Journal of Medical Systems (2019) 43: 324
https://doi.org/10.1007/s10916-019-1440-0

SYSTEMS-LEVEL QUALITY IMPROVEMENT

MARC: A Novel Framework for Detecting MITM Attacks in eHealthcare BLE Systems

Muhammad Yaseen [1] · Waseem Iqbal [1] · Imran Rashid [1] · Haider Abbas [1,2] · Mujahid Mohsin [3] · Kashif Saleem [4] · Yawar Abbas Bangash [1]

Received: 11 August 2018 / Accepted: 21 August 2019 / Published online: 16 October 2019

Abstract

Real-time and ubiquitous patient monitoring demands the use of wireless data acquisition through resource constrained medical sensors, mostly configured with No-input No-output (NiNo) capabilities. Bluetooth is one of the most popular and widely adopted means of communicating this sensed information to a mobile terminal. However, over simplified implementations of Bluetooth low energy (BLE) protocol in eHealth sector is susceptible to several wireless attacks, in particular the Man-in-the-Middle (MITM) attack. The issue arises due to a lack of mutual authentication and integrity protection between the communicating devices, which may lead to compromise of confidentiality, availability and even the integrity of this safety-critical information. This research paper presents a novel framework named MARC to detect, analyze, and mitigate Bluetooth security flaws while focusing upon MITM attack against NiNo devices. For this purpose, a comprehensive solution has been proposed, which can detect MITM signatures based upon four novel anomaly detection metrics: analyzing **M**alicious scan requests, **A**dvertisement intervals, **R**SSI levels, and **C**loned node addresses. The proposed solution has been evaluated through practical implementation and demonstration of different attack scenarios, which show promising results concerning accurate and efficient detection of MITM attacks.

Keywords No-input No-output (NiNo) device · eHealthcare · BLE · MITM attack · Sensors security · MARC

Highlights

?• Simulation of MITM attack using Gattacker on Bluetooth Low Energy (BLE) NiNo devices.
• Detailed comparison and critical analysis of existing solutions.
• Automated detection of MITM and cloned-node attacks using a novel mechanism.
• Design evaluation through implementation of multiple mitigation techniques

This article is part of the Topical Collection on *Systems-Level Quality Improvement*

✉ Haider Abbas
dr.h.abbas@ieee.org

[1] Department of Information Security, College of Signals, National University of Sciences & Technology, NUST, Islamabad 44000, Pakistan

[2] National University of Sciences and Technology NUST, Islamabad 46000, Pakistan

[3] Department of Avionics Engineering, College of Aeronautical Engineering, National University of Sciences & Technology, NUST, Islamabad 44000, Pakistan

[4] Center of Excellence in Information Assurance (CoEIA), King Saud University, Riyadh 12372, Kingdom of Saudi Arabia

Introduction

Numerous solutions have been proposed and implemented in eHealthcare for data acquisition and transmission. Majority of the solutions focus on operational side; however, security is normally a missing priority. Medical sensors used in eHealthcare applications lack in security features due to limited computation power, small form factor, and normally NiNo capabilities. As explained in [1], Bluetooth is primarily used in sensors to measure patient vital signs. In this research, Bluetooth Low Energy (BLE) is focused to analyze its security features – detecting and mitigating MITM attacks in BLE pairing mechanisms. For a complete eHealthcare overview, solutions, and security features comparison, a detailed survey is presented in [2]. As sensors are the data generation nodes, therefore, they are far more vulnerable to multiple attacks on physical, MAC, network and application. Multiple security risks are also introduced as mentioned in [1]. Secure data acquisition and its protection during transmission is of critical importance in this context. Due

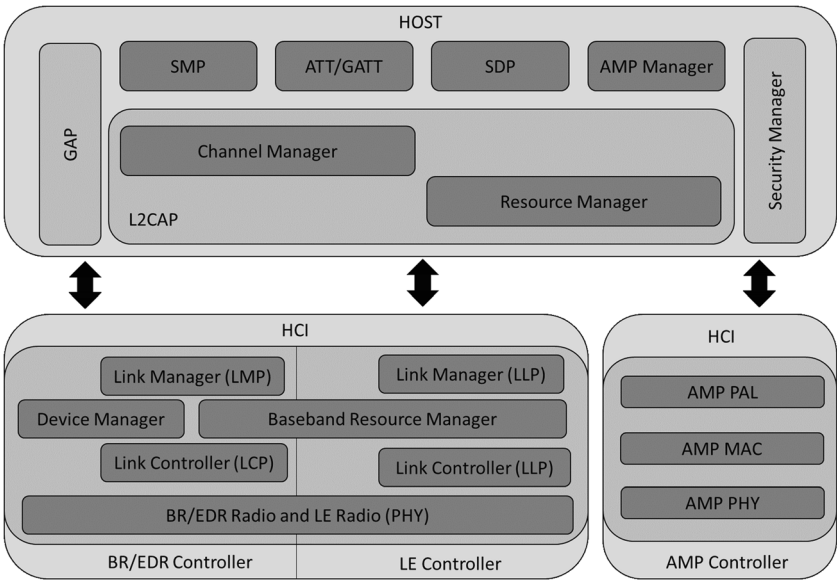

Fig. 1 Bluetooth General Device Architecture. Bluetooth has a layered architecture which includes host and controller. A BLE device may contain all or some of the layers defined above. Bluetooth specification details on the architecture of Bluetooth low energy devices

to heterogeneous nature of sensors deployed in eHealth sector, current methodologies need to be modified to work securely within the environment, while taking care of the criticality of patient health.

End nodes/sensors/devices have generally NiNo capabilities, meager computational resource, low energy consumption constraints, and miniaturized form factors. Because of these limitations, security features are often missing in eHealthcare systems. Bluetooth is one of the most used protocols in eHealth sector owing to its easy accessibility and compatibility with major hardware. Bluetooth supports multiple security features such as node authentication, message integrity, and multiple security modes; however, they are mostly not used because of limited I/O (input/output) capabilities of end sensors. Sensors are bound to use 'Just Works' pairing mechanism which is vulnerable to MITM attack and eavesdropping per Bluetooth standard [3]. Moreover, numerous attacks such as DoS/DDoS, Sybil, blackhole, wormhole and session hijacking, tampering is possible owing to malicious nodes which perform MITM attacks [4, 5]. Eavesdropping and MITM attacks have shown to be easy in protocols like Bluetooth, resulting in compromising of patient's privacy as well [6]. Consequently, there is a need for detecting and mitigating these MITM attacks before they compromise the sensing nodes in a Bluetooth piconet.

Table 1 BLE Pairing Combinations

Device 1	Device 2	Pairing Method Possible
Display Yes/No	Display Yes/No	Numeric Comparison
	Display Only	Numeric Comparison
	Keyboard Only	Passkey Entry
	No-input No-output	Just Works
Display Only	Display Only	Numeric Comparison
	Keyboard Only	Passkey Entry
	No-input No-output	Just Works
Keyboard Only	Keyboard Only	Passkey Entry
	No-input No-output	Just Works
No-input No-output	No-input No-output	Just Works

To the best of our knowledge, no research has been done that implements practical detection of spoofing and interception against MITM attacks, targeting end devices with NiNo capabilities. Moreover, most solutions recommend Out-of-Band (OOB) as pairing channel, discarding "Just Works" pairing method, but medical sensors have very small form factor and implementing OOB mechanism as well as (input/output) I/O capabilities on a very small end node is not a feasible option. Complementing existing security solution requiring I/O capabilities at both ends, this research primarily focuses to safeguard Bluetooth nodes with NiNo capabilities and 'Just Works' pairing method against MITM attacks. The proposed framework is unique in identifying MITM attacks and cloned nodes in NiNo devices at lower layers. It also suggests mitigation techniques to avoid passive and active MITM attacks. Our research stands out because of the following main contributions.

- We have simulated MITM attack with session hijacking and data manipulation on real hardware.
- We have detected MITM attack in BLE devices having NiNo capabilities.
- We have provided mitigation of MITM attacks in NiNo BLE devices using multiple techniques.

Table 2 BLE Literature Review

Sr.	Paper Reference	Classic(BR/EDR)/ BLE	Attacks Analyzed	Countermeasures Proposed	Shortfalls
1	Hataja [24]	Classic	- NiNo Attack - BT-SSP-OOB-MITM	- Additional window at user interface level - JW as optional and OOB as mandatory - Pairing in isolated environment - Using RF fingerprints	- BLE is not discussed - End devices need i/o capabilities for countermeasures
2	Guo [14]	Low Energy	- Battery Exhaustion	- Using Blacklist and priority list	- MITM protection is not presented
3	Saravanan [15]	Classic	- MITM attack	- OOB NFC	- No attack details - End devices are not tested
4	Wang [16]	Low Energy		- Stack Changes	- No MITM focus
5	Albahar [11]	Classic	- Password guessing - Reverse Engineering of end devices	- Minimum password length increase - Limited access to local database of end device	- No Practical implementation - BLE is not discussed - JW not discussed
6	AlMomani [17]	Classic	- Theoretical MITM attack	- ESSP	- Solution required that both devices should have I/O capabilities
7	Jasek [12]	Low Energy	- Spoofed Advertisement - Passive interception - Active Interception - Data manipulation - Replay attack - Brute force attack	- Whitelisting - Do not use Just Works pairing	- Practical detection of spoofing and interception detection mechanism is missing
8	Hataaja [18]	Classic	- Theoretical MITM attack	- OOB as mandatory - Add an extra window at the user end	- Practical detection of spoofing and interception detection mechanism is missing
9	Moon [19]	Classic	- Theoretical MITM attack	- Using signal interval as nonce for key generation	- Both devices need i/o capabilities - Does not cater just works method - Bluetooth classic is checked - Malicious node detected after pairing
10	Albahar [20]	Classic	- Theoretical MITM attack	- Practical implementation of additional window at user end	- No computation process explained - Devices need I/O capabilities
11	Ryan [13]	Low Energy	- Data Injection - Brute force - Signal Jamming	- No countermeasure explained	- Practical detection of spoofing and interception detection mechanism is missing
12	Gajbhiye [25]	Low Energy	- BT-SSP-MITM	- Additional public key setup before exchanging I/O capabilities	- Results in added latency in pairing of BLE devices - Additional Bluetooth specification changes
13	Gajbhiye [26]	Low Energy	- BT-SSP-MITM	- Physical jamming of MITM attacks - Delay in sending I/O capabilities after public key transfer	- Requires anti-jamming hardware at physical layer - Bluetooth specification changes required
14	Sun [27]	Classic	- BT-SSP-PE-MITM	- Introduction of nonce before pairing of Bluetooth devices	- Only Passkey pairing mechanism was discussed - Just works pairing mechanism issue was not discussed

Table 3 Comparison of MARC with other solutions

Sr. No	Author	Classic(BR/EDR)/BLE	Proposed solution against				Cloned Node Detection	Implementation of proposed solution	NiNo Capabilities Required
			MITM Attack	Denial of Service	Active Interception	Other Attack			
1	Haataja [24]	Classic	Yes	No	No	–	No	No	No
2	Guo [14]	Low Energy	No	No	No	Battery Exhaustion	No	Yes	Yes
3	Saravanan [15]	Classic	No	No	No	–	No	No	No
4	Wang [16]	Low Energy	No	No	No	–	No	No	No
5	Albahar [11]	Classic	Yes	No	No	Password guessing, reverse engineering	No	No	No
6	ALMomani [17]	Classic	Yes	No	No	–	No	Yes	No
7	Jasek [12]	Low Energy	No	No	No	–	No	No	No
8	Haataja [18]	Classic	Yes	No	No	–	No	No	No
9	Moon [19]	Classic	No	No	No	–	No	No	No
10	Albahar [20]	Classic	Yes	No	No	–	No	Yes	No
11	Ryan [13]	Low Energy	No	No	No	–	No	No	No
12	Gajbhiye [25]	Low Energy	Yes	No	No	–	No	Yes	N/A
13	Gajbhiye [26]	Low Energy	Yes	No	No	–	No	Yes	N/A
14	Sun [27]	Classic	Yes	No	Yes	–	No	No	No
15	Proposed Solution	Low Energy	Yes	Yes	Yes	–	Yes	Yes	Yes

The rest of the paper is organized as follows. Section 2 presents the background, while section 3 covers the related work. Section 4 explains our novel proposed solution with algorithmic description. Section 5 covers the implementation including testbed details, MITM scenario, and usage of MITM Gattacker tool. Detailed results and analysis of presented metrics and mitigation techniques are discussed in section 6. Section 7 is about discussion and recommendations, and the conclusion and future work directions are presented in section 8.

Background

Bluetooth operates in unlicensed spectrum of 2.4 GHz radio frequency, which is the part of Industrial, Scientific, and Medical (ISM) radio band. BLE splits frequency in to 40 channel of 2 MHz width. Three channels are used for advertising, and remaining 37 channels are used for data communication. Figure 1 from [7] shows Bluetooth general device architecture for 4.0 and above versions.

The GAP roles define the system's topology. A BLE node may perform the role of a peripheral or a central device. The Peripheral is the device that waits for others to connect to it and is normally optimized for low-energy (runs on small, long life batteries). The Central is the device with the initiative; it looks for peripherals and connects to them. It usually does not have power consumption constraints (easily rechargeable). General procedure which is followed during pairing of two BLE devices is explained below.

Peripheral device broadcasts the advertisement. The scan request is made by the central device for which a response is unicasted to the central device by peripheral. Central device then initiates a connection request, besides inquiring the features supported by peripheral device. Peripheral device sends its I/O capabilities in response. After profiling of devices, services and characteristics are exchanged between the two devices and devices will be connected to each other at the end. Data exchanges are performed during the connection such as notifying, updating, reading or writing the data. On completion of the data exchange, end of connection is requested, followed by its termination.

If both devices support a common Out of band (OOB) technology such as NFC/Tethering, then OOB method can be used to pair with each other. If at least one device supports input capability such as keyboard, and other device has some other output capability such as a display (or keyboard input as well) then passkey entry pairing method can be used to pair between the devices. If both end devices have output capability such as displaying six (06) digit number and both devices have input

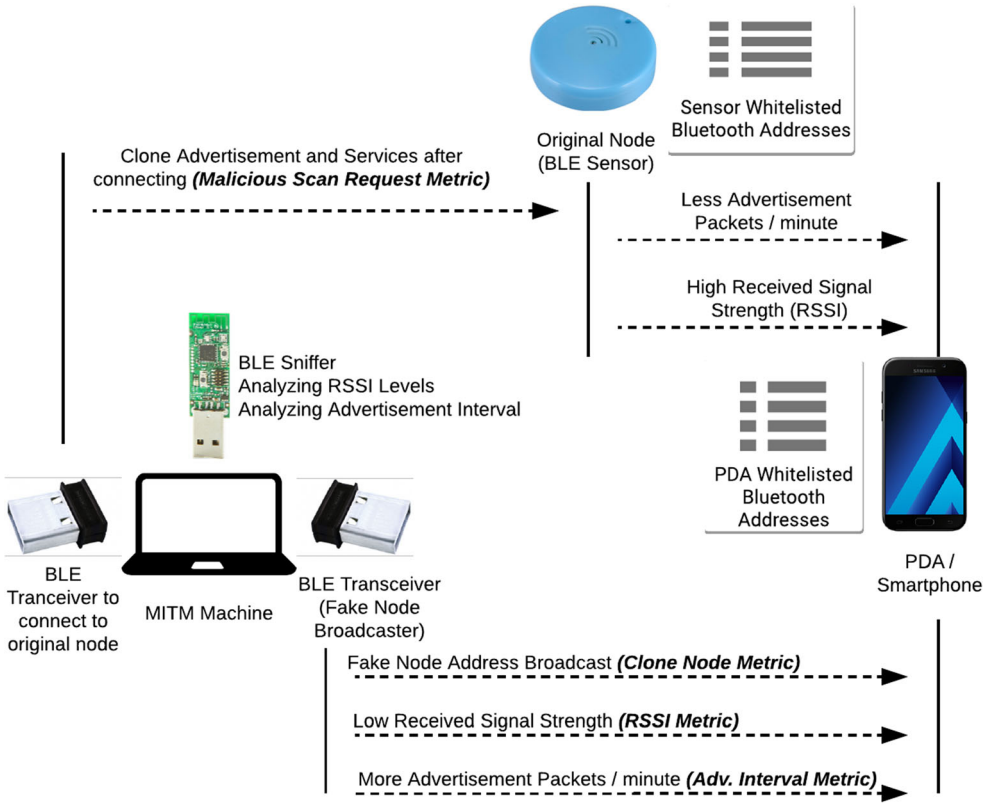

Fig. 2 Proposed Solution Complete Architecture. The figure shows the MITM attack flow and how our proposed solution mitigates the MITM attack

capability (ability to enter "yes" or "no" by any means) the numeric comparison pairing method can be used. If at least one of the communicating device does not support any input/output capability i.e., a keyboard or a display, both of them are constrained to only use "just works" pairing method as explained in the Bluetooth specifications. This method is vulnerable to MITM attacks and is the weakest of all pairing methods from security perspective. In legacy "just works" pairing method, the

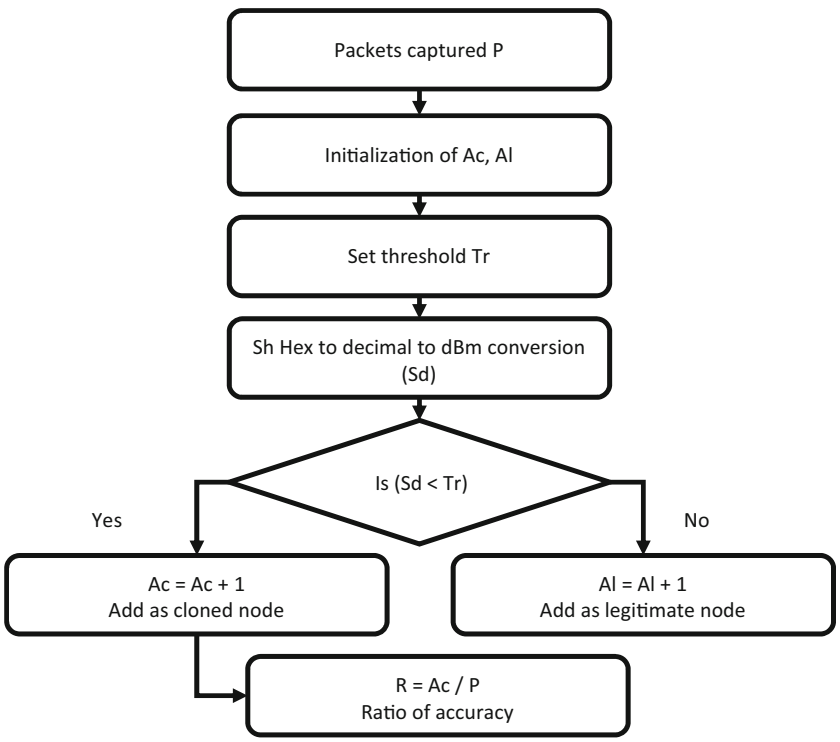

Fig. 3 RSSI Metric Flow Diagram. The flowchart shows how low RSSI levels lead to cloned node detection and ongoing MITM attack

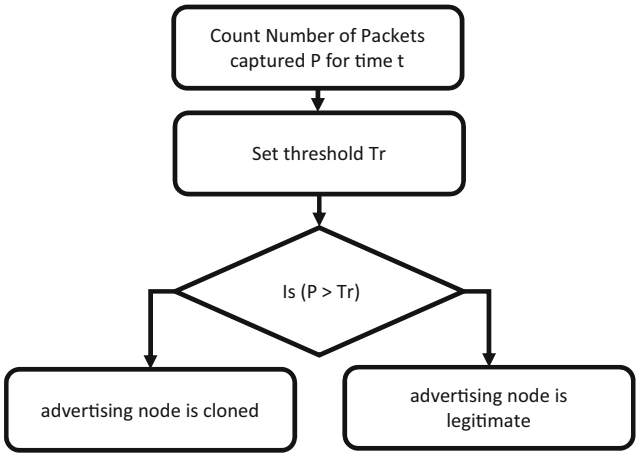

Fig. 4 Advertisement interval Metric Flow Diagram. The flowchart shows how increased advertisement frequency leads to cloned node detection and ongoing MITM attack

Transient Key (TK) (which is used for initialization of session) is set to zeros (generally, "0000"). This removes the requirement of obtaining TK to generate Short Term Key (STK) for any eavesdropper or MITM attacker.

For "just works" secure connection pairings, a numeric comparison pairing method is used. Due to the lack of I/O capabilities of the remote node, the user is not revealed with 6 digit values and final assurance checks are not performed as well. Just Works method generates an unauthenticated Long Term Key (LTK) and does not provide MITM protection during pairing. For details about keys used during pairing, readers are referred to Bluetooth core specifications in [3]. Table 1 summarizes the paring methods based on the I/O capabilities of devices.

Table 1 show that if any one device has NiNo capabilities, then we are bound to use Just Works pairing method which is vulnerable to MITM and eavesdropping attacks.

Related work

Many commercial eHealthcare systems use Bluetooth for sensors to sink node/smartphone communications (see for example [8]). To the best of our knowledge, sufficient literature is missing on Bluetooth security in low energy domain. As endorsed by authors in [9] with the help of multiple real life attacks, MITM attacks on BLE devices are easier to execute. It was also highlighted in the cited research that MITM attack can also be launched on NiNo devices even when link layer encryption is in effect. In addition, authors in [10] provided a security testing framework using a combination of multiple open source tools to simulate multiple attacks on Bluetooth devices.

Authors in [11] focused on MITM attacks, described vulnerabilities that facilitate these attack in Secure Simple Pairing (SSP) and proposed generic solutions. Their work also highlighted solutions applicability in practical deployments. However, BLE security was not discussed in general and Just Works pairing method was not explored in detail. Another work [12] presented an overview of BLE stack, its vulnerabilities, BLE attacks on multiple pairing methods, and presented a tool "Gattacker" to implement MITM attacks. This tool is used further to simulate MITM attacks in our research, which additionally offers viable mitigation techniques as well.

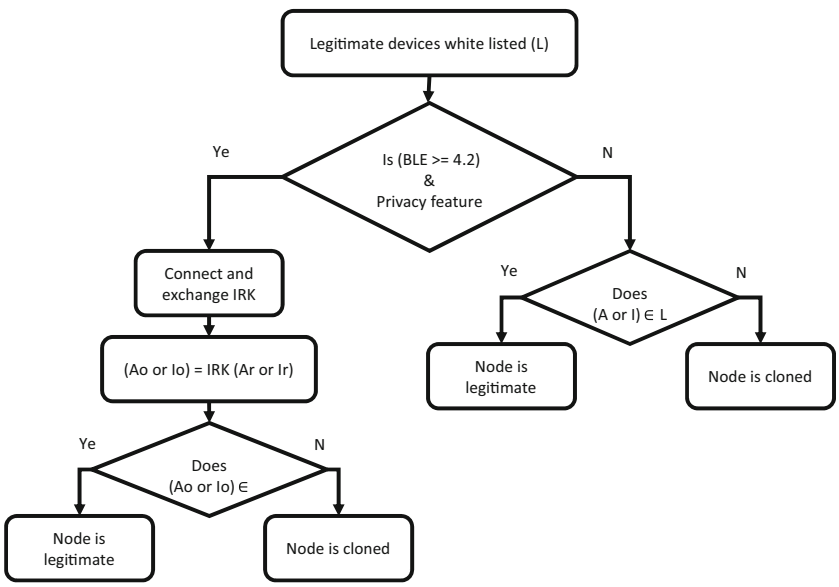

Fig. 5 Cloned Node Address/ Malicious Scan Request Metrics Flow diagram. The flowchart shows how Bluetooth address in advertisement/scan request packets lead to cloned node detection and ongoing MITM attack. It is to note that this flowchart explains two metrics namely cloned node address and malicious scan request

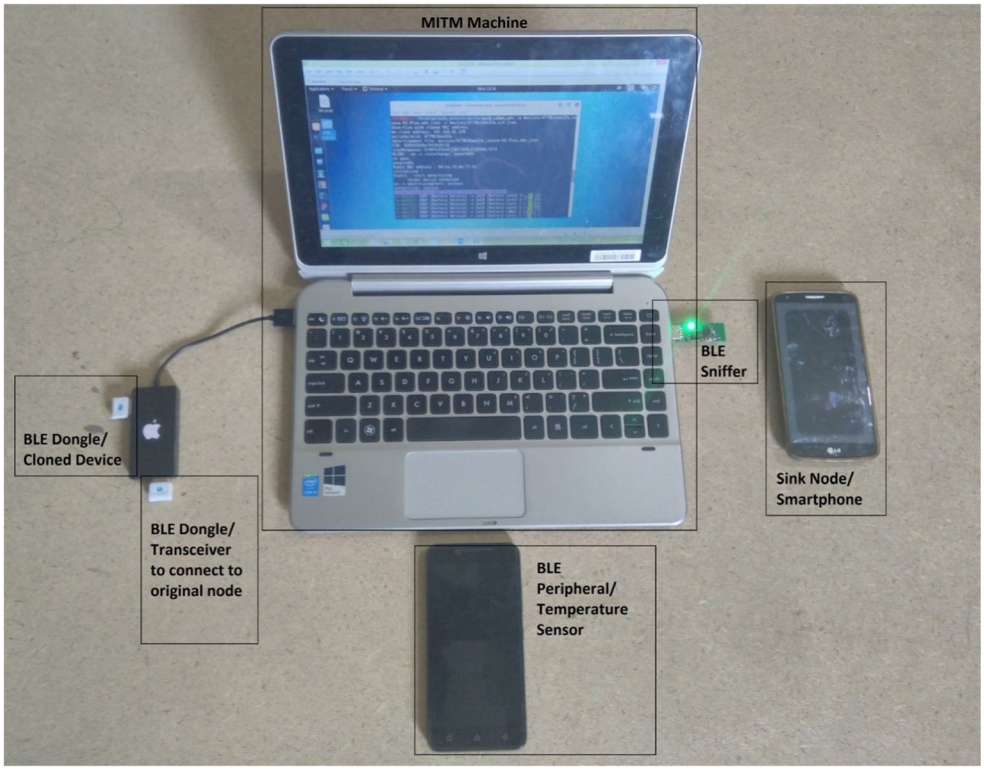

Fig. 6 Scenario Setup Overview. The figure shows the test bed, components, their usage and placement for simulation. These include six components including hardware machine, virtual Linux machine, BLE sniffer, BLE dongles, peripheral node and sink node

Work done in [13] demonstrated practical implementation of BLE sniffing, packet injection, and Brute forcing on BLE packets. Authors, however, did not mention any countermeasure for the attacks. Two practical MITM attacks on Bluetooth Secure Simple Pairing (SSP) were presented namely NiNo attack and BT-SSP-OOB-MITM attack, focusing on OOB channel. The authors proposed multiple countermeasures such as additional window at user interface level, pairing in isolated environment, and using RF fingerprints; however, the proposed solution was only applicable over devices with I/O capabilities. Another work [14] proposed a solution to counter battery exhaustion attacks by maintaining blacklists and priority lists at all devices, but root cause i.e., MITM attack was not focused in the research.

A novel Bluetooth MITM attack in [15] was also presented on SSP using OOB association model. Authors proposed NFC as an OOB pairing channel, but end devices were not tested for the solution. Another research paper focused on privacy enhancements in BLE advertising [16]. The proposed solution required some prerequisites such as pre-stored database in both initiator and responder, and prior connection of devices was needed for transfer of Identify Resolving Key (IRK). Moreover, some stack changes were also proposed. We agree to the idea that protection should be done at the advertisement level to stop MITM attacks, and the same approach is followed in this research as well. Authors in [17] presented ESSP (Enhanced Secure Simple Pairing mechanism) to counter MITM attack; however, their solution required I/O capabilities in both devices.

Authors in [18] presented a theoretical MITM attack and proposed OOB as a mandatory option, Moreover, adding extra window at user interface was also proposed but practical solution was missing in the research. Authors in [19] proposed a unique solution to prevent MITM attack using physical signal interval as a nonce for key generation; however, both end devices need I/O capabilities and malicious MITM node is not detected before pairing to the central device. The authors in [20] presented an enhanced Just Works pairing model to prevent MITM attack using additional window at user interface level; however, solution required I/O capabilities at both ends and it also lacked computation process mechanism.

Authors in [21] presented a security framework for BLE devices in which they proposed initial keys transfer using out of band channel like QR codes. There are other notable solutions to mitigate MITM attacks such as presented in [22] in which the authors proposed the use of complex user tokens, stored on servers and end user only had to remember easy passwords. This technique offers resistance against well-known MITM and offline password guessing attacks. Another notable technique to alleviate MITM attacks was presented in [23], which propose to control valid flow of information by controlling the frequency of noisy packets for a communication channel.

Table 2 summarizes the literature discussed in this section.

Table 3 shows a comprehensive comparison between current solutions and our solution.

As shown in Table 3, there is no solution that encompasses all security aspects of MITM, as covered by this research. As mentioned in Table 2, most solutions focus on classic Bluetooth rather than low energy. Most solutions simulate MITM attacks, but very few solutions propose remediation/mitigation techniques to stop this attack. Mitigation of Denial of service attacks, active interception of packets and cloned node is also not presented in many of the existing solutions discussed in Table 2.

Proposed solution

If MITM attack can be detected and blocked, then many associated attacks such as Spoofing, Wormhole, Selective forwarding, Sybil, Blackhole, energy exhaustion, PIN cracking, and Brute forcing attack can also be countered, because the mentioned attacks require an attacker to first become the member of the network. If a solution exists that detects intrusion and prevents it before a malicious node enters in to the network, then major attacks can be prevented, thus achieving data acquisition and transmission protection as a result. It can therefore, be induced that an effective technique to block many of these attacks is to detect and prevent network entry of malicious nodes; thus resulting into MITM attack mitigation. Readers not familiar with these attacks/techniques, are referred to [28] for further details on the mentioned attacks.

As mentioned above, MITM attacks are the root cause of many attacks. The solution proposed by our research focuses on mitigating MITM attacks. MITM attack is simulated first using Gattacker MITM tool (explained in section 5). Denial of service and eavesdropping of the broadcasted data will be shown as a result of successful MITM attack in section 6. Our solution mainly focuses on devices having NiNo capabilities. It is because of this reason that 'Just works' pairing method must be used for pairing. We present a comprehensive solution that can be used to analyze BLE device's broadcast packets, and subsequently detects malicious/bogus nodes based on the analysis of these packets. Attacks that are mentioned above can therefore, be mitigated by preventing MITM attacks. Moreover, NiNo capabilities and no hardware changes are required in our solution by end devices to detect and prevent malicious nodes from entering in to the piconet.

A comprehensive solution named as MARC, has been proposed that comprises four metrics. First two metrics are quantitative, while the other two are based on qualitative analysis.

- Accuracy of identification of cloned packet based on the RSSI level
- Count of advertised packets in a specified amount of time.
- Identification of a malicious cloned node Bluetooth address
- Malicious connection requests sent by cloned node.

Last two metrics cannot be quantified based on the nature of these metrics. They can only be implemented by using white listing of legitimate devices. Figure 2 helps in visualizing our proposed solution.

No solution, according to the authors' knowledge, implemented/simulated to mitigate the MITM attack in BLE. Many of them however, proposed theoretical mitigation techniques. Furthermore, no MITM mitigation technique catered for absence of input/output capabilities of end sensor nodes in their solutions. As analyzed and compared in section 3, this framework is unique in identifying MITM attacks, cloned nodes in NiNo devices, and suggests mitigation techniques to avoid passive and active MITM attacks.

Algorithmic description and procedure of four metrics is explained below.

Metric 1: RSSI level

The first metric to identify cloned node is based on the Received Signal Strength Indicator (RSSI). When sensor is relatively close to the sink device or smartphone, the RSSI received is higher as compared to the cloned node, which is relatively far from the smartphone. Based on this principle, cloned and legitimate nodes are differentiated.

As explained in the algorithm [1], the advertisement packets P are received at the smartphone. Error packets are removed by checking the Frame Check Sequence (FCS) value at the end of packets. Then advertisement statistics for cloned (Ac) and legitimate (Al) are initialized. RSSI Threshold value (Tr) is set based on the environment. This value can be different in every scenario and is based on average distance between smartphone and Bluetooth medical sensor. Required RSSI value is extracted from packets, and then converted into decimal. If the extracted decimal value is less than the threshold value, then the cloned node variable is incremented. If extracted decimal value is greater than the threshold value Tr, then the legitimate node variable is incremented. For more accuracy ratio of cloned node variable and total packets analyzed can be calculated over a larger number of packets (Figs. 3, 4, 5 and 6).

Algorithm 1: RSSI level

```
Input:
P:  Number of packets captured and analyzed per experiment
Ac: Number of Cloned node advertisements
Al: Number of Legitimate node advertisements
Tr: RSSI Threshold
Sh: Received signal strength indication value extraction (hex)
Sd: Received signal strength indication value (decimal)
R:  Ratio of cloned vs total number of packets captured
Output:
Decision whether the node is cloned or legitimate
Procedure (Ac, Al, Tr, Sh, P):
BEGIN:
Ac = Al =0;
//Set threshold Tr according to the environment
Sd =-93+HextoDecimal(Sh)+1//TI CC2540 module conversion formula
if(Sd < Tr)
//If RSSI value is less than specified threshold add as a cloned node
   Ac = Ac +1
else
   Al = Al +1//else add as a legitimate node
endif
R = Ac / P //Ratio of clone tagged packets and total number of packets
END
```

Metric 2: Advertisement interval

Cloned node is detected based on the advertisement frequency or interval between the advertisements. MITM attacker tries to deliver its cloned advertisement packets to the sink node/ smartphone before original node, thus advertising far more packets in a specified time as compared to a legitimate node. This attack is synonymous to rushing attack where fake packets reach the destination before original packet.

As explained in algorithm [2], advertisement packets *P* are received at the smartphone. Error packets are removed by checking FCS value at the end of packets. Threshold value *Tr* is set based on the environment. It is to highlight that this value is based on average number of advertisement packets received in a certain amount of time *t*. If number of received packets is larger than threshold value *Tr,* the advertisement node is cloned and when it is less, the advertisement node is a legitimate one.

Algorithm 2: Advertisement Interval

```
Input:
P:    Number of Advertisement packets recorded in specified amount of time t
t:    Specified time to record number of packets
Tr:   Advertisement Threshold
Output:
Decision whether the node is cloned or legitimate
Procedure (P, t, Tr):
BEGIN:
Stream of advertisement packets P captured for specified amount of time t
//Set threshold Tr according to the environment
if (P > Tr)Advertisement Node is Cloned
//If recorded packets within time t are greater than specified threshold
advertisement node is Cloned
Else Advertisement Node is Legitimate
endif
END
```

Metric 3: Cloned node address

This metric uses advertised Bluetooth address to detect a cloned node. When the cloned node advertises with its own address, then the cloned node can be detected. MITM attacker can clone device services and characteristics, and it can clone Bluetooth address as well. However, Bluetooth 4.2 introduced a privacy feature in which Bluetooth device advertises a random address in each new session, and IRK key is used to resolve device original Bluetooth address after connection is established. This metric can be used when an attacker is unable to spoof Bluetooth address.

As explained in algorithm [3], advertisement packets P are received. The Advertised Bluetooth address is extracted from Adv PDU header (*TxAdd* value). In addition, this metric requires a list L of whitelisted Bluetooth addresses, which are already stored on smartphone. If Bluetooth device supports version 4.2 and above, in addition to using the privacy feature, then smartphone must first connect to the Bluetooth device. Next is to extract the original Bluetooth address using *IRK* and compare against the list of whitelisted addresses. If there is a match then the node is a legitimate otherwise it is a malicious node. In case, privacy feature is not used, then the advertised address is compared directly against the whitelisted address list.

Algorithm 3: Advertisement Interval

```
Input:
P:    Advertisement packets
Ar:   Advertised Bluetooth address random (random address can be identified
      by checking Adv PDU header's TxAdd value)
Ao:   Advertised address Resolved using IRK
A:    Advertised Bluetooth address
L:    White listed addresses list

Output:
Decision whether the node is cloned or legitimate

Procedure(P, Ar, Ac, A, L):
BEGIN:
//Legitimate device Bluetooth address is white listed in sink node/Smartphone
during initialization phase and in an isolated environment
//Peripherals start advertising
if (Both nodes have version >= 4.2 & devices are using privacy feature)
      Connect to advertising node and exchange IRK
      Ao= IRK (Ar)
      if (Ao ∈ L) Node is legitimate
      else Node is cloned
      endif
elseif (A ∈ L) Node is legitimate
else Node is cloned
endif
END
```

Metric 4: Malicious scan request

The cloned sink node is detected based on the initiator Bluetooth address. This unique metric checks the authenticity of sink node/smartphone from which the request is initiated. Before sending in a connect request, scan request is sent by the smartphone on which advertising nodes respond with their services, characteristics, and capabilities. Malicious originator/smartphone can be identified by comparing its Bluetooth address against whitelisted Bluetooth addresses stored in the advertising node.

As explained in Algorithm 4, the scan requests from sink node/smartphone are received. Bluetooth address is extracted from Scan Request PDU header *Scan*Avalue. In addition, this metric requires a list L of whitelisted Bluetooth addresses stored already on advertising node. If sink node supports version 4.2 and above in addition to using the privacy feature, then advertising node must

connect to the sink node. Next is to extract original Bluetooth address using *IRK* and compare it against the list of whitelisted addresses. If there is a match, sink node is legitimate otherwise it is a malicious node. In case, privacy feature is not used, then the advertised address is compared directly against the whitelisted address list.

Algorithm 4: Malicious Scan Request

```
Input:
P:    Connect Request Packet
Ir:   Connection Initiating node Bluetooth address random (random address can
      be identified by checking Adv PDU header RxAdd value)
Io:   Connection Initiating node address Resolved using IRK
I:    Connection Initiating node Bluetooth address
L:    White listed addresses list
Output:
Decision whether the sink node is cloned or legitimate
Procedure(P,Ir, Io, I, L):
BEGIN:
//Legitimate device Bluetooth address is white listed in advertising node
during initialization phase and in an isolated environment
//Peripherals start advertising
if (Both nodes have version >= 4.2 & devices are using privacy feature)
      Connect to smartphone and exchange IRK
      Io = IRK (Ir)
      if(Io ∈ L)Sink Node is legitimate
      else Sink Node is cloned
      endif
elseif(I ∈ L)Sink Node is legitimate
else Sink Node is cloned
endif
END
```

Implementation

This section covers the scenario setup for performing the experiments alongside details of Gattacker module; it also presents the adopted approach for performing sniffing of BLE packets and MITM attack. Moreover, testbed components used for the solution are also briefly discussed with their specifications and versions used.

Testbed

Following four components are used to create our testbed for simulating MITM attacks. Fake/cloned node detection and blocking of these nodes is done based on the metrics defined in section 4 (Table 4).

First component is CSR8510 chipset which is a Bluetooth v4.0 single-chip radio and baseband IC for PCs and consumer electronics devices. It operates on Bluetooth version 4.0 having both capabilities of classic Bluetooth and BLE. Its maximum transmitter power is 9.75 dBm and receiver sensitivity of -91 dBm. Maximum input voltage is 4.8 V. It supports USB 2.0 interface for connecting to laptop and other hardware appliances. The core chipset is developed by Qualcomm and multiple vendors use this chip to develop BLE enabled dongles. These dongles are used to simulate MITM attack, which is explained in sub-section 5.3.

Second component is Texas Instruments' CC2540 USB dongle which was used for analyzing BLE packets. This BLE sniffer is used to capture and enable BLE packets from a PC. It is also used for analyzing BLE packets and for system level debugging. The dongle comes preprogrammed as a packet sniffer. The SmartRF Packet Sniffer application was used for interfacing BLE sniffer with the PC. The PC software application that comes with the device is used to display and store capture radio packets by RF listening device. USB interface is provided to connect it to the PC. Multiple protocols including BLE and Zigbee are supported. The BLE sniffer filters and decodes packets along with the options of filtering and storage to a binary file format. The machine that was used for interfacing packet sniffer was installed with Microsoft Windows 8.1 Pro ×64 bit. It has a processor of Intel Core M-5Y10C clocked at 1Ghz, 4GB of RAM and 32GB of internal storage.

Table 4 Testbed description and specifications

Sr.	Component	Description	Specification
1	Physical Machine	Used for conducting all experiments. All hardware, virtual servers and software was installed on this machine.	Microsoft Windows 8.1 Pro ×64 bit. Processor of Intel Core M-5Y10C clocked at 1Ghz, 4Gb of RAM and 32Gb of internal storage.
1	CSR8510 Chipset	Used for cloning BLE devices and broadcasting fake advertisement packets	Bluetooth v4.0 single-chip radio
2	CC2540 USB Dongle	Used for eavesdropping BLE packets	Processor Intel Core M-5Y10C clocked at 1Ghz, 4Gb of RAM and 32Gb of internal storage
3	Linux Machine (Virtual)	Used for running Gattacker tool	Kali linux version 2016.2 running on VMware workstation version 12.5.2. 1 virtual processor with 1Gb of RAM and 80Gb of hard disk
4	Android Device	Hardware used for running applications	Lenovo K5 Plus with android version 5.1.1
5	nRF Connect	Used for broadcasting cloned advertisement packets	–
6	BLE peripheral simulator	Used for broadcasting cloned advertisement packets	–

Third component comprises a couple of Linux machines which were setup in a virtualized environment on VMware workstation version 12.5.2. Linux flavor of Kali 2016.2 is used for conducting the experiments. Both virtual machines had 1 virtual processor allocated with 1GB of RAM and 80GB of hard disk allocated to each. Linux machines were connected on a virtual VMware network. Software libraries used by these machines are explained in sub-section 5.3 along with the description of the Gattacker tool.

Fourth component is nRF connect application by Nordic semiconductors which is used for BLE peripheral connections, while BLE peripheral simulator by WebBluetoothCG was used for simulation of BLE peripheral. These two applications are easily available on google play store; the mobile phone used was Lenovo K5 plus having android version 5.1.1.

Scenario setup

Scenario was setup using two Kali Linux machines being used for MITM attack. Peripheral was simulated on an android device with app call “BLE Peripheral Simulator”. Listener was also an android device with “nrf connect” app as a listener. Linux machines were connected with a BLE dongle CSR 8510 each. For sniffing the BLE packets Texas Instruments BLE Packet Sniffer was used. Figure 6 shows the overall setup used for conducting the experiments.

Linux machines were configured in VM Workstation each with a CSR 8510 BLE dongle. TI BLE Packet sniffer was connected with Windows 8.1 machine. Multiple experiments were conducted (detailed in section 6) to simulate the MITM attack and analyze the captured BLE packets for detection of malicious cloned node. A Temperature sensor peripheral was simulated, cloned, and tested in the experiments.

Gattacker

We used Gattacker MITM tool to perform attack simulation and analyzing BLE packets [12]. It requires Node.js, Noble and Bleno libraries as a prerequisite. Our experimental setup used version 4.7.2 of Node.js, version 1.8.1 of Noble and version 0.4.2 of the Bleno library. In our scenario, we used two Kali Linux machines to set up central and peripheral modes. Linux machine acting as central, is connected to peripheral acting machine using Websocket layer. In the subsequent text, central machine refers to Linux machine acting as central device in Gattacker tool, and peripheral machine refers to Linux machine acting as peripheral device in the tool.

Gattacker consists of three main json modules that communicate with each other to perform MITM attack. These functions along with their main objectives are briefed below. The flowchart given in Fig. 7 explains the overall process of simulating MITM attack.

Results and analysis

Based on the testing setup explained in section 5, this section discusses simulation results. Based on the setup, tools, and techniques explained in section 5 we were able to detect MITM attacks as well as the cloned nodes.

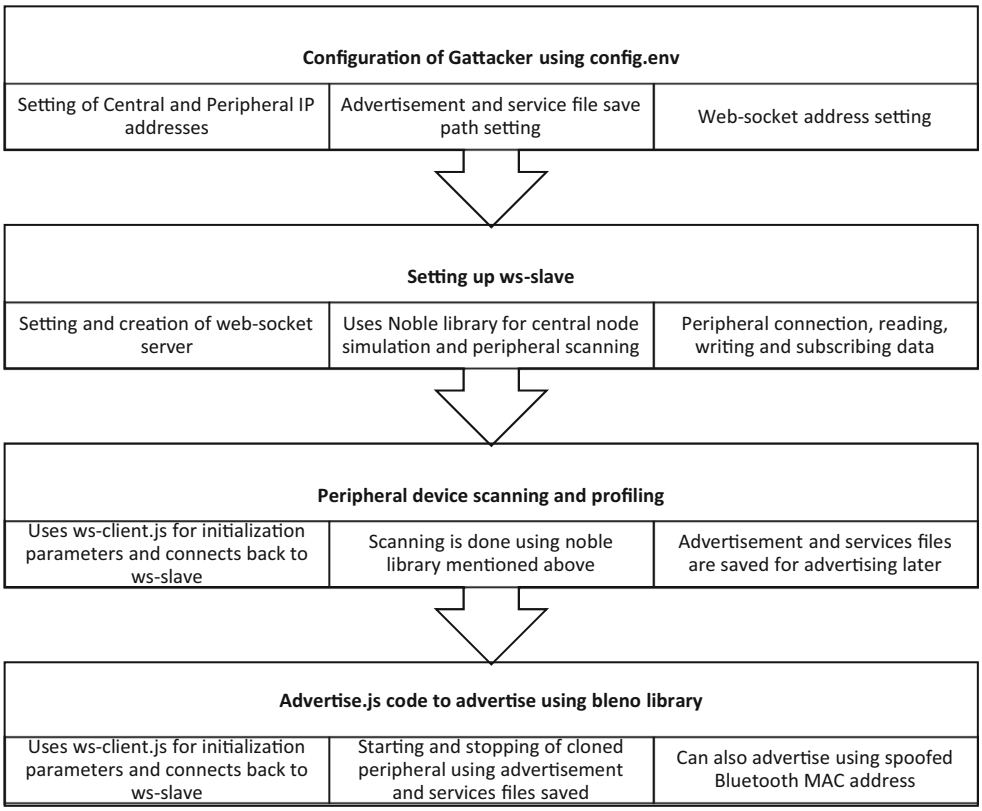

Fig. 7 Gattacker working overview. The figure shows Gattacker flow and its associated modules and libraries for performing MITM attack

MITM detection metrics

The efficiency of four anomaly detection metrics leveraged by MARC was analyzed using the test-bed setup described in section 5. A combination of these evaluation metrics lead to the detection of MITM attack and the cloned nodes. Consequently, we have proposed a comprehensive solution that can secure data acquisition and transmission in eHealthcare systems. The experiments were performed based on an assumption that no peripheral BLE device has any input/output capability. All metrics were extracted from BLE peripheral advertisement packets.

We evaluated multiple metrics for detection of cloned notes and based on high accuracy and effective detection mechanism, the details of only four metrics are discussed below. Some of the other metrics considered by our experiments include: (i) Bluetooth address in the advertisement packets, which was declared unsuitable due to built-in privacy features; (ii) Entropy of received advertisement packets, which failed to credibly differentiate between original and cloned node packets because of multiple random values in advertisement instances; and (iii) Checking for read only Universally Unique Identifiers (UUID), but manufacturers did not provide access to these UUIDs and implementation of this feature had a lot of variations at manufacturer's end. Selection of optimal metrics with desired results was based on a series of iterative experiments under different experimental setups and conditions. Subsequently, suitable metrics were shortlisted based on the success ratio, false positives/negatives, usability and effectiveness of detection mechanism.

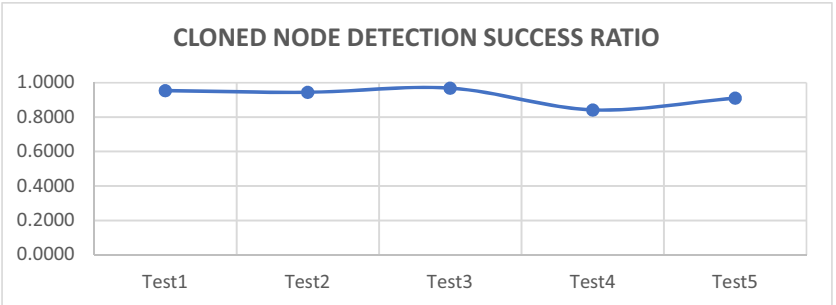

Fig. 8 Cloned node detection success ratio. The figure shows an average accuracy of 93% based on the five experiments performed using RSSI detection metric

Table 5 Cloned Node detection success ratio

Sr No	Test	Total Packets	RSSI < -50 dBm Cloned Node Detection	RSSI > -50 dBm Original Node Detection	Detection Accuracy (Cloned Detection/ Total Packets) %
1	Test1	500	477	23	95.40
2	Test2	500	472	28	94.40
3	Test3	500	484	16	96.80
4	Test4	500	421	79	84.20
5	Test5	500	455	45	91.00
				Accuracy Average	92.36%

Metric 1: RSSI levels

First, MITM attack is performed as mentioned in sub-section 5.3. One BLE dongle from MITM machines connects to original BLE device, and the second BLE device after cloning original device services and characteristics started advertising the same data. Cloned node starts advertising and BLE packet sniffer (acting as a RSSI measurement unit) is listening to the advertisements. It is to highlight that RSSI measurements are made on random movement of sensor node. Distance between sending and receiving nodes is also random during the experiment.

In our experiment, all packets analyzed were from a cloned node. We have set a threshold level of -50 dBm to detect the cloned node. This value is chosen based on testing environment explained in sub-section 5.2. Moreover, this threshold must be set based on the environment and sensor nodes used. BLE nodes motion trajectory during the experiments was random thus simulating real world patient scenario. If RSSI level is below the threshold value, then node advertising is cloned, otherwise it is an original node. Based on the five experiments with 500 recorded advertised packets each, we achieved a cloned node detection accuracy of 92.36% and only 7.64% packets were falsely reported as original advertisement as tabulated in Table 5.

Figure 8 shows cloned node detection success ratio based on the experiments conducted.

Metric 2: advertisement intervals

In this experiment, we need to record number of advertisement packets. We have compared packet count from a legitimate node, and a cloned node. Gattacker MITM tool is used to perform MITM as in the RSSI level metric. To further verify that only cloned advertisements reach the required destination, Gattacker also connects to the original node so that it stops advertising as stated in the standard, "*When node connects to a master device, it stops advertising*". To connect to sink node/smartphone, attacker first reduces the advertisement interval. As in sub-section 5.3, we have performed MITM attack and recorded advertisement intervals before and after the attacks as shown in the Fig. 9.

As shown in Fig. 9 advertisement interval in original advertisement (left snippet) is 259 ms, while cloned node advertises (right snippet) at an interval of 22 ms.

Another way of calculating this metric is through advertisement frequency of packets and it can be calculated at sink node/smartphone end. It is calculated based on the number of packets advertised in each interval of time. To save energy, sensor nodes limit advertisement packets i.e., they advertise minimum specified amount of advertisement packets. However, to reach sink node/smartphone first attackers increase advertisement frequency. We conducted five experiments to count the number of advertised packets by a legitimate node as compared to a cloned node. Table 5 shows that cloned node advertises a lot more packets (average is 2614) as compared to an original node (average is 236). These packets were recorded in a 60 s time interval each (Table 6).

Figure 10 shows a huge difference between the numbers of packets advertised by an original node as compared to a cloned node. It is evident from the table that cloned node advertises on average 1100% more packets as compared to the legitimate node.

Metric 3: clone node address

As shown in Fig. 11, cloned node is advertising with its own address rather than original device Bluetooth address.

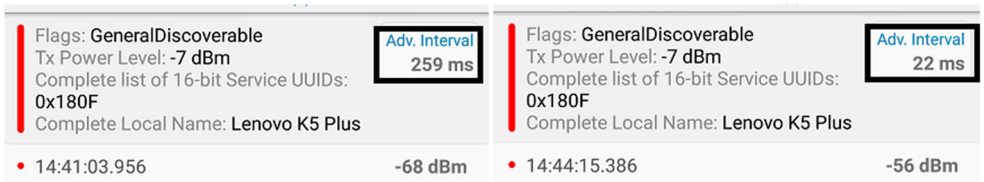

Fig. 9 Advertisement Interval Comparison between original and cloned node

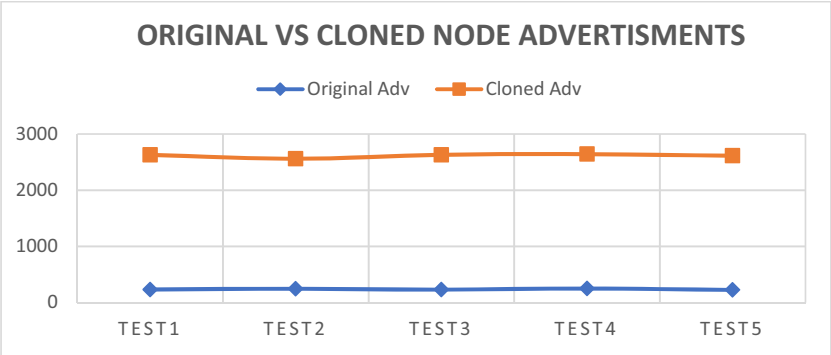

Fig. 10 Original vs Cloned node advertisement count. The figure shows that cloned nodes advertise 11 times more as compared to original node

This metric however, can also be used after connecting to the peripheral node. It assumes that original nodes, Bluetooth addresses are already whitelisted in to sink node/smartphone in an isolated environment before the system is initialized and starts advertising. Moreover, Bluetooth devices support version 4.2 and are using privacy feature in which original device address is only resolvable using IRK. Advertisement packets contain a random Bluetooth address. After a node connects to the sink node/smartphone, cloned node is detected from whitelisted addresses after resolution of connected Bluetooth address using IRK. If both devices (master and slave) support Bluetooth version 4.2, then original nodes will never advertise their original Bluetooth addresses; consequently, the attacker will never be able to detect original node address, and thus, cloned nodes will be detected.

Metric 4: malicious scan requests

Last metric to detect cloned node is through multiple scan requests. In normal situations, a sensor node advertises and smartphone/sink node requests a scan to detect device preliminary data such as services and characteristics. Figure 12 shows the scan request packet.

As we can see in Fig. 12, PDU type is a scan request; address of device requesting the scan is mentioned in *ScanA* field; and address of device to which request is unicasted is mentioned in *AdvA* field. As shown in Fig. 13 malicious node sends a scan request packet to the original node. As mentioned in MITM attack steps, malicious node sends a connect request packet to connect to the original node. If a node is unicasting scan request with a Bluetooth address that is not whitelisted, then it is a malicious node. This metric assumes that legitimate devices whitelisting is done before and all broadcasted packets are analyzed by sink node/smartphone so that these malicious scan request packets can be detected.

This figure shows the Bluetooth addresses in scan and connects requests from non-whitelisted Bluetooth address.

Discussion and recommendations

Based on the metrics described in Section 4 to detect intrusion, we present some recommendations that can be implemented to mitigate these MITM attacks. These mitigation techniques can be implemented on sink node/smartphone that have more computational power and energy. Packets advertised need to be analyzed at raw packet level on sink node/smartphone; and based on the simplicity of metrics, it will not add delay, and computational and energy overhead.

Directed advertising

Connectable directed advertising can be used to advertise only to the specified sink node/smartphone, as mentioned in Volume 6 part B section 2.1.3.2 [3]. If only specified smartphone/sink node can receive a directed advertising, then no attacker/eavesdropper is able to listen to the advertisements and thereby clone the device services and characteristics. This type of advertisement, however, does not contain any data in advertised packet. Although this is not very common,

Table 6 Advertisement count comparison between original and cloned node

Sr No	Test	Original Adv Count/min	Cloned Adv Count/min	Cloned Adv/min Original Adv/min	Percentage %
1	Test1	233	2629	11.28	1100
2	Test2	245	2559	10.44	1044
3	Test3	230	2630	11.43	1143
4	Test4	250	2641	10.56	1056
5	Test5	225	2612	11.61	1161
	Average	236.6	2614.2	11.07	1107

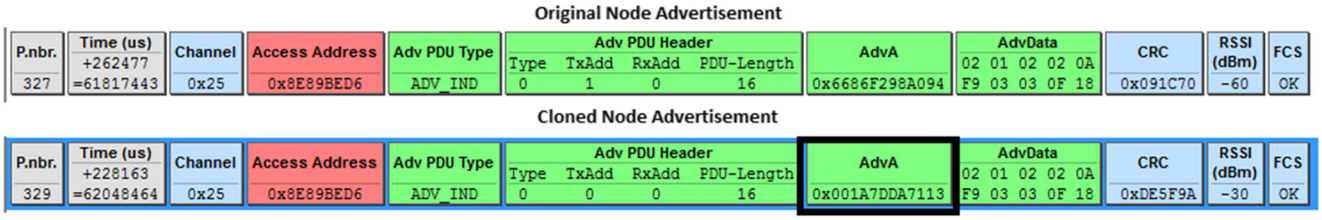

Fig. 11 Original vs Cloned node advertisement address comparison. The figure shows difference between Bluetooth addresses of original node and cloned node. This is achieved by whitelisting legitimate Bluetooth addresses

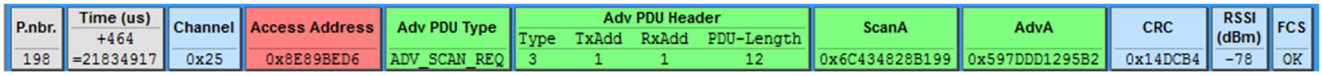

Fig. 12 Sample of a scan request packet. The figure shows malicious scan request from a Bluetooth address that is not whitelisted

however, it can be utilized in eHealthcare scenario where sensors must advertise only to a specified sink node/smartphone.

Bluetooth address whitelisting

As mentioned in multiple metrics in section 4.3 and 4.4Bluetooth address whitelisting is another important mitigation technique. As also described in [7], maintaining a complete inventory of Bluetooth addresses and connecting to only trusted devices is recommended, which leads to whitelisting of legitimate devices in a network. This mitigation technique will assist in detecting cloned node addresses and malicious scan requests. Although, Bluetooth address spoofing is possible, but it can be mitigated with addresses whitelisting and using Bluetooth version 4.2, in which privacy enable devices advertise using random Bluetooth address.

Analyzing advertisement interval

Attackers use the rushing technique to reach sink node/smartphone before the original node. They increase advertisement frequency of advertised packets and immediately connect to original node so that it stops advertising. To mitigate this attack, counters can be used to measure advertised packets frequency (measure advertisement interval between two received advertisements). Based on specific threshold, legitimate and cloned nodes can be differentiated. This can be implemented in a sink node/smartphone application. Most of the solutions discussed here can be implemented in software or at application level and no additional hardware changes are required.

Threshold on RSSI level

As mentioned in section 6.1 RSSI levels can be very helpful in differentiating between cloned and legitimate BLE nodes. Based on the environment and movement of sensors, a specific threshold value can be set to differentiate between the two nodes. If RSSI received by sink node/smartphone is below a threshold value then received advertisement can be tagged as cloned device advertisements, and thus, can be blacklisted.

Conclusion and future work

We overviewed Bluetooth low energy, its characteristics, classes, devices architecture, security features, pairing methods, and its multiple roles at different layers. Multiple attack methods were also discussed and reason for selecting MITM was also explained. Based on the literature review, which was completed specifically for BLE security it was highlighted that BLE is prone to MITM attacks. Secure data acquisition and transmission protection was one of our main objectives. Detecting and mitigating MITM attacks can secure eHealthcare system at sensors end and can stop numerous attacks at this layer.

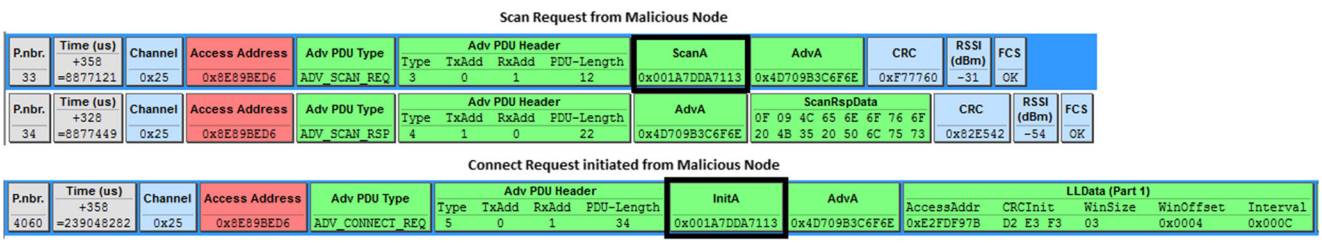

Fig. 13 Scan and connect request initiated from cloned node

Implementation section covered the testbed components that were used for carrying out the experiments, their specifications were also discussed. Scenario setup was explained and MITM tool Gattacker was explained afterwards. Results and analysis section covered MITM detection metrics and their mitigation techniques including RSSI levels, advertisement intervals, cloned node address and malicious scan requests.

Based on multiple simulation results, this research highlighted novel security checks and controls, which are recommended for inclusion in Bluetooth low energy to detect and thwart MITM attacks. As the literature suggests, security is not a prime consideration when developing a solution, thus, offering doorways for serious security threats against patient's privacy and safety. The focus of this research was on strengthening the security of BLE-based sensor's data acquisition and transmission in eHealth sector for countering against the stated threats. A critical analysis of BLE protocol exposed several vulnerabilities, fostering MITM attacks in devices having NiNo capabilities, which is the case in medical sensors. Multiple solutions proposed in the past lacked implementation/simulation and demanded multiple requirements at hardware and software level. Research presented in this paper proposed multiple novel detection and mitigation techniques/metrics without the need of any hardware or software changes. We performed multiple simulation tests to detect and stop MITM attacks, thus strengthening the security of BLE-driven medical sensors.

Multiple MITM detection and mitigation metrics were proposed and proof of concept was provided in the research presented in this article. However, complete implementation of mitigation techniques at application level can be sought as a future work. Our solution does not require any specific changes at hardware or software level, so implementation of the presented solution is rather easy. Addition of multiple metrics to detect these types of attacks and a working solution based on this framework may also be considered as a future work. Analyzing of overhead caused by implementing these techniques is also included in our future work. We propose that these mitigation techniques or changes may be incorporated in Bluetooth standard as a mandatory requirement to stop MITM attacks in just works pairing method of devices having NiNo capabilities. Mutual authentication of both devices and brute force restrictions are also important in realizing an overall secure Bluetooth protocol. Unique device fingerprinting is also an important way forward to stop device cloning and MITM as a result.

Acknowledgements The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group no (RG-1439-022).

References

1. Bello, O., Zeadally, S., Toward efficient smartification of the Internet of Things (IoT) services. *Futur. Gener. Comput. Syst.*, 2017.
2. Yaseen, M., Saleem, K., Orgun, M. A., Abbas, H., Al-Muhtadi, J., Iqbal, W., Rashid, I., Secure sensors data acquisition and communication protection in eHealthcare: Review on the state of the art. *Telematics Inform.*, 2017.
3. B.S. Proprietary, Bluetooth Core Specification version 5.0, in, Bluetooth, 2016.
4. Agrawal, V. M., and Chauhan, H., An overview of security issues in mobile ad hoc networks. International Journal of Computer Engineering and Sciences 1:9–17, 2015.
5. Jo, M., Han, L., Tan, N. D., and In, H. P., A survey: Energy exhausting attacks in MAC protocols in WBANs. Telecommunication Systems 58:153–164, 2014.
6. Kang, J., Adibi, S., A review of security protocols in mhealth wireless body area networks (WBAN), in: Communications in Computer and Information Science, Springer Science + Business Media, pp. 61–83, 2015.
7. Padgette, J., Guide to bluetooth security. NIST Special Publication 800:121, 2017.
8. Thilakanathan, D., Chen, S., Nepal, S., Calvo, R., and Alem, L., A platform for secure monitoring and sharing of generic health data in the cloud. Future Generation Computer Systems 35:102–113, 2014.
9. Arney, T. O., A literature review on the current state of security and privacy of medical devices and sensors with bluetooth low energy, 2018.
10. Ray, A., Raj, V., Oriol, M., Monot, A., Obermeier, S., Bluetooth low energy devices security testing framework, in: 2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST), IEEE, pp. 384–393, 2018.
11. Albahar, M.A., Haataja, K., Toivanen, P., Bluetooth mitm vulnerabilities: A literature review, novel attack scenarios, novel countermeasures, and lessons learned. *Int. J. Inf. Technol. Secur.* 8, 2016.
12. Jasek, S., Gattacking Bluetooth smart devices, in: Black Hat USA Conference, 2016.
13. Ryan, M., Bluetooth: With low energy comes low security, in: WOOT, 2013.
14. Guo, Z., Harris, I. G., Jiang, Y., Tsaur, L.-F., An efficient approach to prevent Battery Exhaustion Attack on BLE-based mesh networks, in: Computing, Networking and Communications (ICNC), 2017 International Conference on, IEEE, pp. 1–5, 2017.
15. Saravanan, K., Vijayanand, L., Negesh, R., A novel bluetooth man-in-the-middle attack based on SSP using OOB association model, arXiv preprint arXiv:1203.4649, 2012.
16. Wang, P., Bluetooth low energy-privacy enhancement for advertisement, in, Institutt for telematikk, 2014.
17. ALMomani, I., Al-Saruri, M., and Al-Akhras, M., Secure public key exchange against man-in-the-middle attacks during secure simple pairing (ssp) in bluetooth. World Applied Sciences Journal 13: 769–780, 2011.
18. Haataja, K. M., Hypponen, K., Man-in-the-middle attacks on bluetooth: a comparative analysis, a novel attack, and countermeasures, in: Communications, Control and Signal Processing, 2008. ISCCSP 2008. 3rd International Symposium on, IEEE, pp. 1096–1102, 2008.
19. Moon, J., Jung, I. Y., and Yoo, J., Security enhancement of wireless sensor networks using signal intervals. Sensors 17:752, 2017.
20. Albahar, M. A., Haataja, K., Toivanen, P., Towards enhancing just works model in bluetooth pairing. *Int. J. Inf. Technol. Secur.* 8, 2016.

21. Zhang, Q., Liang, Z., and Cai, Z., Developing a new security framework for bluetooth low energy devices. CMC-Computers Materials & Continua 59:457–471, 2019.
22. Shen, J., Yuen, T. T., Choo, K.-K. R., Zeng, Q., AMOGAP: Defending against man-in-the-middle and offline guessing attacks on passwords, in: Australasian Conference on Information Security and Privacy, Springer, pp. 514–532, 2019.
23. Jie, Y., Choo, K.-K. R., Li, M., Chen, L., Guo, C., Tradeoff gain and loss optimization against man-in-the-middle attacks based on game theoretic model. *Futur. Gener. Comput. Syst.*, 2019.
24. Haataja, K., Toivanen, P., Two practical man-in-the-middle attacks on bluetooth secure simple pairing and countermeasures. *IEEE Trans. Wirel. Commun.* 9, 2010.
25. Gajbhiye, S., Karmakar, S., Sharma, M., and Sharma, S., Bluetooth secure simple pairing with enhanced security level. Journal of information security and applications 44:170–183, 2019.
26. Gajbhiye, S., Karmakar, S., Sharma, M., and Sharma, S., Two-party secure connection in Bluetooth-enabled devices. Information Security Journal: A Global Perspective 27:42–56, 2018.
27. Sun, D.-Z., Mu, Y., and Susilo, W., Man-in-the-middle attacks on secure simple pairing in Bluetooth standard V5. 0 and its countermeasure. Personal and Ubiquitous Computing 22:55–67, 2018.
28. Hassan, S. S., Bibon, S. D., Hossain, M. S., Atiquzzaman, M., Security threats in bluetooth technology. *Comput. Secur.*, 2017.