The objectives

A. Present an overview of Bluetooth technology with a focus on its security, vulnerabilities, threats, and risk mitigation solutions.
B. Provide real-life examples of recent Bluetooth exploits.
C. Discuss several recommended measures to secure Bluetooth communication.

2. Related Work

As Bluetooth technology continues to evolve and improve, security experts continue their research with the goal of obtaining information on the newest version of the technology to gain insight, enhance security, and develop updated versions.

In "Bluetooth Security Threats and Solutions: A Survey", the authors presented reports of Bluetooth threats since the technology's inception up until 2007 [8]. There have been additional attacks recorded, as well as an increase in the range of devices that have been targeted. This comes as a result of the increase in IoT devices in the market. This is further discussed in our work.

In "A Survey on Security Threats and Vulnerability attacks on Bluetooth Communication", the authors discuss different Bluetooth vulnerabilities [9]. These vulnerabilities are further discussed, and a Bluetooth attack taxonomy is introduced in our work.

In "Security Threats in Bluetooth Technology", the authors provide an overview of Bluetooth technology, including its background and architecture, as well as different types of attacks and prevention methods [10]. Our work introduces Bluetooth 4, addresses Bluetooth's relationship to IoT, and provides real-life examples of exploits.

In "Bluetooth Low Energy Mesh Networks: A Survey", the authors provide an in-depth overview of BLE Mesh Networks and briefly touch on security in IoT networks [11]. Our work further expands on this by discussing specific Bluetooth vulnerabilities and threats.

In "A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends", the authors provide information on wireless security and physical layer-based attacks [12]. Our work focuses in on Bluetooth technology and provides real-life examples of Bluetooth exploits.

In addition to the works cited above, more research on Bluetooth technology has been done and several other papers have been published that exceed the scope of our work. Our paper provides updated information on types of Bluetooth attacks, as well as discusses real-life exploits.

3. Overview of Bluetooth Technology

In this section, we provide an overview of Bluetooth technology, which is used for short-range wireless communication. The technology enables data communication in computers, mobile phones, medical devices, and many other wireless devices. Below, we discuss Bluetooth frequency and connectivity ranges, including the three device classes of connectivity, the potential for interference, and how Bluetooth prevents such interferences; the Bluetooth piconet, which describes and illustrates device connectivity formation; and the Bluetooth protocol stack for Bluetooth 1, 2, and 3 and another for Bluetooth 4, which describe and illustrate Bluetooth protocol architecture. We also describe potential attacks and security failures in the different layers.

3.1. Bluetooth Frequency and Connectivity Ranges

Bluetooth enables, low power communication between devices that are in close proximity of each other. It operates in the unlicensed Industrial, Scientific, and Medical (ISM) radio frequency (RF) 2.4 GHz spectrum and has a range from between 0.5–1 m to 100 m [2].

The technology has three classes of devices, which offer three different connectivity ranges. Class 1 devices offer a range of 100 m and transmit at 100 mW [2]. Class 2 devices, the most common devices, have a range of 10 m and transmit at 2.5 mW [2]. Class 3 devices have a range of about 1 m and transmit at 1 mW [2].

One main advantage of the technology is its ability to transmit both voice and data simultaneously; however, the 2.4 GHz radio frequency spectrum is shared with many consumer appliances (i.e., microwaves, baby monitors, and cordless phones), which could cause interference [2,13]. According to research, coexistence in wireless technologies resulted in Bluetooth signals being harmed significantly by other wireless technologies [13]. For this reason, Bluetooth technology uses hops in the Bluetooth frequency at 1600 hops per second and a technology called spread spectrum to avoid interference.

3.2. Bluetooth Piconet

Bluetooth technology can be used to connect almost any two Bluetooth-enabled devices. The devices can communicate by the formation of a Bluetooth network, known as a piconet. A piconet is a spontaneous, ad hoc network that enables two or more Bluetooth devices to communicate with one another [2,14]. In the network, one device is designated as the master, while all other devices are designated as slaves [2]. There can only be one master device, which serves as the controlling device in the piconet. There can be up to seven active slave devices. These devices are able to request and transmit data to the master device. The connection between a cell phone (master) and a smartwatch (slave) is an example of a simple Bluetooth piconet. Figure 1 below provides an illustration of a sample Bluetooth piconet.

Figure 1. Bluetooth Piconet.

3.3. Bluetooth Protocol Stack

Figure 2 below illustrates the Bluetooth protocol stack for Bluetooth versions 1, 2, and 3. The stack consists of a variety of Bluetooth protocols, including the Logical Link Control Adaptation Protocol (L2CAP), Link Management Protocol, Radio Frequency Communications (RFCOMM) protocol, and the Service Discovery Protocol (SDP) [15,16]. While all of the protocols in the stack are illustrated below, there are cases when only a small vertical piece of the stack is needed.

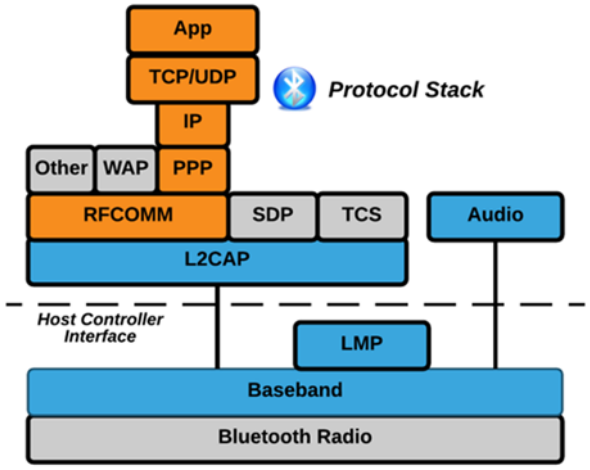

Figure 2. Bluetooth Protocol Stack (Bluetooth 1, 2, and 3).

A Host Controller Interface (HCI) which is also seen in the illustration (Figure 2) above, is the command interface, which incorporates a baseband controller and link manager. This interface enables both hardware access and register control.

Figure 3 below illustrates the protocol stack for Bluetooth 4 [17]. The stack consists of three layers, the Controller Layer, the Host Layer, and the App Layer [17]. Each of the layers incorporate different protocols. The Controller Layer incorporates the Physical Layer, Direct Test Mode, Link Layer, and Host Controller Interface [17]. The Host Layer incorporates the Logical Link Control and

Adaptation Protocol, Attribute Protocol, Security Manager, Generic Attribute Profile, and Generic Access Profile [17]. The App Layer incorporates Applications [17].

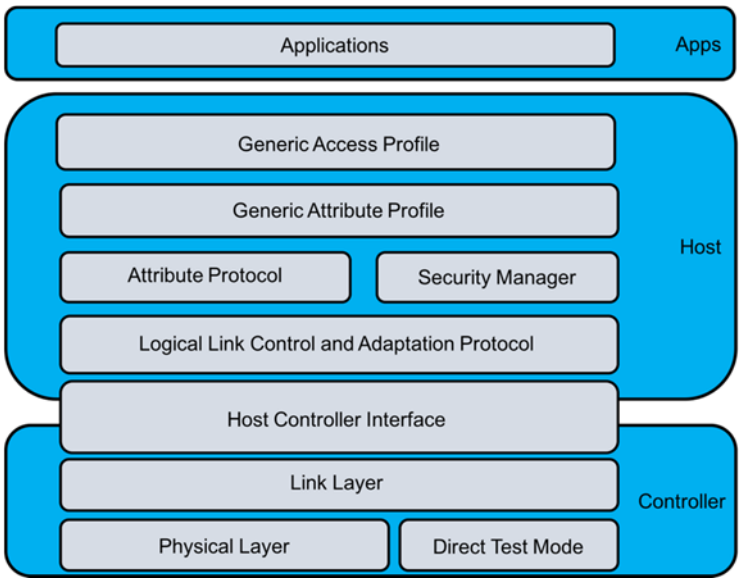

Figure 3. Bluetooth Protocol Stack (Bluetooth 4).

4. Bluetooth Security

In this section, we discuss Bluetooth security. There are two guides and standards for Bluetooth protocols and security, NIST 800-121-R1 and IEEE 802.15.1. NIST 800-121-R1 details the recommended Bluetooth security processes. These recommendations include the authentication and verification of the sender, confidentiality regarding information, and authorization in regard to who has control over access to the information. IEEE 802.15.1 is the standard for Bluetooth Wireless Technology. It discusses Bluetooth security in addition to the protocols surrounding Bluetooth technology.

A. Bluetooth Security Modes

All Bluetooth devices operate in 1 of 4 defined access security modes: Security Mode 1 (non-secure); Security Mode 2 (service level enforced security); Security Mode 3 (link level enforced security); and Security Mode 4 (service level enforced security with encrypted key exchange).

The Security Mode determines available service security levels. Security Modes 1 and 3 do not specify service security levels. Security Mode 2 can enforce any combination of the following basic security services: authentication, confidentiality, and authorization. Security Mode 4 specifies five levels of service security [3]. In this mode SHA-256 is used for hashing and AES CCM is used for encryption. It also uses Secure Simple Pairing (SSP) for key generation. Mode 4 is listed as the mandatory mode for Bluetooth versions 2.1 + EDR and newer versions [18].

B. Bluetooth Trust Modes

In addition to the security modes discussed above, there are two levels of trust for Bluetooth devices, trusted and untrusted. They are described as follows:

(1) Trusted—A trusted device has established a fixed relationship with another device and has unrestricted access to all services.
(2) Untrusted—An untrusted device only has access to a restricted set of services. Although the device has passed authentication successfully, it does not have a fixed relationship with another device.

C. Discoverability in Devices

Discoverability modes of Bluetooth devices also affect the device's security. Devices in discoverable mode are more vulnerable, as they can be recognized. The device name, class, list of services, and technical information are all exchanged in discoverable Bluetooth devices that are in range (approximately 10 m). In addition, every Bluetooth device has a unique 48-bit address used for identification, known as the BD_ADDR . This address is similar to a MAC address, which isa manufacturer assigned address for hardware that serves as a unique identification number . The BD_ADDR, like a MAC address, is assigned by the manufacturer .

D. Bluetooth Security Services

The first time that two devices attempt a connection, a trusted relationship needs to be established through authentication . Authentication is performed by using challenge-response, based on BD_ADDR and a link key . The link keys, once established, are kept by both devices to be usedfor future pairing . In older versions of Bluetooth (v2.0 and earlier), common secret PIN codes,which are passkeys required for first time Bluetooth connections, are used . The PINs are usedby both devices and consist of between 4 and 16 characters . These codes are specifically used forlink-key generation . This is illustrated in Figure 4 below [8,19]. In some cases, once the PIN is set,it cannot be changed . It is also important to note that two devices cannot communicate or be pairedif the devices have fixed PINs [20]. Newer versions of Bluetooth (v2.1 and later) use SSP for the pairingprocess, which utilizes public key cryptography instead of a PIN [2,19]. This protocol is illustrated inFigure 5 below [19].

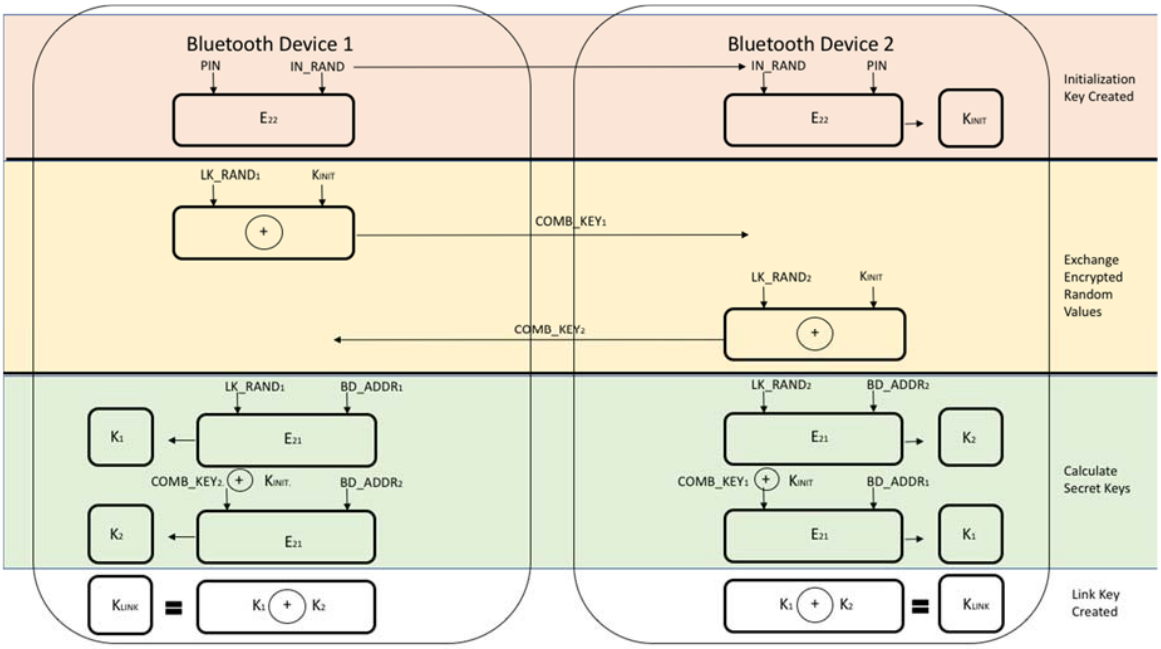

Figure 4. Link-Key Generation with PIN.

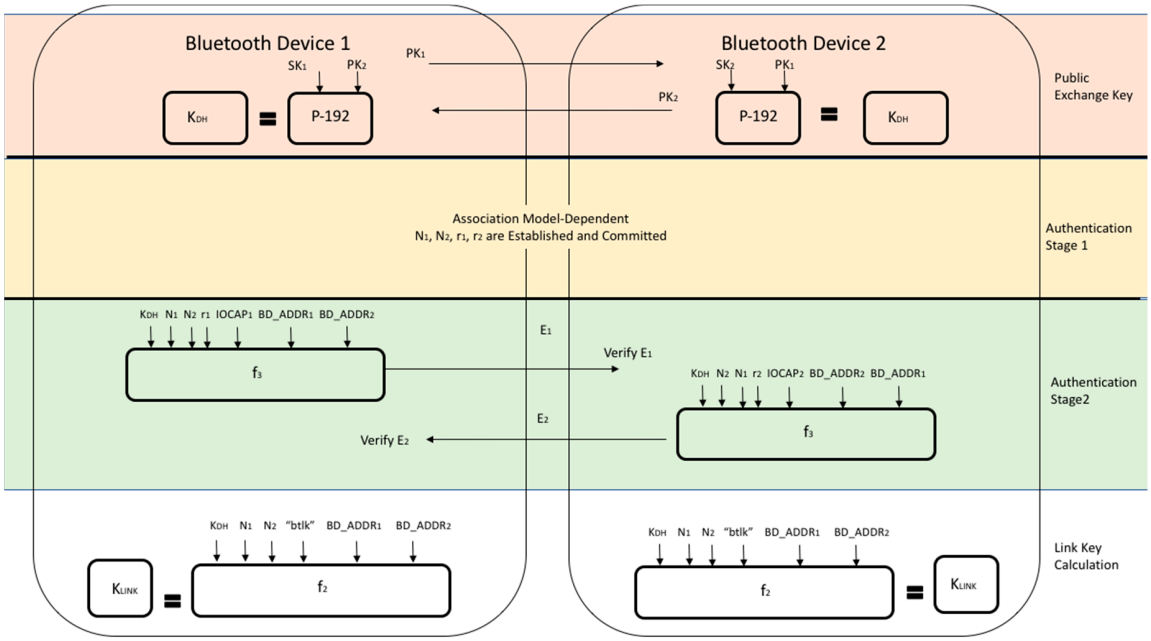

Figure 5. Link-Key Establishment for SSP.

Authorization begins by first determining whether the device had previously been authorized as a trusted device [2]. If the device database lists it as a trusted device, then access to services is granted [2]. If the device is not listed as a trusted device, trust must first be established before it can be authorized [2].

Confidentiality is achieved through encryption, more specifically the use of the E0 stream cypher [2]. A link key and the BD_ADDR of the device are used to develop a keystream that when combined with plaintext achieves a cyphered text [2,12]. Attacks and cryptanalysis attempts on E0 have proven that the stream cipher is vulnerable to attacks.

E. Built-in Security Features

Bluetooth technology has certain built-in features that help secure the technology. They include:

(1) Adaptive Frequency Hopping: Frequency hopping in Bluetooth uses a 2.4 GHz ISM band with 79 channels to enable hops at 1600 hops per second. During the hopping, existing frequencies are excluded. The ability to frequency hop reduces both jamming and interference.
(2) E0 Cipher Suite: The cipher generally has a key length of 128 bits and uses stream ciphering.
(3) Undiscoverability: This prevents devices from responding to scanning attempts. A device's 48-bit BD_ADDR address is also concealed.
(4) Pairing: Pairing enables devices to communicate. A device's BD_ADDR must be known for a pairing request to be made. The BD_ADDR, which is discussed in the previous two sections, is identified from knowledge of previous pairing or by scanning.

5. Bluetooth Vulnerabilities and Threats

In this section, we discuss the vulnerabilities in different versions of Bluetooth. We then discuss Bluetooth attacks, which are a result of vulnerabilities in the technology. Finally, we introduce the Bluetooth threat taxonomy and discuss common Bluetooth attacks.

5.1. Vulnerabilities in Bluetooth Versions

The version of Bluetooth that is being used, and the security of communications between devices, which is only as strong as the weakest link (i.e., the device with the oldest (weakest) version) is important when discussing Bluetooth vulnerabilities [2]. Since many older devices are still being used today, the vulnerabilities in the older versions of Bluetooth continue to be present [2].

(1) Versions before Bluetooth 1.2: Link keys, which are based on static unit keys, are used for pairing and can be reused [2]. If the key is retrieved, malicious devices can eavesdrop on the original devices, as well as spoof the original device and/or connected devices [2].

(2) Versions before Bluetooth 2.1 + EDR: Codes that consist of short PINs are permitted [2]. These PINs are easy for attackers to guess due to their short length [2]. These versions are lacking in PIN management, which is a desirable security capability at an enterprise level [2]. In addition, the keystreams in these early versions become vulnerable after being connected for 23.3 h [2]. This is the time period at which the keystream repeats [2]. This increases an adversary's ability to decrypt messages [2].

(3) Versions 2.1 and 3.0: If Security Mode 4 devices are connecting to devices that do not support Security Mode 4, earlier security modes are used in the connection [2]. For example, it is possible that Security Mode 1, which offers no security, will be used [2]. This rollback in security modes makes versions 2.1 and 3.0 more vulnerable to attacks [2]. In addition, SSP static keys are used in versions 2.1 and 3.0, which increases the device's vulnerability to Man-in-the-Middle attacks [2].

(4) Versions before Bluetooth 4.0: There is an unlimited number of authentication challenge requests, which enables adversaries to obtain information on many challenge responses [2]. This allows them to gain insight on secret link keys [2]. In addition, the stream cipher E0 function, which is used in early versions, is considered weak [2].

(5) All versions of Bluetooth: Adversaries can view and potentially modify link keys if they are stored improperly [2]. In addition, encryption key lengths may be small, which can make them vulnerable to attackers [2]. It is possible that encryption keys can be as small as 1 byte [2]. Regarding authentication, there is no user authentication [2]. The Bluetooth standard only includes device authentication [2]. It is important to note that a device can remain in discoverable/connectable mode for an indefinite period of time [2,3].

5.2. Bluetooth Taxonomy of Attacks

The Bluetooth threat Taxonomy illustrated below in Figure 6, outlines, and classifies Bluetooth-based threats [21]. This classification system can help determine the severity of threats, provides precautionary methods, and presents reactionary strategies [21]. Some threats may display characteristics of several classifications; however, they are classified based on their predominant characteristic [21].

Classification	Method	Threats
Obfuscation	Techniques are used to hide the attack and prevent detection.	HCIConfig (Device Name)
		HCIConfig / BTClass (Class of Device)
		Bdaddr (Device Address)
		SpoofTooph
Surveillance	Device monitoring is done to collect information.	HCITool (Device Discovery)
		Sdptool (Service Discovery)
		Redfang
		Blueprinting
		Bt Audit
		War-Nibbling
		Bluefish
		BNAP BNAP / BlueProPro
		BlueScanner
Range Extension	Range of connectivity is extended so attacks can be conducted at a distance.	BlueSniping / Bluetooone
Sniffing	Sniffer is used to intercept data by capturing network traffic.	Merlin / FT4USB (External Based)
		BlueSniff (Frequency Based)
		HCIDump (Host Based)
Man-In-The-Middle	Attackers trick devices into thinking they are paired, when in reality they are both connected to the attacker.	Bthidproxy
Unauthorized Direct Data Access	Data stored in cloud is directly accessed due to vulnerabilities.	Bluesnarf / Blooover
		BTCrack / Btpincrack
		Car Whisperer
		HeloMoto
		Bluebugger
		HID Attack
		Btaptap
Denial of Service	Services are disrupted, making a machine or network unavaiable to users.	BlueSmack / Tanya
		Blueper
		BlueJacking / BlueSpam / Smurf
		vCardBlaster
		Signal Jamming
		BlueSYN / Pingblender (Multi-Vector DoS)
		Battery Exhaustion
Malware	Intrusive or harmful software is put on a computer to disrupt operations, steal data, or extort a target for ransom.	BlueBag
		Caribe
		CommWarrior
		Skuller
Fuzzer	Injects data into a stack or program and has the ability to detect bugs.	Bluetooth Stack Smasher / BluePAss
		BlueStab
		HCIDump Crash
		L2CAP Header Overflow
		Nokia N70 L2CAP DoS
		Sonyericson Reset Display

Figure 6. Bluetooth Threat Taxonomy.

5.3. Common Bluetooth Attacks

The pairing process is a main contributor to security issues found in Bluetooth [2]. Attacks can be performed during different stages of the pairing process including before the pairing process has completed and after devices are paired [2]. For example, attackers may be able to carry out Man-in-the-Middle attacks based on information they collected after pairing [2]. Some of the more common attacks on Bluetooth are described below:

A. MAC Spoofing Attack

The attack is performed before encryption is established and during the formation of the piconet when link keys are being generated [8]. Devices are able to authenticate each other by generating link-keys [3,22]. During the attack, attackers can imitate another user [8]. They also have the ability

to terminate connections or intercept/modify data with the use of special tools [8]. Figure 7 below illustrates a MAC Spoofing attack [23].

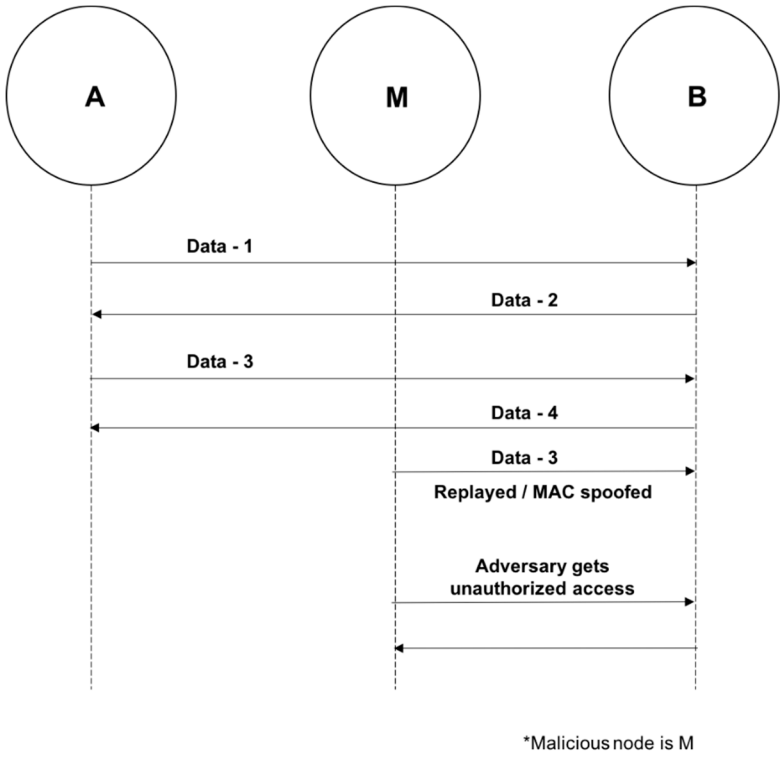

Figure 7. MAC Spoofing Attack.

B. PIN Cracking Attack

The attack occurs during the device pairing and authentication process. An attacker uses a frequency sniffer tool to collect the RAND and the BD_ADDR of the targeted device. A brute-force algorithm (E22 algorithm) is then used to test all possible permutations of the PIN with the data previously collected until the correct PIN is found [8]. Figure 8 below illustrates the PIN cracking attack structure [24].

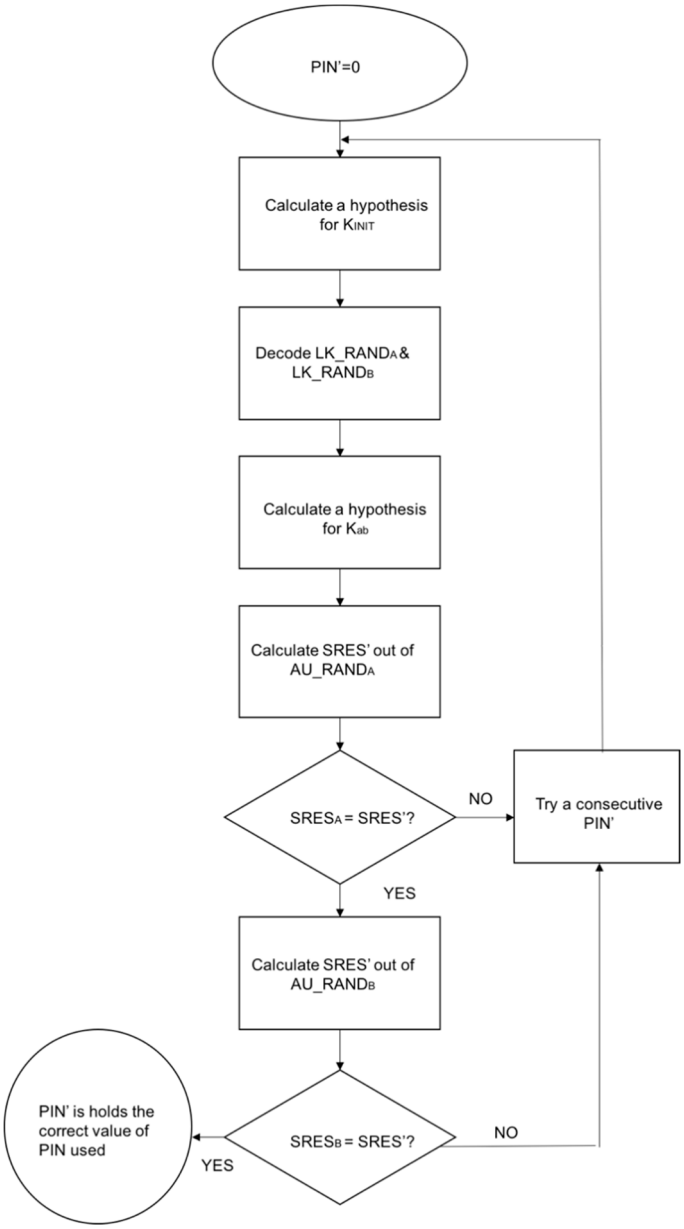

Figure 8. PIN Cracking Attack.

C. Man-in-the-Middle Attack

Man-in-the-Middle Attacks such as the one illustrated in Figure 9 below occur when devices are attempting to pair [25]. During the attack, messages are relayed unknowingly between the devices [9]. This enables authentication without the shared secret keys [9]. In a successful attack, the user believes the pairing was successful; however, this is not the case, as the two devices are paired to the attacker [8,9].

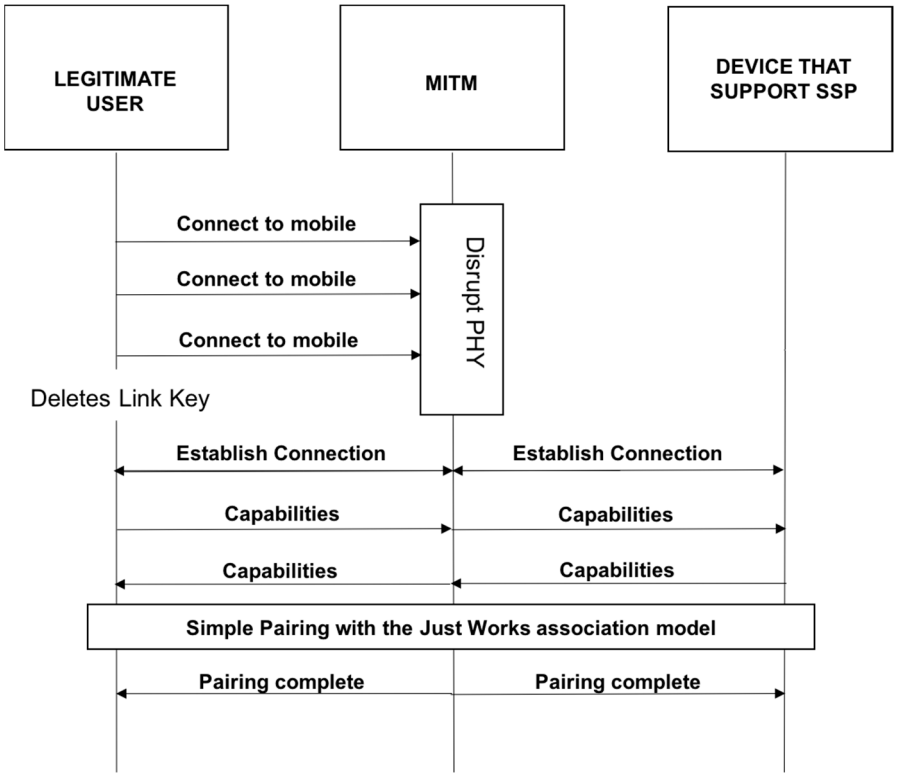

Figure 9. Man-in-the-Middle Attack.

D. BlueJacking Attack

During a BlueJacking attack, the attacker sends unsolicited messages to a device to trick the user into using an access code [8]. This enables the adversary to access files on the targeted device. The devices involved in the attack and the exact source of the message received need to be within a specific range, 10 m, for the attack to be successful [8]. This attack is commonly used in crowded areas (e.g., airports, shopping malls, and train stations) [8]. While it does not usually involve the alteration of data, it could make devices susceptible to other attacks [8].

E. BlueSnarfing Attack

The attack involves hacking into a mobile phone and stealing any of the data stored in the phone's memory, such as contacts, calendar entries, images, etc. [8]. During the attack, the attacker connects by exploiting the OBEX File Transfer Protocol, a file transfer program used in Bluetooth [26]. This enables the attacker to pair with the user's device. Figure 10 below illustrates a BlueSnarfing attack [27].

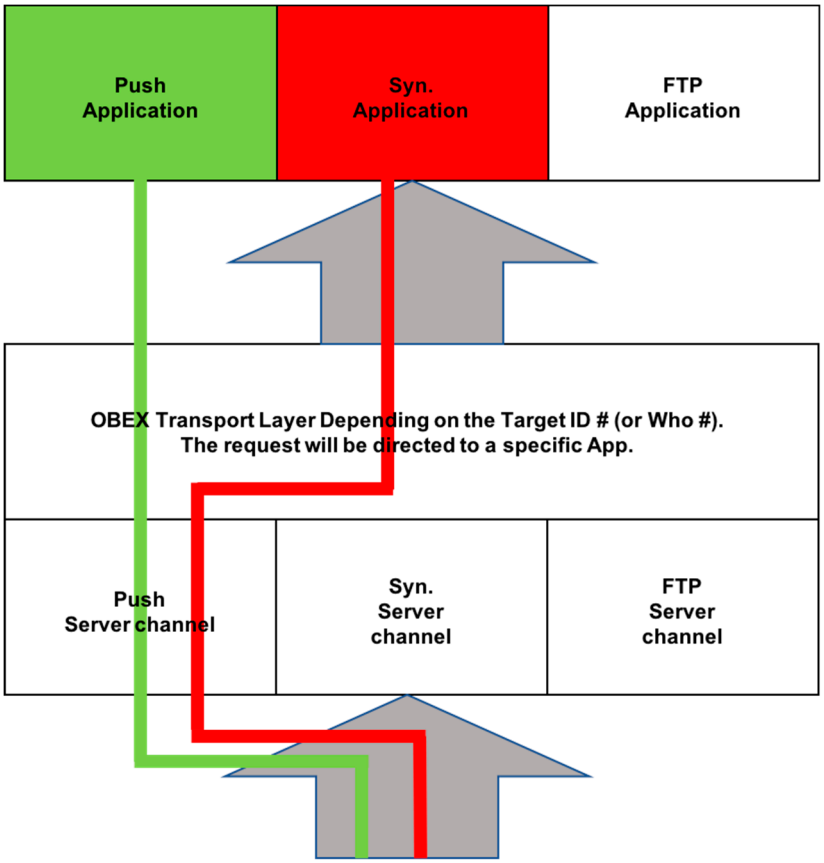

Figure 10. BlueSnarfing Attack.

F. BlueBugging Attack

The attack occurs in the RFCOMM protocol [28]. Physical connections are made via L2CAP + base band and it emulates serial RS-232 connections [28]. During the attack, the attacker connects to the target device without the knowledge of the owner [8]. The attacker then takes over the device by gaining access to the device's set of "AT" commands, which are attention commands that send instructions to the module [8]. This enables the attacker to execute commands as if he was the device owner [8]. The attacker can also steal information and access the phone's services and settings [8].

G. BlueBump Attack

The attack occurs when there is a weakness in the handling of link keys [29,30]. During the attack a business card is sent between the attacker and user [30]. By forcing the user to accept the card, a trusted and authenticated connection is made [30]. After the pairing, the user then has the ability to delete the link key; however, the user unknowingly still has an active connection to the attacker [30]. The attacker is then able to pair with the device later without authenticating by simply requesting link-key regenerations [30]. The attacker can continue to pair with the target device if the key is not deleted [30].

H. BlueDump Attack

During the attack, the attacker spoofs the BD_ADDR of one of the devices to connect with the other [31]. During pairing, the target device sends an authentication request [31]. The attacker responds with 'HCI_Link_Key_Request_Negative_Reply [31]. This is a result of the attacker not having a link key. In some cases, the device targeted deletes its link key [31]. It then goes into pairing mode [31].

I. BluePrinting Attack

The attack is carried out by combining the information that is revealed about a device to gain additional information, such as the manufacturer, device model, and firmware version [8,32]. This attack can only be performed when the BD_ADDR of the device is known [8].

J. Blueover Attack

Blueover and its successor Blueover II are auditing tools that are used to determine if a Bluetooth device is vulnerable, but they can also be used to initiate a BlueBugging attack [8].

K. BlueBorne Attack

The attack is conducted by exploiting a stack buffer overflow flaw [33]. By targeting the processing of pending client L2CAP configuration responses, the attacker is able to hijack Bluetooth connections [33]. This enables them to control a targeted device's embedded content and functions. For the attack to be successful, only MAC and Bluetooth addresses are needed [34].

L. Fuzzing Attack

The attack occurs when the adversary attempts to cause a device to behave abnormally by sending malformed data packets and non-standard data to a device's Bluetooth radio [3,35]. The attacker then watches how the device reacts to the data packets being sent [3]. If the device's operations become sluggish or stop during these attacks, the attacker could infer that the protocol stack has vulnerabilities [3].

M. Off-Line PIN Recovery Attack

During the attack, the attacker tries to intercept the IN_RAND value, LK_RAND values, AU_RAND value, and SRES (signed response) value [9]. The SRES value is a matching variable needed for authentication [9]. The attacker then uses brute-force to obtain a PIN that could be used to determine the correct SRES value, which is equal to the intercepted SRES value [9].

N. Brute-Force BD_ADDR Attack

This attack is a scanning attack on the last three bytes of the BD_ADDR of a device [8]. It is important to note that the first three bytes, which are known publicly, can be set as fixed [8,9].

O. Reflection/Relay Attack

The attack occurs when the adversary impersonates a device [8]. The attacker is not looking for any undisclosed information during the attack [9]. It simply authenticates the connection by reflecting/relaying device information [8].

P. Backdoor Attack

The attack occurs when establishing a trusted relationship during pairing [8]. During the attack, the adversary does not appear in the register of paired devices on the target device [8]. After a relationship is established, the attacker has access to the devices services and resources [8]. This access is unbeknownst to the device owner [8]. The BD_ADDR of the target device needs to be known for a backdoor attack to be successful [9]. It also needs to be determined that the device targeted by the attacker is vulnerable to the attack [8].

Q. Denial of Service Attacks

DDoS (Distributed Denial of Service) and ordinary DoS are two types of Denial of Service attacks [36]. For ordinary DoS attacks, the attacker tries to crash the network or restart the system by sending packets to the targeted system [36]. DDoS attacks can be done by a single attacker [36]. These attacks can disable a network [36]. They also can restrict a networks accessibility to a larger network [36]. The attacks target the Physical Layer in the protocol stack or those above the Physical

Layer. Some typical Denial of Service (DoS) attacks are BD_ADDR duplication, BlueSmack, BlueChop, L2CAP guaranteed service, battery exhaustion, and Big NAK (Negative Acknowledgement), which is an attack using a continuous retransmission loop [37].

R. Worm Attacks

The attacks occur when a malicious software or Trojan file sends itself to available Bluetooth devices. Examples of these attacks are:

(1) Cabir Worm: A malicious software that targets Bluetooth technology. Mobile phones that use the Symbian series 60 interface platform are vulnerable to the attacks [8]. For the attacks to be successful, the user must accept the worm [8]. This causes the malware to install on the device [8]. The worms are usually disguised in applications, which results in users unknowingly accepting them [8]. Once installed, the software is the able to use the compromised device to search for and send itself to other available devices [38]. The Mabir worm is a form of the Cabir worm [8]. This worm replicates by using Multimedia Messaging Service messages and Bluetooth [8].
(2) Skulls Worm: The Skulls Worm, a malicious SIS (Symbian Installation System) trojan file, targets Symbian mobile phones with the Series 60 platform [8]. The worm poses as a Macromedia Flash player [8]. The user must open and install the SIS file for the worm to become active [8]. It then searches for additional devices to infect and the process repeats itself [8,38].
(3) Lasco Worm: The Lasco worm, is a combination of a Bluetooth worm and SIS file [8]. It targets and infects Symbian mobile phones that support the Series 60 platform [8]. The user must open and install the velasco.sis file [8]. This prompts the activation of the worm [8]. It can then begin searching for additional devices to infect and the process repeats itself [8,38].

S. Bluesmack Attack

The attack is a DoS attack on Bluetooth devices and is similar to the "Ping of Death" attacks that are carried out on IP-based devices, which are networked devices [39]. It is done by sending pings that are approximately 600 bytes, as well as L2CAP echo requests to Bluetooth devices [19]. This results in the input buffer to overflow and the targeted device to be knocked out.

T. MultiBlue Attack

The attack occurs when an attacker has access to the device they wish to hack [40]. The MultiBlue dongle, a bluetooth capable 4 GB thumb, is used to gain access to and take over the targeted device [40]. The attacker then uses the MultiBlue application to see all discoverable devices within range and send pairing requests [40]. The targeted device then presents a code, a pre-shared key, which needs to be entered into the MultiBlue application [40]. This key is needed for authentication and encryption [40]. The attacker then has control of the device [40].

U. HeloMoto Attack

The attack is a mix of the previously mentioned BlueSnarfing and BlueBugging attacks [28]. A vulnerability caused by the erroneous implementation of "trusted devices" on Motorola devices is exploited during the attack [28]. When exploited, the adversary's device is stored as a trusted device on the target device's trusted list [28]. After connecting to the OBEX push profile, the attacker tries sending a vcard [28]. The adversary is then able to elude authentication and connect to the target device's headset profile [41]. BlueBugging is also used to take control of the device [41].

V. Bluecasing/War Nibbling Attack

The attack occurs when a phreaker, a telephone network hacker, uses laptops or PCs with high gain antennas and special software to discover and exploit vulnerabilities in Bluetooth phones to obtain access [35].

The ability to identify and describe these different attacks enable us to better understand the threats surrounding IoT Bluetooth devices. In addition, by classifying the threats, we can determine the severity of the various Bluetooth attacks. Next, we discuss ways to mitigate the risks and identify potential countermeasures used in response to these threats.

6. Bluetooth Risk Mitigation and Countermeasures

The attacks that are described in the previous section discuss Bluetooth flaws that result in vulnerabilities, which can be exploited by an attacker to steal data, send messages, make phone calls, and connect to the Internet using the compromised device [2]. We now discuss risk mitigation techniques and countermeasures.

6.1. Mitigation Techniques

Mitigating Bluetooth vulnerabilities differ significantly from mitigating vulnerabilities in a computer system. While application software patches are used to resolve vulnerabilities in computer systems, Bluetooth devices require upgrades in device firmware [2]. These upgrades cannot be developed by the general public and/or user community [2]. Therefore, Bluetooth devices will continue to be vulnerable to attacks even if mitigation solutions become available [2,29].

While all attacks cannot be prevented, and security is not guaranteed, there are countermeasures that can be used to secure Bluetooth communications [2]. Some of those mitigation techniques are described below:

A. Enhancement of Bluetooth user awareness: it is necessary to educate Bluetooth users to ensure they have knowledge of the proper Bluetooth security practices [2]. These security practices include:

(1) Default settings should be updated to achieve optimal standards [2].
(2) Ensuring devices are in and remain in a secure range. This is done by setting devices to the lowest power level [2].
(3) Using long and random PIN codes, which make the codes less susceptible to brute-force attacks [2].
(4) Changing the default PIN for devices and frequently updating this PIN (i.e., once every other month).
(5) Setting devices to undiscoverable mode by default, except as needed for pairing [2]. Most active discovery tools require that devices be in discoverable mode to be identified. Devices set to undiscoverable mode will not be visible to other Bluetooth devices. Devices previously configured, better known as trusted devices, will be able to connect and communicate while in this hidden mode.
(6) Turning off a device's Bluetooth when not needed or in use, especially while in certain public areas such as shopping malls, coffee shops, public transportation, clubs, bars, etc [2]. This can prevent users from receiving advertisements from other Bluejackers.
(7) Refraining from entering passkeys or PINs when unexpectedly prompted to do so.
(8) Frequently updating software and drivers to have the most recent product improvements and security fixes.
(9) It is recommended that users refrain from using non-supported or not secure Bluetooth-enabled devices or modules. This includes Bluetooth versions 1.0 and 1.2.
(10) Pairing devices as needed [2]. Users need to maintain that any pairing should take place in a secure non-public setting [2]. This will help prevent attackers from intercepting pairing messages [2]. As previously mentioned, a crucial part of Bluetooth security is pairing, so users should have knowledge regarding eavesdropping [2].

(11) Users should use SSP instead of legacy PIN authentication for the pairing exchange process when it is possible. This will help mitigate PIN cracking attacks.
(12) All lost or stolen Bluetooth devices should be unpaired from devices they had previously been paired with [2]. Unpairing will prevent an attacker from accessing the users other devices through the Bluetooth pairing [2].
(13) Users should never accept transmissions from unknown or suspicious devices [2]. Content should only be accepted from trusted devices [2,8].
(14) All devices that are paired should be removed immediately after use.
(15) Devices should be monitored and kept at close range.

B. Instead of basing link keys on unit keys, they should be based on combination keys [2]. This will prevent Man-in-the-Middle attacks [2].
C. Use link encryption for all data transmissions to prevent any eavesdropping, including passive eavesdropping [2]. Use of the HID boot mode mechanism, a connectionless human interface device, should be avoided, as it sends traffic in plaintext.
D. Users should ensure all links are encryption-enabled when using multi-hop communication [2]. Failure to do so could result in the entire communication chain being compromised [2].
E. Require mutual authentication for network connected devices [2]. This will provide confirmation that the network connections are legitimate [2].
F. Lower the risk of broadcast interceptions by encrypting the broadcasts [2].
G. The maximum encryption key size should be used [2]. In addition, a minimum key size should also be set—128 bits is recommended [2]. The utilization of these minimum and maximum keys will protect devices from brute-force attacks [2].
H. To provide the highest level of security, Security Mode 3 is highly recommended [2]. This mode of security, which is implemented at the link level, is one of the highest levels of Bluetooth security [2].

6.2. Applications for Protecting Bluetooth Devices

(1) Bluetooth firewall: The Firewall application protects devices, specifically Android devices, from all Bluetooth related attacks [42]. Users are alerted upon any Bluetooth activity [42]. The application also enables you to see Bluetooth capabilities on devices or specific apps [42].
(2) Bluetooth file transfer: This application only enables authorized devices to be connected [40].

7. Commercial Product Examples

In this section, we discuss a few Bluetooth IoT commercial products and applications. We highlight examples of real-life exploitations, as well as explore how to apply the Bluetooth risk mitigations discussed in the previous section.

7.1. Bluetooth Automotive Hacks

Bluetooth comes standard in most automobiles and supports hands-free calling and music streaming from a user's smartphone. If the vulnerabilities in the technology are exploited, the driver, passengers, and other individuals on the road may be in danger. Below we describe two different automobile attacks, one using a smartphone and the other using Car Whisperer technology.

A. Automobile Hack Using a Smartphone

Researchers from the University of Washington were able to attack a car's Bluetooth system; the one which allows a driver to make hands-free cell phone calls [43]. A vulnerability in the system's implementation was discovered by researchers [43]. During the exploit, a trusted device was used or researchers were able to elude authentication to authorize a new connection [43]. They then called the vehicle and executed a malicious code to take control of the car [43,44]. This enabled the attackers to

send commands and override several of the vehicles controls [44]. The target of the attack was a 2009 mass-production sedan [43]. Researchers confirmed that they were able to gain full control of the car's internal computer systems [43]. This was a result of a vulnerability in the Link Manager Protocol/Link Layer Protocol.

B. Automobile Hack Using Car Whisperer

Car Whisperer software is used to trick automobile Bluetooth systems into connecting with a Linux computer. European wireless security experts, the Trifinite Group, developed the technology in order to demonstrate the limitations of Bluetooth systems [45]. Their specific focus was on the vulnerabilities resulting from the use of standard passkeys [45]. The Car Whisperer software is used to exploit the very short four-digit security key assigned by car manufacturers [45]. It is important to note that the same code is often used for the keys (i.e., 1234 or 0000) [45]. This key grants access to the system [45].

During the hack, continuous device inquiries for visible Bluetooth devices are conducted by running cw_scanner script [45]. The attackers are specifically looking for devices that are not only visible, but share the same device class [45]. Once these devices are identified, carwhisperer binary is executed by the cw_scanner [45]. This connects the device and cw_pin.pl script provides the pass key required for connecting the devices [45]. A control connection is made to the SCO links, synchronous connection-oriented radio links used to connect a slave and a master in a piconet to transmit voice and data [45,46]. Audio can then be sent to or recorded from the targeted device by using the carwhisperer binary [45]. Attackers also have the ability to inject audio into the car and/or eavesdrop [45].

The Trifinite Group was able to conduct an attack using the software [47]. By using a special directional antenna, they were able to extend the Bluetooth connection to about a mile [47]. They then utilized the software to connect to the vehicle's Bluetooth [48]. Upon connecting, the researchers were able to listen in on and send audio to about ten cars [47]. The attack was conducted over a one-hour period [47].

These attacks exploit vulnerabilities in the Link Manager Protocol/Link Layer Protocol. To prevent the type of attacks discussed above, manufacturers need to build security in to the vehicle's firmware. In addition, drivers should be sure they are running the latest software updates on their vehicles.

7.2. Bluetooth Medical Hacks

Bluetooth technology has become the preferred communication method used in medical devices [49]. This wireless way of communication along with the device's corresponding apps are set to change the face of the healthcare industry. Some Bluetooth-enabled devices include blood glucose monitors, oximeters, asthma inhalers, as well as other devices that are used to diagnose injuries, administer medication, and securely transmit information from patients to providers [50]. The vulnerabilities in these devices present new life-threatening security challenges. Possible exploitations of a defibrillator and an insulin pump along with its connected app are illustrated below.

A. Bluetooth Hacks on Defibrillators

According to a recent report in WIRED, security experts who studied at the Midwestern medical facility chain, over the course of two years found critical security vulnerabilities in medical devices [51]. One finding allowed Bluetooth-enabled defibrillators to be manipulated to deliver random shocks to a patient's heart or prevent a medically needed shock from occurring [51].

Specific product brands and their vulnerabilities were not identified by the researchers; however, they announced that a wide variety of devices shared common security vulnerabilities, resulting in the lack of authentication needed to access or manipulate the equipment [51]. They also identified the use of weak passwords or default and hardcoded passwords such as "admin" or "1234", which were assigned by the device vendor [51]. These vulnerabilities are in the Link Manager Protocol/Link Layer

Protocol. To mitigate these vulnerabilities, the defibrillator's software and hardware would most likely need to be updated.

B. Bluetooth Hacks on Insulin Pumps and Companion App

Bluetooth technology has given diabetes patients a more efficient and effective way to manage their diabetes by providing them with the ability to easily monitor their blood glucose levels [52]. Glucose monitors can be connected to companion smart apps on smartphones, which not only capture data, but also send alerts to patients [52]. The technology can be hacked by individuals in close range and Man-in-the-Middle and eavesdropping attacks can be executed. During the attacks, data being communicated between the devices could be intercepted, decrypted, and captured [52]. This attack targets vulnerabilities in Bluetooth's Host Security protocols. To mitigate these vulnerabilities, the software and hardware of the glucose monitor would most likely need to be updated [52].

7.3. Bluetooth Smartwatch/Smart Bracelet Hacks

We are seeing a rapid growth of smartwatches and smart bracelets, including Fitbits and iWatches in the market. These devices use Bluetooth communication channels and mechanisms, which make them vulnerable to Bluetooth attacks.

Below we discuss Bluetooth hacks that were conducted to exploit vulnerabilities on a Samsung Gear Live watch using Bitdefender and a Dax-Hub Sw-28 smart bracelet using GATTacker.

A. Bitdefender Hack on Samsung Gear Live

A proof-of-concept hack was executed by experts at Bitdefender [53]. The attack targeted a Samsung Gear Live smartwatch that was paired with a Google Nexus 4 smartphone [53]. The smartphone was running the very secure, Android L (Preview Version) [53]. The use of sniffing tools enabled researchers to uncover the PIN used to protect the smartwatch and smartphone connection [53]. This showed that an attacker may be able to decode a user's data, including Google Hangout messages, as well as Facebook conversations [53].

The six digit PIN code that secures Bluetooth communication between most smartwatches and Android devices was exploited during the attack [53]. The attacker can easily perform a brute-force attack on the PIN, as the "key space" is composed of only 1 million possible key combinations [53].

The vulnerability is in the Link Manager Protocol/Link Layer Protocol and can be remediated by the manufacturer by requiring a password for Bluetooth pairing, as well as implementing encryption for the data communication.

B. Man-in-the-Middle Attack on Dax-Hub SW-28 Smart Bracelet

According to WIT (Wessex Institute of Technology) Press, a researcher at Tech Leader, AppSec Labs conducted research with the objective of conducting a Man-in-the-Middle attack on a Bluetooth smart mobile app and smart bracelet [54]. The smart bracelet selected for the attack was a Dax-Hub SW-28 smart bracelet, which is a bracelet that captures information on an individual's sports activities, as well as other health-related data. All data captured from the bracelet is synced with the PowerSensor app, which is the app designated for the smart bracelet [54].

The attack was carried out by running GATTacker central (ws-slave) on a virtual machine (VM) and running and configuring GATTacker peripheral to the IP address of the ws-slave [54]. A scan was conducted to detect the MAC address of the smart bracelet, which then enabled the researcher to discover advertisements and services that identified with the MAC address [54]. These advertisements and services were utilized to simulate a fake device [54]. The fake device then used GATTacker to advertise the target's name to get the victim to connect to the fake device [54]. In this case, when the mobile app scanned for devices, the fake advertisement was located, and the app and bracelet were connected through a Man-in-the-Middle attack [54]. If the devices were previously paired, the fake

device and mobile app would immediately connect [54]. This is provided that MAC spoofing was used to present the MAC address of the real smart device [54].

Once the app and smart bracelet were connected, the researcher was able to view all the communication data between the two devices [54]. In addition, each time a request was sent from the bracelet to the app, the researcher was able to modify the data [54]. The modification was done by adding a hook to the notification event identifier that was used to execute a code, which would update the data being communicated automatically [54]. GATTacker hooking technology was used to create the hook [54].

In the end, the hack enabled the researcher to modify and exaggerate the distance walked on a treadmill [54]. In addition, because the target smart bracelet had mobile camera and music capabilities, the researcher was able to exploit these capabilities as well by using BtleJuice's replay attack [54]. By replaying previous requests between the bracelet and mobile app, the attacker was able to use his computer to take pictures and play music on the victim's mobile device [54].

This type of hack exploits vulnerabilities in the Link Manager Protocol/Link Layer Protocol. It is preventable by implementing updated and additional security controls [54]. These controls include data encryption and signature, as well as a strong authorization and authentication processes [54].

7.4. Bluetooth Smartphone Hacks

According to Pew Research Center's survey in 2011, 77% of Americans own a smartphone [55]. Most of these devices are Bluetooth-enabled, which makes them vulnerable to attacks. Below we discuss Bluetooth hacks that can target smartphones.

Researchers conducted an experiment with the goal of learning how many devices they could infect with viruses in a public place using Bluetooth.

For the experiment, the researchers visited public places with a suitcase consisting of a computer that was equip with a Bluetooth sniffing program [56]. While 10 m is the average range a mobile phone can communicate via Bluetooth, an antenna can be used to increase the attack range [56]. During the experiment, the researchers were able to detect more than 1400 devices in less than 23 h [56].

In a similar experiment, researchers attacked Bluetooth-enabled devices and obtained over 300 address books [56]. This attack was conducted from the 11th floor of a Las Vegas hotel on targets that were on the ground in a taxi stand [56]. It is important to note that similar attacks can be carried out on tablets that are Bluetooth-enabled.

These attacks exploit vulnerabilities in the Link Manager Protocol/Link Layer Protocol. To mitigate these types of attacks, users should disable their Bluetooth when it is not in use.

7.5. Bluetooth Smarthome Hacks

According to CNBC and HIS Markit, in 2016 there were 80 million smarthome devices delivered around the world [57]. This was a significant increase, 64%, from 2015 [57]. Many of these devices are Bluetooth-enabled, which makes them vulnerable to attacks. Below we discuss exploited vulnerabilities in Nest and Dropcam cameras, smartlocks, speakers, personal assistants, as well as in an interconnected smarthome.

A. Bluetooth Camera Hack

In 2017, a mobile security and IoT hacker, Jason Doyle, identified three different vulnerabilities in Nest Cam Indoor, Nest Cam Outdoor, Dropcam Pro, and Dropcam security cameras that caused the camera's feed to cut out [58]. Each of the vulnerabilities were found in a specific firmware, which was version 5.2.1 [58].

The first two vulnerabilities involve sending parameters to the camera via Bluetooth [58]. They consist of either Wi-Fi password parameters or Wi-Fi SSID parameters, also known as service set identifier parameters [58]. SSID is a 32 character ID used for wireless networks. When these vulnerabilities are exploited, they cause a buffer overflow and result in the camera crashing

and rebooting [59]. The third vulnerability that was found enables a camera to be completely disconnected [58]. This vulnerability is exploited by using Bluetooth to send new, non-existent Wi-Fi SSID parameters [58].

These L2CAP vulnerabilities were exploited by attackers within Bluetooth range of the cameras. Alphabet, Nest's parent company, noted it was working on a fix to mitigate the vulnerabilities [58]. Users should make sure they are updating their devices and running the latest updates to mitigate these types of vulnerabilities.

B. Smarthome Hack to Gain Physical Access

Yossi Atias, the General Manager of IoT Security at BullGuard demonstrated how vulnerable some smarthome devices are to cyberattacks, which are intentional attacks to exploit computer systems, networks, and/or enterprises that are technology dependent [60,61]. In his demonstration, he constructed a fictional, secure smarthome that consisted of devices normally found in today's homes, including a smart alarm, IP camera, smart lock, and Amazon Echo [60]. He then conducted a live hack on these IoT connected devices [60]. His ability to exploit the smarthome devices enabled him to gain physical entry into the fictional smarthome [60].

One way to prevent these attacks on smarthome devices is for manufacturers to build security in from the device's hardware up [60].

C. Bluetooth Hacks on Personal Assistants

BlueBorne attacks, which exploit vulnerabilities in the L2CAP, were conducted on the Amazon Echo and Google Home to demonstrate the vulnerabilities on these smarthome Bluetooth devices [62]. The Amazon Echo had a remote code execution vulnerability in the Linux Kernal and information disclosure flaw in the SDP server [62]. The specific vulnerability was dependent on the variant's operating system [62]. The vulnerability discovered in Google Home was in Android's Bluetooth stack and was identified as an information disclosure vulnerability [62]. If exposed, the Google Home vulnerability can cause DoS [62].

It is important to note that Bluetooth cannot be disabled on these devices which, before the patches and automatic updates, left them vulnerable to these attacks [62]. The type of attacks illustrated above could be performed by any attacker who is in range of the personal assistant devices [62]. Users should always ensure they are running the most recent update on these devices [62]. Amazon and Google released patches for these vulnerabilities [62].

D. Bluetooth Smartlock Hacks

According to Tom's Guide, researchers Anthony Rose and Ben Ramsey tested 16 Bluetooth smart locks [63]. Of the 16 locks, they were able to wirelessly hack and open 12 of them [63]. They noted that the locks used BLE; however, the issues were not specifically with the BLE protocol [63]. The vulnerabilities were due to the way manufacturers implemented the lock's Bluetooth data communication with the corresponding smartphone app [63].

The researchers found that there were several different types of vulnerabilities that could be exploited on the smartlocks. These vulnerabilities were in the Link Manager Protocol/Link Layer Protocol. In some cases, passwords were sent in plaintext giving anyone with Bluetooth sniffing capabilities the ability to capture the passwords [63]. In other cases, passwords were sent twice giving attackers the ability to change the intercepted passwords and lock out the legitimate user [63]. The researchers also found that some lock manufacturers encrypt the passwords during the Bluetooth transmission; however, the locks could be unlocked with passwords that were still encrypted [63]. They did not need to decrypt the passwords to open the lock [63]. They were also able to stage Man-in-the-Middle attacks between the lock and the connected app or put the lock into an error state which opened the lock by changing a byte in a proprietary encryption [63].

The researchers noted that the locks they were not able to hack used two-factor authentication, encryption, and did not have hardcoded passwords in the software [63]. These security measures can be implemented to prevent smarthome locks from being hacked.

E. Bluetooth Speaker Hacks

Bluetooth speakers are becoming increasingly popular. According to Smart Industry, Bluetooth speakers that connect to wireless networks can be exploited to gain unauthorized access to a network [64].

Vulnerabilities can be found in the Link Manager Protocol/Link Layer Protocol. Risk can be mitigated by either turning the Bluetooth off while the speaker is not in use or by the manufacturer updating the hardware.

7.6. Bluetooth Hacks on Children's Toys

Children's connected toys are on the rise. Some of these toys contain cameras, speakers, GPS capabilities, microphones, and data storage among many other things. The influx of these Wi-Fi and Bluetooth-connected toys coming to the market pose serious threats, specifically to children's safety. Below we discuss how vulnerabilities were exploited on a teddy bear along with other connected children's toys.

A. Bluetooth Hack on a Teddy Bear

An 11-year-old boy, Reuben Paul, attended a cybersecurity conference at the World Forum in The Hague and performed a live hack exploiting vulnerabilities in his cloud-connected teddy bear [65]. Vulnerabilities in the Link Manager Protocol/Link Layer Protocol enabled the attack.

The exploit was performed by connecting the bear to Wi-Fi and Bluetooth, which enables the bear to transmit and receive messages [65]. Reuben used a Raspberry Pi that was connected to his computer to scan for Bluetooth-connected devices [65]. Through the scan, he was able to download many device numbers, some of which belonged to devices of important executives attending the conference [65]. By utilizing one of the numbers and the Python programming language, Paul was able to turn on the bear, turn on the bear's lights, and even record a message from the audience [65].

Building security into the hardware of these toys and the issuing of patches could mitigate these types of Bluetooth vulnerabilities and exploits.

B. Additional Hacks on Children's Toys

In addition to the teddy bear discussed above, there have been additional children's connected toys identified that have Link Manager Protocol/Link Layer Protocol vulnerabilities. According to the Guardian, these toys include Furby Connect, i-Que Intelligent Robot, Toy-Fi Teddy, and CloudPets. Researchers determined that no passwords, pins, or any other authentication was needed to gain access to the toys [66]. The researchers reported the following findings: Furby Connect was able to connect with any device in Bluetooth range [66]. The app for i-Que Intelligent Robot was able to be downloaded by anyone [66]. This gave individuals the ability to find the i-Que when in Bluetooth range and connect with the robot's voice via text field [66]. Toy-Fi Teddy's lack of authentication enabled hackers to not only send messages, but receive voice messages from children [66]. Cloud pets, which allow messages to be sent to children from friends was also able to be hacked because the Bluetooth connection was unsecure [66].

Building security into the hardware of these toys could prevent these types of Bluetooth exploits.

7.7. Issues with Vulnerabilities in Commercial Products

Bluetooth devices are not easily upgradable, specifically the hands-free systems used in vehicles. Therefore, devices with older versions of Bluetooth are left vulnerable to attacks.

It is also worth mentioning, that while security experts advocate that one should turn off Bluetooth when not in use, some companies' processes (e.g., Apple) are contradictory to this. Updates pushed out to Apple devices result in Bluetooth being turned on by default. The devices affected by the updates

include iPods, iPhones, and iPads. With the number of Apple device users, Apple could possibly and unnecessarily be exposing its customers to Bluetooth attackers.

8. Recommendations to Secure Bluetooth Communications

While not all attacks can be prevented, it is important to take the necessary steps to secure Bluetooth communications.

A. Recommendations for Users

Users should educate themselves on Bluetooth technology and proper security practices. Before purchasing IoT devices, users should do their due diligence on the device's security features and capabilities. Device owners should frequently visit the device manufacturer's website to be cognizant of firmware updates or patches that have been issued [67].

B. Recommendations for Manufacturers and Product Engineers

Engineers should identify security principles and apply them throughout the development of a product [67]. Developing threat models and applying knowledge learned from previous attacks could help prevent repeat attacks, as well as new foreseeable threats [67]. They should be aware of present vulnerabilities and update firmware and issue patches as necessary. Manufacturers should be sure to inform users of these updates via their website or email for registered users. They should also ensure devices have the most recent version of Bluetooth. Finally, they should develop documentation for users to help increase their awareness on how to secure their devices.

9. Conclusions

Bluetooth has gained acceptance worldwide and has become a common feature in our everyday wireless devices. The technology has been available for about 15 years and has become the go-to, as well as a convenient, solution for connecting devices, including phones, cameras, televisions, speakers, headphones, smartwatches, medical devices, as well as personal assistants. The ability to utilize the technology's capabilities for voice and data transmission over short distances has contributed to its popularity.

In this article, we reviewed Bluetooth security including security services and features. We then discussed vulnerabilities in various versions of Bluetooth, as well as numerous Bluetooth threats, which are largely due to the process of pairing. We also examined Bluetooth risk mitigation and countermeasures. Finally, we presented real-life exploitations and risk mitigations of Bluetooth commercial products, as well as provided recommendations on how to secure Bluetooth communications.

There are currently a wide variety of security vulnerabilities that are affecting Bluetooth technology, as well as Bluetooth-connected devices. For this reason, it is important for users to understand the risks involved with using Bluetooth technology on their devices, as well as the mitigation techniques that can be used to protect their devices and information from attackers. Implementing the recommended security measures above will help to mitigate any Bluetooth related risks. Every individual should be responsible for securing their Bluetooth communications, as there is not one trusted, central party taking the necessary action.

Future work should consider extending the above analysis to other wireless technology standards such as Wi-Fi and ZigBee. Also, future research could potentially focus on power attacks on BLE, as it is targeted for low-energy applications.

Funding: This research was funded by Fordham University, Faculty Research Program.

Conflicts of Interest: The authors declare no conflict of interest.