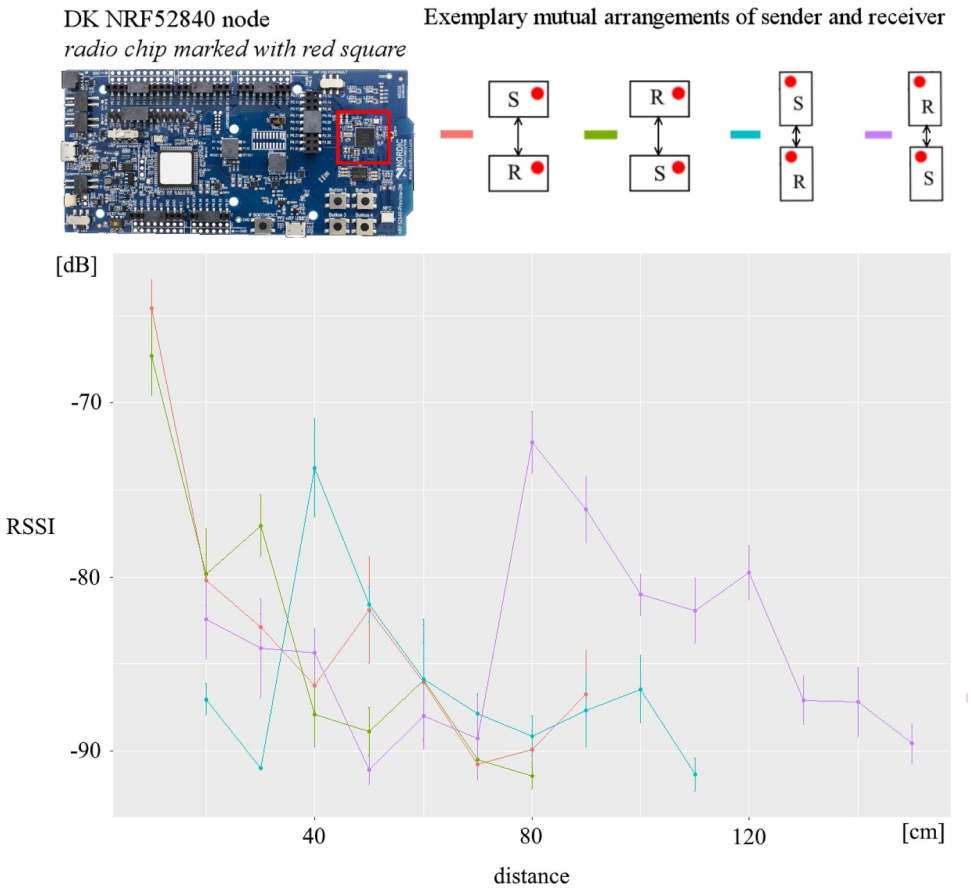

Fig. 4. Mean and standard deviation of RSSI for exemplary mutual arrangement of sender (S) and receiver (R) depending on distance between them. Four exemplary mutual arrangement were compared. The location of the radio chip on the device is marked with the red dot.

7. Summary and conclusions

In this work novel IDS for detecting unknown attacks in BM networks by one or multiple watchdogs was introduced. Additionally, the method for optimal placement of those watchdogs within the network and new BM simulator, that enables quick optimal watchdogs positions determination for the network of the given topology. The presented results show that simulation can be successfully used to obtain deployment locations for set of watchdogs that ensure high performance of anomaly detection in Bluetooth Mesh networks — for the proposed scenario all observation windows were assessed correctly if watchdogs were optimally placed in accordance to the results of the simulation.

Thanks to simulation calculating and comparing numerous possible configurations is very efficient. The simulation process allows on testing one thousand of possible solutions in a dozen of seconds. The verification process in laboratory network was very time consuming (2 days for few sets of watchdog locations) and required the changes in watchdog placement, which were possible in laboratory environment but can be complicated to apply in production environments. Hence the possibility of modelling the placement of our Bluetooth Mesh IDS nodes in a simulator is a key factor which allows on effective deployments in real life networks. The quality of cooperative anomaly detection obtained by the proposed BM IDS system was more than satisfactory.

The encountered problem with modelling transmission range and variability of RSSI can be even greater in case of networks

deployed in area with multiple obstacles (walls, furniture etc.). Due to the reflection and absorption of radio waves, modelling in simulator is much more difficult and requires additional input data to perform useful simulation. Thus, in the future studies we propose to change definition of watchdog location to the set of neighbouring nodes. Then the watchdog should be deployed in such place, that these and only these neighbouring nodes are directly connected to it. Such approach would complicate watchdog deployment process. In addition, due to aforementioned problems with modelling of radio transmission, the location calculated by simulator may not exist in real network. Thus, simulator should additionally suggest which potential relaxations of the set of neighbouring node (i.e. which network node should be added or removed) will have the smallest influence on the quality of the given solution.

References

[1] Leading the IoT: Gartner Insights on How to Lead in a Connected World, in: M. Hung (Ed.), Gartner, 2017.
[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: a survey on enabling technologies, protocols, and applications, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2347–2376, https://doi.org/10.1109/COMST.2015.2444095.
[3] R. Mahmoud, T. Yousuf, F. Aloul, I. Zualkernan, Internet of things (iot) security: current status, challenges and prospective measures, 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST), (2015), pp. 336–341, https://doi.org/10.1109/ICITST.2015.7412116.
[4] Y. Zhou, H. Wang, S. Zheng, Z.Z. Lei, Advances in ieee 802.11ah standardization for machine-type communications in sub-1ghz wlan, 2013 IEEE International Conference on Communications Workshops (ICC), (2013), pp. 1269–1273, https://doi.org/10.1109/ICCW.2013.6649432.
[5] S. Li, L.D. Xu, S. Zhao, 5g internet of things: A survey, J. Ind. Inf. Integrat. 10 (2018) 1–9, https://doi.org/10.1016/j.jii.2018.01.005. URL http://www.sciencedirect.com/science/article/pii/S2452414X18300037
[6] C. Mathias, Bluetooth is dead, Avaliable online: http://www.eetimes.com/(2003).
[7] Bluetooth SIG, Bluetooth 5.1 core specification, Avaliable online: https://www.bluetooth.com/specifications/bluetooth-core-specification/(2019).
[8] Bluetooth Mesh Working Group, Mesh profile specification v 1.0.1, Avaliable online: https://www.bluetooth.com/specifications/mesh-specifications/(2019).
[9] S. Darroudi, C. Gomez, Bluetooth low energy mesh networks: a survey, Sensors 17 (2017), https://doi.org/10.3390/s17071467.
[10] J. Yin, Z. Yang, H. Cao, T. Liu, Z. Zhou, C. Wu, A survey on bluetooth 5.0 and mesh: new milestones of IoT, ACM Trans. Sen. Netw. 15 (3) (2019) 28:1–28:29, https://doi.org/10.1145/3317687.
[11] A. Adomnicai, J.J. Fournier, L. Masson, Hardware security threats against bluetooth mesh networks, 2018 IEEE Conference on Communications and Network Security (CNS), IEEE, 2018, pp. 1–9.
[12] O. El Mouaatamid, M. Lahmer, M. Belkasmi, Internet of things security: layered classification of attacks and possible countermeasures, Electron. J. Inf. Technol. (9) (2016).
[13] W. Xu, W. Trappe, Y. Zhang, T. Wood, The feasibility of launching and detecting jamming attacks in wireless networks, Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, ACM, 2005, pp. 46–57.
[14] L.K. Bysani, A.K. Turuk, A survey on selective forwarding attack in wireless sensor networks, 2011 International Conference on Devices and Communications (ICDeCom), IEEE, 2011, pp. 1–5.
[15] B.B. Zarpelão, R.S. Miani, C.T. Kawakani, S.C. de Alvarenga, A survey of intrusion detection in internet of things, J. Netw. Comput. Appl. 84 (2017) 25–37, https://doi.org/10.1016/j.jnca.2017.02.009.
[16] A.A. Gendreau, M. Moorman, Survey of intrusion detection systems towards an end to end secure internet of things, 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud), (2016), pp. 84–90, https://doi.org/10.1109/FiCloud.2016.20.
[17] R. Mitchell, I.-R. Chen, A survey of intrusion detection techniques for cyber-physical systems, ACM Comput. Surv. 46 (4) (2014) 55:1–55:29, https://doi.org/10.1145/2542049.
[18] J.P. Amaral, L.M. Oliveira, J.J.P.C. Rodrigues, G. Han, L. Shu, Policy and network-based intrusion detection system for ipv6-enabled wireless sensor networks, 2014 IEEE International Conference on Communications (ICC), (2014), pp. 1796–1801, https://doi.org/10.1109/ICC.2014.6883583.
[19] P. Kasinathan, C. Pastrone, M.A. Spirito, M. Vinkovits, Denial-of-service detection in 6lowpan based internet of things, 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, 2013, pp. 600–607.
[20] G. Thamilarasu, S. Chawla, Towards deep-learning-driven intrusion detection for the internet of things, Sensors 19 (9) (2019), https://doi.org/10.3390/s19091977.
[21] D. Oh, D. Kim, W.W. Ro, A malicious pattern detection engine for embedded security systems in the internet of things, Sensors 14 (12) (2014) 24188–24211, https://doi.org/10.3390/s141224188.
[22] T.-H. Lee, C.-H. Wen, L.-H. Chang, H.-S. Chiang, M.-C. Hsieh, A lightweight intrusion detection scheme based on energy consumption analysis in 6lowpan, in: Y.-M. Huang, H.-C. Chao, D.-J. Deng, J.J.J.H. Park (Eds.), Advanced Technologies, Embedded and Multimedia for Human-Centric Computing, Springer Netherlands, Dordrecht, 2014, pp. 1205–1213.
[23] C. Cervantes, D. Poplade, M. Nogueira, A. Santos, Detection of sinkhole attacks for supporting secure routing on 6lowpan for internet of things, 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), (2015), pp. 606–611, https://doi.org/10.1109/INM.2015.7140344.
[24] B. Sudqi Khater, A.W.B. Abdul Wahab, M.Y.I.B. Idris, M. Abdulla Hussain, A. Ahmed Ibrahim, A lightweight perceptron-based intrusion detection system for fog computing, Appl. Sci. 9 (1) (2019), https://doi.org/10.3390/app9010178.
[25] F. Hosseinpour, P. Amoli, J. Plosila, T. Hämäläinen, H. Tenhunen, An intrusion detection system for fog computing and iot based logistic systems using a smart data approach, Int. J. Digital Content Technol. Appl. 10 (2016).
[26] L. Scrucca, M. Fop, T.B. Murphy, A.E. Raftery, mclust 5: clustering, classification and density estimation using Gaussian finite mixture models, R J. 8 (1) (2016) 205–233. URL https://journal.r-project.org/archive/2016-1/scrucca-fop-murphy-etal.pdf
[27] S. Darroudi, C. Gomez, Silvair: Information-centric networking a revolutionary approach to wireless lighting control, 2018. https://gallery.mailchimp.com/56e667868fdc46d69bb4d83b1/files/a7282f35-8e95-4ec0-834a-d094520fe68c/Whitepaper_ICN_10.10_final.pdf.
[28] K. Benkic, M. Malajner, P. Planinsic, Z. Cucej, Using RSSI value for distance estimation in wireless sensor networks based on ZigBee, Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on, (2008), pp. 303–306.
[29] M. Marks, E. Niewiadomska-Szynkiewicz, J. Kolodziej, High performance wireless sensor network localisation system, Int. J. Ad Hoc Ubiquit. Comput. 17 (2–3) (2014) 122–133.