B

Bluetooth Low Energy (BLE) Security and Privacy

Yue Zhang[1], Jian Weng[1], Rajib Dey[2], and Xinwen Fu[2]
[1]College of Information Science and Technology, Jinan University, Guangzhou, China
[2]Department of EECS, University of Central Florida, Orlando, FL, USA

Synonyms

Provide synonyms to the article title

Definitions

This survey focuses on privacy and security of Bluetooth Low Energy (BLE), covering the attack vectors and corresponding countermeasures.

Bluetooth Low Energy

We are approaching toward a new era of the Internet of things (IoT). Manufacturers are rushing to enrich their products with more IoT functionalities. Bluetooth Low Energy (BLE/BTLE) will act as one of the backbones for IoT wireless communication technologies. Compared with the Bluetooth Classic technology, BLE provides larger coverage area while maintaining a lower power consumption. According to the Bluetooth SIG forecast, over 90% mobile devices will support BLE by 2018 (Group 2014).

Bluetooth Low Energy (BLE) is part of Bluetooth Core Specification and is a wireless technology specifically designed to be used for novel applications in IoT. It was released in 2010 along with Bluetooth 4.0. Since then BLE 4.1, BLE 4.2, and BLE 5 (Wikipedia 2018) have been released alongside its classic counterpart.

BLE is not compatible with Bluetooth Classic. Thus, a device implementing only the low energy feature cannot communicate with a device that only implements Bluetooth Classic (Gomez et al. 2012). Other differences between BLE and Bluetooth Classic are listed in Table 1.

In the rest of this survey, we will discuss two main protocols of BLE, its privacy features, and the technical details that protect the BLE privacy. We will discuss the security issues of BLE, involving security goals and association models for secure communication. We will introduce major attacks on BLE protocols, such as DOS attack, eavesdropping, and the man-in-the-middle (MITM) attack. We will discuss vulnerabilities and design issues that cause these attacks. Finally, we will give suggestions on how to make BLE device more secure.

ATT and GATT Protocol

The ATT protocol is a client-/server-based protocol (Bluetooth 2014). The server maintains a

X.(S.) Shen et al. (eds.), *Encyclopedia of Wireless Networks*,
https://doi.org/10.1007/978-3-319-32903-1_298-1

Bluetooth Low Energy (BLE) Security and Privacy, Table 1 Differences between BLE and Bluetooth Classic (Wireless World 2018)

Features	Bluetooth Classic	BLE (Bluetooth Low Energy)
Range	<30 m	More than 50 m
Power consumption	30 mA	Less than 15 mA
Speed	700 Kbps	1 Mbps
Frequency channels	79 channels	40 channels (includes 37 data channels and 3 advertising channels)
Modulation	GFSK (modulation index 0.35)	GFSK (modulation index 0.5)
Latency in data transfer	100 ms	3 ms
Message size(bytes)	358 (Max)	8–47
Throughput	0.7–2.1 Mbps	Less than 0.3 Mbps
Slaves	7	Unlimited

Bluetooth Low Energy (BLE) Security and Privacy, Table 2 Attribute for Device Name

Handle	Type (UUID)	Value	Permissions
0×0001	Device Name (0×2900)	Smart Lock	Read

set of attributes which can be accessed by the client once connected. Each of these attributes has a handle, a type with a universally unique identifier(UUID), a value, and a set of permissions. The handle is a 16-bit, non-zero value that is assigned by the server to uniquely index an attribute. The type is defined by UUID. Given the burden of transferring UUID values, the typical 128-bit UUID is converted to the 16-bit UUID in BLE. The UUID indicates the functionality of an attribute. Permissions restrict the accessibility of each attribute for another device, such as *read*/*write*, and *notify*. Table 2 is an example that demonstrates an attribute.

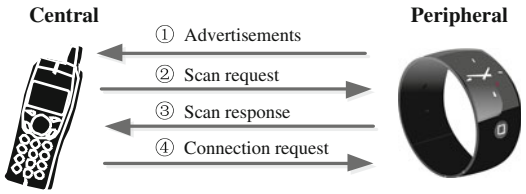
The GATT protocol is based on the ATT protocol and organizes attributes into *services* (Townsend et al. 2014). A service contains zero or several *characteristics*. Each characteristic contains zero or several *descriptors* as well. Characteristics are containers for user data. These descriptors are attributes that convey other information about a characteristic and its value. Services may include other services as their building blocks. The major service that contains other services is called the primary service, while the auxiliary ones are named secondary services. Secondary services must be included in primary services.

Connection Setup

There are two roles in the connection setup procedure. The client is named central device. The central device can be some smart terminals that often run on an operating systems. For example, our smartphone often plays the role of a central device. The server is named peripheral device. A peripheral device is normally a device that has specific functions, such as a smart thermometer, a smart light, or a smart lock. Sometimes, our smartphone can also be a peripheral, as long as it can provide services to some other devices. A peripheral can advertise and notify other devices that it is there. The central device responds to these advertisements with a scan request and then sets up a connection. Note that the advertisements and scan modes vary from each other (Bluetooth 2014).

We elaborate the connection setup procedure in Fig. 1.

1. The peripheral device broadcasts *advertisement* packets indicating it is connectible. These advertisement packets often contain the basic information about the device, such as the manufacturer.
2. The central device receives these advertisements and then sends a *scan* request to the peripheral to query more information about the peripheral.



Bluetooth Low Energy (BLE) Security and Privacy, Fig. 1 Initial states of Bluetooth Low Energy connection

3. The peripheral device responds with a *scan response* packet that may contain more detailed information of itself, such as device name.
4. After handling the *scan response*, the central then sends the *connect request*, if the response is in line with its expectations. The *connect request* includes the channel information that these two devices will follow during the communication.

Once a connection is established, the central device becomes the **master**, while the peripheral device becomes the **slave**. The **master** may have several **slaves** at the same time, while the **slave** can only have one **master** in 4.0 (Bluetooth 2010). This situation changes in the 4.1 (Bluetooth 2013), which allows one **slave** to have links to more than one **master** at a time.

Privacy of Bluetooth Low Energy

Privacy in BLE is an ability that avoids one device being tracked by other untrusted devices. This type of ability depends on if the device can prevent its address being decoded by other unauthorized devices. If this address gets exposed, other devices may follow the address from the advertising phase. To maintain the privacy, two devices share a secret key known as the Identity Resolving Key (IRK). The IRK is used to generate a random address that only can be resolved by the peer device who is maintaining the IRK. In the rest of this section, we will describe this process in detail.

Bluetooth Device Address

Each Bluetooth device is allocated a uniquely 48-bit Bluetooth device address (BD_ADDR). These addresses are classified as public device addresses and random device addresses. The public device address consists of two 24-bits long values, known as a company ID and a company-assigned ID. The random device address is a randomly generated address. It can be either static random address or private random address. The static random address does not change in one power cycle, while the private random address can change in one power cycle. The private random address may change at every connection. In terms of the private random address, two subtypes of addresses are included, non-resolvable private address and resolvable private address. The resolvable private address (RPA) is the foundation for privacy in BLE.

RPA Generation and Resolution

A RPA is generated from a random number (known as prand) and an Identity Resolving Key (IRK) which is shared during the pairing process. The RPA is used by one device, and only the corresponding device that maintains the same IRK could resolve it. The format of a RPA is shown in Fig. 2.

Assume that two devices already have the IRK, respectively. To generate a resolvable private address, a device first generates a 24-bit random value (known as prand) and then passes this value with the aforementioned IRK to a hash function. The random address is concatenated by the prand and the output of this hash function, as shown in Eq. (1). Correspondingly, once another device receives this RPA, it splits it to extract the hash value and prand. Then, it performs the hash calculation as well as the first device. The RPA is resolved only when the output matches the extracted hash value.

$$\begin{aligned} Hashvalue &= Hash(Prand, IRK) \\ RPA &= Hashvalue||Prand, \end{aligned} \tag{1}$$

Bluetooth Low Energy (BLE) Security and Privacy, Fig. 2 The format of resolvable private address

Other Policies

Device filtering is a set of policies that avoid untrusted connections. It is also an effective way to reduce power consumption. There are three types of policies: advertising filter policy, scanner filter policy, and connection initiator filter policy. Suggested by their name, these policies affect different states before setting up the connection. When the policies are enabled, a device does not need to respond to every peer device, since most of the devices are filtered out. To do this, a device should maintain a white list locally, which stores the address and its type. For example, the advertising filter policy determines in which way could an advertiser process the scan/connection request. When the specific advertising filter policies are applied, devices may only allow scan/connection requests from whitelist.

The directed connectible advertisement is another way to protect the privacy of the device and avoid unwanted scan or connection requests. This special type of advertisement contains a RPA for the device it intends to connect to. When this type of advertisement is enabled, the device shall only share its information with the ones it wants to connect to.

Security of Bluetooth Low Energy

The application of the BLE technology will continue to grow exponentially in coming years. It also draws attackers attention at the same time. Thus, the security of BLE will have colossal importance.

The security architecture of Bluetooth has evolved over time. Initially, Bluetooth was not as secure as it is today. It could only provide weak protection for the message integrity and can be easily forged. After Version 2.1 + EDR, the Secure Simple Pairing (SSP) was introduced to mitigate the situation. This was a great revolution for Bluetooth Security. Four association models, which are practiced as of now, were also introduced at this version, known as Just Works, Numeric Comparison, Passkey Entry, and Out-Of-Band (OOB).

Along with the release of Version 4.0, the Bluetooth Special Interest Group (Bluetooth SIG) added the security model for Low Energy. However, only three types of association models were supported at this time (Just Works, Passkey Entry, and Out-Of-Band). Although they have the same name as SSP, the security provided is not the same. Some of these association models have been proved to be vulnerable.

In Version 4.1, Bluetooth SIG upgraded SSP by supporting the P-256 elliptic curve and added FIPS-approved algorithms for device authentication, which was known as Secure Connections feature. They also added message integrity by supporting AES-CCM. But they did not change the security features for the Low Energy counterpart.

It is generally acknowledged that the updates in Version 4.2 are crucial for BLE, which greatly enhances its security. At first, it added the Secure Connections feature to Low Energy, such as upgrading the LE pairing model by supporting the AES-CMAC and P-256 elliptic curve algorithms. Secondly, it included the Numeric Comparison as a new association model in Secure Connections. Lastly, it also applied the Secure Connections features to key generation.

Security Goals

To maintain the security in Bluetooth Low Energy wireless technology, the designers must address two security goals: protection against passive eavesdropping attacks and protection against man-in-the-middle (MITM) attacks. Passive eavesdropping attacks occur when a user is communicating with the other, while an attacker is listening to their conversation

Bluetooth Low Energy (BLE) Security and Privacy, Table 3 How association model can be used according to different I/O capabilities

Association models / I/O	Display	Display, Yes/No	Keyboard	No input/output	Keyboard, display
Display	Just works	Just works	Passkey entry	Just works	Just works
Keyboard only	Passkey entry	Passkey entry	Passkey entry	Just works	Passkey entry
No input/output	Just works	Just works	Just works	Just works	Just works
Keyboard, display	Passkey entry	Numeric comparison	Passkey entry	Just works	Numeric comparison

stealthily without their consent. The MITM attack is a scenario where the attacker secretly emulates the partner of both communicating devices during a conversation. The attacker here has the ability to listen and change the data of that private communication.

To defend against passive eavesdropping, the traffic between two devices must be encrypted. In the earlier versions of BLE (4.0 and 4.1), the encryption is vulnerable to a brute-force attack. The Temporary Keys (TKs), which are used for further key generation and distribution, are not strong enough. However, this situation was mitigated in the newly released BLE Secure Connection. Although passive eavesdropping can be thwarted by a strong link layer encryption, the link layer encryption may be vulnerable to MITM attacks. Some association models can effectively defend against MITM attacks.

Association Models

Association models are mechanisms that determine how two devices could authenticate each other and securely exchange data with each other. In BLE, association models are used during the pairing process. As of now, there are four association models. They are Just Works, Numeric Comparison, Passkey Entry, and Out-Of-Band (OOB). Which type of the association model would be used depends on the I/O capabilities of both devices. Table 3 shows the mapping of association models according to different I/O capabilities. The rows and the columns represent the I/O capabilities of two devices, respectively. Display stands for whether the device is equipped with a display, such as a screen. While keyboard stands for whether it has a keyboard. Yes/no here can be a space key, which could be used by the user to confirm or reject a value displayed on a screen.

Now we will discuss these association models in detail. We will focus whether these models could or could not cover the security goals mentioned earlier:

- **Numeric Comparison (only for LE Secure Connections)**: Numeric Comparison is applicable when both devices have the capability to display and confirm information. For example, two smartphones want to pair with each other. In this association model, both devices display a six-digit number on its screen. If these two numbers are the same, they are authenticated properly. The numbers are generated from the public key of each device, a nonce, and an 8-bit code by using the AES-CMAC function. During MITM attacks, public keys of both devices will not match. As a result, the numbers displayed on each device will not match. In this way we would avoid MITM attacks effectively.
- **Just Works**: Just Works is designed for scenarios where at least one of the devices has no input/output capability. For example, a mouse and a laptop want to pair with each other. In this association model, they use the same protocol as the Numeric Comparison, but the six-digit number never shows up. Because Just Works does not provide an interface for a user to know if the number matches, it is vulnerable to MITM attacks.

- **Passkey Entry**: Passkey Entry is designed for scenarios where one device supports entering the six-digit number (Passkey), but does not have the capability to display. For example, a keyboard and a laptop want to pair with each other. In this association model, the final confirm value is generated from the public key of both devices, a nonce and the one part of the Passkey by using AES-CMAC function.

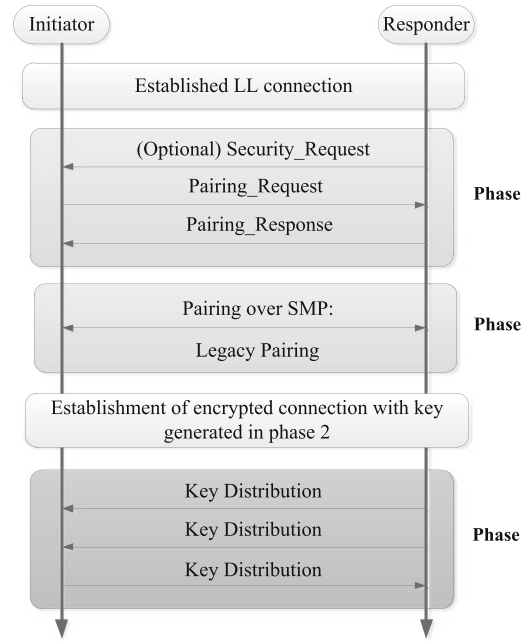
  In LE legacy pairing (known as 4.1 and 4.0), Just Works and Passkey Entry both suffer from the passive eavesdropping, since it does not use Elliptic Curve Diffie-Hellman as Secure Simple Pairing does not.
- **Out-of-Band (OOB)**: Out-of-Band is designed for scenarios where both devices possess an extra communication channel, such as WLAN or NFC, for example, two phones equipped with NFC. In this association model, the capability to defend against MITM attacks or passive eavesdropping attacks depends on what type of communication channel being used and security of that communication channel.

Pairing and Encryption

Pairing occurs when two BLE devices build a connection, which is a process used for key exchanging. Pairing involves three phases. One of the association models is used for authentication during the pairing process. Depending on the version of Bluetooth Low Energy, there are two types of pairing methods. In 4.1 and 4.0, LE legacy pairing is used, while LE Secure Connections are supported in 4.2.

LE Legacy Pairing

We first introduce LE legacy pairing. It contains three steps, as shown in Fig. 3. In the first step, both devices exchanges their I/O capabilities and security requirements, which is used to determine the association model. In the second step, the two devices determine a Temporary Key (TK) and use this to generate the Short-Term Key (STK). In the third step, a Long-Term Key (LTK) would be generated for future communications.



Bluetooth Low Energy (BLE) Security and Privacy, Fig. 3 LE legacy pairing

Determining Temporary Key (TK) TK is a 128-bit key that is used to generate the Short-Term Key (STK). Recall that three types of association models are supported to share the TK, known as Just Works, Passkey Entry, and OOB. In Just Works, the TK is always set to 0. In Passkey Entry, validated portion of TK is a random number between 000000 and 999999, while other bits can be padded with 0. Only in OOB, the 128-bit TK can be applied. During the pairing process (Zegeye 2015), few values exchanged in plain text in the air, which can expose the TK to the attacker. These values are confirm value, devices information, pair request and response, Mrand (random value from master), and Srand (random value from slave). By knowing the *confirm value generation function* c1 that is defined in 4.0 (as shown in Eq. (2)), attackers could use these values to brute-force the TK in the Passkey Entry model (only 1,000,000 possibilities). Not to mention that in the Just Works model, TK is set to 0.

$$Mconfirm = c1(TK, Mrand, Pairinginfo, deviceinfo)$$
$$Sconfirm = c1(TK, Srand, Pairinginfo, deviceinfo) \quad (2)$$

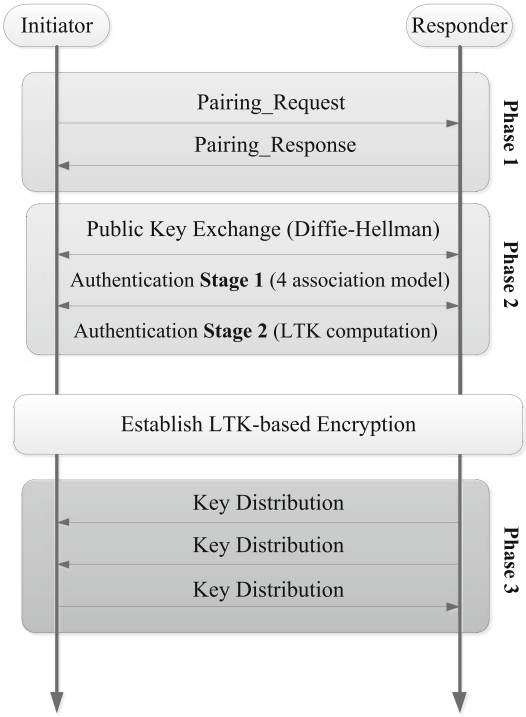
Determining Short-Term Key (STK) After checking the correctness of the confirm values, devices exchange the generated random numbers (Srand and Mrand). Then the Short-Term Key (STK) is generated using the TK and random numbers (as shown in Eq. (3)).

$$STK = AES128(TK, Srand||Mrand) \quad (3)$$

Determining Long-Term Key (LTK) Devices which support encryption in the Link Layer Connection State in the slave role will generate LTK, the Encrypted Diversifier (EDIV) and 64-bit Random Number (Rand), IRK (Identity Resolving Key), and CSRK (Connection Signature Resolving Key). Among them, the LTK is used to generate session key (SK) for Link Layer Connection, while the IRK is used for random resolvable private address resolution. CSRK supports the data signing operation, which adds another security layer. A signature will be generated with the signing algorithm and a counter by using the CSRK to avoid replay attacks.

LE Secure Connections

Compared to the LE legacy pairing, LE Secure Connections offer significantly stronger security. The TK and STK are not used. Instead, a single LTK is used to encrypt the connection, which is exchanged by using the standard (FIPS) compliant Elliptic Curve Diffie-Hellman (ECDH) public key (Sachin Gupta Rohit Kumar 2018). As shown in Fig. 4, we introduce the three steps in LE Secure Connections. In the first step, both devices settle on a certain set of domain parameters. In the second step, the two devices exchange their public keys and compute a Long-Term Key (LTK). In the third step, a set of keys are generated and distributed for future communications.



Bluetooth Low Energy (BLE) Security and Privacy, Fig. 4 LE Secure Connections

Public Key Exchange In this step, each device sends its public key to the other device (PK1 and PK2). This public key is generated from the Elliptic Curve Diffie-Hellman (ECDH). It is an effective way to thwart eavesdropping, since the private key is never exposed in the air during the pairing.

Authentication Stage 1 To perform the authentication, one of the aforementioned association models must be used in the first authentication stage. At this stage, each device will generate a nonce (N1 and N2), which is needed to calculate a commitment (C1 and C2). Taking the Numeric Comparison as an example, the process can be described as Eq. (4). During this stage, N1 and N2 are exchanged, while the generated value D is displayed on each devices. Since there are no extra user inputs in Numeric Comparison, the user input values (ri1 and ri2) are set to 0. The Just Works model follows the same protocol as Numeric Comparison. However, Just Works is

still vulnerable to the MITM attack, since there are no user interactions during the whole process. For the OOB or Passkey Entry, the process is similar. For example, in the Passkey Entry model, the *ri* values are not set to 0 but set to the input values of users. And the last calculation is also more complex.

$$\begin{aligned} Device1 : C1 &= f(PK1, PK2, N1, ri1); \\ &(ri1 = 0) \\ Device2 : C2 &= f(PK1, PK2, N2, ri2); \\ &(ri2 = 0) \\ D &= g(PK1, PK2, N1, N2) \end{aligned} \quad (4)$$

Authentication Stage 2 In this stage, each device collects the exchanged values to generate a Long-Term Key. Both devices will also perform DHKey (Diffie-Hellman key) check process after LTK is generated. Each side uses the DHkey as input to compute new confirmation value by using the previously exchanged information in the DHKey check process. If the check passes through on both sides, the devices start to generate the LTK. LTK is used for the link encryption and sharing keys.

Attacks on Bluetooth Low Energy

It is generally acknowledged that security in LE legacy pairing is vulnerable from many perspectives. Attackers are more likely to choose 4.1 and 4.0 BLE versions as their target as they use the legacy pairing. On the other hand, vulnerabilities can also stem from the implementation or design by any particular device manufacturer.

DoS Attack

A denial-of-service attack (DoS attack) is an attack that renders the network or computing resource unavailable to the legitimate users by disrupting the services. In BLE, the DoS attack is easy to launch and hard to defend.

In Version 4.0, the slaves can only have one master. If the master is connected to the slave, the slave will stop broadcasting the advertisement packets and will become invisible to any other devices. This makes it vulnerable to DoS attacks, since attackers could enhance their advertisement rate to win such a connection race. The situation has been mitigated in Version 4.1 as it supports the multi-master model.

The jamming attack is another DoS attack. It blocks the advertisements of a salve and makes it unconnectable. As an example, Sebastian et al. present a design of selective jammers against BLE advertising, which can block advertisements of a specific salve from being revived by other devices (Brauer et al. 2016). They theoretically prove that the selective jammer is effective and hard to detect.

Another DoS attack is known as denial-of-sleep (DoSL) attack. It can paralyze the device by draining out their battery. In this type of attack, the attackers will consume the power of the device by connecting/disconnecting with them repeatedly (Uher et al. 2016).

Passive Eavesdropping

Passive eavesdropping occurs when two devices communicate with each other and an attacker is listening to their conversation stealthily without their awareness. BLE versions 4.0 and 4.1 are subject to this type of attack, since they use the LE legacy pairing method. Specifically, for the Just Works and the Passkey Entry of LE legacy pairing, TK is guessable, since the possibility of a TK is very limited. Since BLE uses only 40 frequency channels compared with 79 channels in Bluetooth Classic, BLE traffic is easy to follow. Manufacturers are more devoted to usability than security. Their user guides often do not mention pairing or security even if a more secure pairing method is available. Users may not use any secure pairing at all, and the BLE communication traffic will be in plaintext. Therefore, the attacker can easily deploy attacks such as command replay or injection against their products (Jasek 2016).

In 2011, Michael Ossmann released the first open-source low-cost Bluetooth test tool, named Ubertooth One (Ossmann 2011). In 2013, Mike Ryan built a BLE sniffer over Ubertooth and explored the vulnerability in the LE legacy pairing. A tool called *Crackle* was released at that

time to attack Just Works and the legacy Passkey Entry pairing methods (Ryan et al. 2013) of BLE 4.0 and 4.1. In 2014, The Adafruit Bluefruit LE sniffer was introduced Townsend (2014). The Passkey Entry pairing method in BLE 4.2 is secure with a new passkey exchange protocol.

MITM Attack

The MITM attack brings havoc to the BLE protocol. In a MITM attack, the attacker pretends to be a one device (a slave or a master), intercepts the connection from the its peer device, and then relays messages between the two original communicating parties. The Just Works pairing method of any version of BLE suffers from this attack. For example, for LE Secure Connections, Just Works does not check if the six-digit confirm number matches, mostly because there is no I/O such as a display on the devices. Sometimes, even if a victim device has necessary IO capabilities, an attacker could trick them force to use the Just Works. For example, an attacker impersonates as one device and tells the other device that it has no I/O capability. As a result, the only choice is the Just Works pairing method. Therefore, the MITM attack can then be launched (Haataja and Hypponen 2008).

The spoofing attack is a special type of MITM attack or can be regarded as a building block of the MITM attack. The attacker pretends to be a legitimate device and communicates with the target collecting sensitive information, such as password. That is, in a spoofing attack, the attacker does not relay messages, but pretends to be the slave of interest communicating with the master. Numeric Comparison (only for LE Secure Connections) and Passkey Entry could avoid this type of attack. The OOB pairing can defeat the MITM attack if the OOB channel is secure.

To perform a MITM attack, an attacker needs two role players (i.e., two BLE devices), a DoS attacker and a fake service provider. The DoS attacker blocks the communication between the master and the slave, while the fake service provider forges a fake Bluetooth MAC address that is the same as the victim slave's MAC address, fake advertisements, and fake services. To know all such information, the attacker has to perform sniffing and reverse engineer the applications and even the target hardware sometimes.

Attacks Against BLE Apps

Given the extra cost (such as a display or a Wi-Fi adapter) of using more secure association models and convenience, BLE device designers may prefer to use Just Works. As we know, Just Works is subject to the MITM attack. To address this, the developers of the smart device can implement extra security mechanisms on the application layer, such as password-based or challenge-response authentication.

We now discuss design pitfalls of the application layer security mechanisms.

Mis-Bonding (or Co-located attack in the context of BLE) Under the context of Bluetooth-enabled smartphones, Mis-Bonding refers to the problem that an unauthorized app can access the sensitive user data on the smart device. The data was only designed for the official app initially. Because of vulnerabilities existing in the smartphone's permission mechanism, the data becomes accessible for all apps, even a malicious one. In this case, a malicious app could stealthily gather the data of a device that paired with the smartphone like an official app and then fed the data to a remote malicious server (Naveed et al. 2014). Because this is a vulnerability in application layer, the link layer encryption is useless in this case.

Password authentication Many BLE-enabled devices use the password authentication. The implementation is often problematic.

- **Clear-text data transformation:** Passwords transmitted in plaintext can be captured through passive eavesdropping.
- **Unlimited login attempts:** This allows the attacker to launch a brute-force attack. However, there is a trade-off if the number of login attempts is limited. Limiting login attempts allows the attacker to perform DoS attacks.
- **Small password space:** Many devices only support six-digit passwords, since a complex

Bluetooth Low Energy (BLE) Security and Privacy, Table 4 Bluetooth Classic attacks vs. these in BLE

Attack classification	attack description	Threat level	Examples	If exits in BLE
Surveillance	Gather user information (not privacy)	Low	Blueprinting (tsb 2017a)	Y
Range extension	Extend the attack range	Low	BlueSniping (tsb 2017c)	Y
Obfuscation	Hide attacker's identity	Low	bdaddr (Whigu 2018)	Y
Fuzzer	Submit nonstandard input to achieve malicious result	Medium	BlueSmack (tsb 2017b)	Y (Sullivan 2015)
Sniffing	Passive eavesdropping	Medium	FTS4BT (fte 2018)	Y
DoS	Bock the services	Medium	Signal jamming (Köppel 2013)	Y
Brute-force address	Brute-force the address of devices to trace connection	Medium	Address exhaustion (altCross et al. (2007))	Y
Brute force in pairing	Brute force in the temporary key or PIN during the pairing	High	btpincrack (Project 2017)	Y
Mis-Bonding	Unauthorized app can access the sensitive data from smartphone	High	DMB (Naveed et al. 2014)	Y
Relay attack	With the help of relay device, implement unwanted unlock	High	Home lock attack (Ho et al. 2016)	Y
MITM	Attacker mask itself as parter to achieve malicious result	High	bthidproxy (Mishra 2012)	Y
Malware	Malwares that propagated through Bluetooth	High	CommWarrior (Wei et al. 2008)	Y
Unauthorized access	Control the smartphone and steal sensitive data	High	Car Whisperer (tsb 2017d)	Y

and long password may reduce usability. This makes the devices suffer from brute-force attacks. Designers uses specific a hash function or an encoding function to make up the shortness, but finally proved to provide few valid mitigations for this. Attacker can also use multipliable devices in parallel to perform an off-line brute-force attack.

Bluetooth Low Energy v.s. Bluetooth Classic

In Dunning (2010), John Paul et al. summarized the basic attacks in Bluetooth Classic. We add new attacks into the table and try to identify the difference between BLE attacks and attacks against Bluetooth Classic. As shown in Table 4, most attacks against Bluetooth Classic are also threats against BLE.

Defense Measures

Based on the discussion in this paper, we present guidelines for BLE security as follows.

- **Prevent advertisement abuse:** Do not put any sensitive information into broadcast packets.
- **Enforce the user to pair**: This is an effective way to thwart the passive eavesdropping.
- **Use version 4.2 and above**: BLE 4.1 and 4.0 suffer from the TK brute-force attack.
- **Do not use the Just Works if possible**: Just Works is vulnerable to MITM attacks, even in BLE 4.2.
- **Implement encryption in application layer** Designers should not invent their own cryptographic algorithms and protocols if there are well-known ones that can do the job. Encryp-

tion should be adopted to prevent passive monitoring data leaks.

- **Strengthen the password authentication**: Increasing the password space and limiting the login attempts are the most straightforward but effective ways to strengthen password authentication, and the communication should be encrypted.
- **Protect source code carefully**: It can be seen that a lot of attacks can be launched partially because attackers are able to reverse engineer the BLE apps (Cyr et al. 2014; Rieck 2016; Allan and Mistry. 2014)). Therefore, obfuscation, JNI, code encryption, and other methods should be applied to protect source code.
- **Implement OTA (over-the-air) capabilities into each device**: Enough space should be allocated in the device memory so that code can be updated to patch holes.

Conclusion

This survey focuses on privacy and security of Bluetooth Low Energy (BLE), covering the attack vectors and corresponding countermeasures. Given the vulnerabilities in BLE 4.0 and 4.1, we highly recommend that BLE 4.2 and 5 should be adopted whenever possible. The Just Works pairing strategy in all BLE versions are subject to MITM attacks. It should not be used. In case that Just Works has to be used because of cost and usability issues, application level encryption and authentication is recommended.

References

Allan A, Mistry S (2014) Reverse engineering the estimote. http://makezine.com/2014/01/03/reverse-engineering-the-estimote/. Accessed 26 June 2018

Bluetooth S (2010) Bluetooth core specification version 4.0. Specification of the Bluetooth System

Bluetooth S (2013) Bluetooth core specification version 4.0. Specification of the Bluetooth System

Bluetooth S (2014) Bluetooth core specification version 4.2. Specification of the Bluetooth System

Brauer S, Zubow A, Zehl S, Roshandel M, Mashhadi-Sohi S (2016) On practical selective jamming of Bluetooth low energy advertising. In: 2016 IEEE conference on standards for communications and networking (CSCN). IEEE, pp 1–6

Cross D, Hoeckle J, Lavine M, Rubin J, Snow K (2007) Detecting non-discoverable bluetooth devices. In: International conference on critical infrastructure protection. Springer, pp 281–293

Cyr B, Horn W, Miao D, Specter M (2014) Security analysis of wearable fitness devices (fitbit). Massachusetts Institute of Technology, p 1

Dunning J (2010) Taming the blue beast: a survey of Bluetooth based threats. IEEE Secur Priv 8(2):20–27

fte (2018) Fts4bt™ bluetooth® protocol analyzer and packet sniffer. http://www.fte.com/products/FTS4BT.aspx

Gomez C, Oller J, Paradells J (2012) Overview and evaluation of Bluetooth low energy: an emerging low-power wireless technology. Sensors 12(9):11734–11753

Group BSI (2014) Mobile telephony market. https://www.bluetooth.com/news-events/news. Accessed 26 June 2018

Haataja KM, Hypponen K (2008) Man-in-the-middle attacks on Bluetooth: a comparative analysis, a novel attack, and countermeasures. In: 3rd international symposium on communications, control and signal processing, 2008. ISCCSP 2008. IEEE, pp 1096–1102

Ho G, Leung D, Mishra P, Hosseini A, Song D, Wagner D (2016) Smart locks: lessons for securing commodity internet of things devices. In: Proceedings of the 11th ACM on Asia conference on computer and communications security. ACM, pp 461–472

Hruska J (2017) Bluetooth low-energy technology isn't just another Bluetooth revision its a whole new technology. http://suo.im/4KhIEQ. Accessed 26 June 2018

Jasek S (2016) Gattacking Bluetooth smart devices. In: Black Hat USA conference

Köppel S (2013) Bluetooth jamming. ETH Zürich, Zürich

Mishra N (2012) Defence against man in middle attack in secure simple pairing. J Glob Res Comput Sci 3(5):78–82

Naveed M, Zhou Xy, Demetriou S, Wang X, Gunter CA (2014) Inside job: understanding and mitigating the threat of external device mis-binding on android. In: NDSS

Ossmann M (2011) Project Ubertooth: building a better Bluetooth adapter. http://ossmann.blogspot.com/2011/02/project-ubertooth-building-better.html. Accessed 26 June 2018

Project TO (2017) btpincrack. http://openciphers.sourceforge.net/oc/btpincrack.php. Accessed 26 June 2018

Rieck J (2016) Attacks on fitness trackers revisited: a case-study of unfit firmware security. arXiv preprint arXiv:160403313

Ryan M et al (2013) Bluetooth: with low energy comes low security. WOOT 13:1–7

Sachin Gupta Rohit Kumar (2018) Ble v4.2: creating faster, more secure, power-efficient designs—part 4. http://www.electronicdesign.com/communications/ble-v42-creating-faster-more-secure-power-efficient-designs-part-3

Sullivan H (2015) Security vulnerabilities of Bluetooth low energy technology (BLE). Tufts University

Townsend K (2014) How the Bluetooth sniffing on android works. Sebastopol, California. https://learn.adafruit.com/introducing-the-adafruit-bluefruit-le-sniffer/introduction

Townsend K, Cuff C, Davidson R, Davidson A (2014) Getting started with Bluetooth low energy. O'Reilly Media, Inc.

trifinite (2017) Bluesnarf. https://trifinite.org/trifinite_stuff_bluesnarf.html. Accessed 26 June 2018

tsb (2017a) What is blueprinting in bluetooth security? https://www.thesecuritybuddy.com/?s=Blueprinting. Accessed 26 June 2018

tsb (2017b) What is bluesmack attack? https://www.thesecuritybuddy.com/?s=BlueSmack. Accessed 26 June 2018

tsb (2017c) What is bluesniping? https://www.thesecuritybuddy.com/bluetooth-security/what-is-bluesniping/. Accessed 26 June 2018

tsb (2017d) What is car whisperer? https://www.thesecuritybuddy.com/bluetooth-security/what-is-car-whisperer/. Accessed 26 June 2018

Uher J, Mennecke RG, Farroha BS (2016) Denial of sleep attacks in Bluetooth low energy wireless sensor networks. In: Military communications conference, MILCOM 2016-2016 IEEE. IEEE, pp 1231–1236

Wei X, Li Z, Chen Z, Yuan Z (2008) Commwarrior worm propagation model for smart phone networks. J China Univ Posts Telecommun 15(2):60–66

Whigu (2018) Change your Bluetooth device MAC-address. http://blog.petrilopia.net/linux/change-your-bluetooth-device-mac-address/. Accessed 26 June 2018

Wikipedia (2018) Bluetooth low energy. https://en.wikipedia.org/wiki/Bluetooth_Low_Energy. Accessed 26 June 2018

Wireless World R (2018) Bluetooth vs BLE-difference between Bluetooth and BLE(bluetooth low energy). http://www.rfwireless-world.com/Terminology/Bluetooth-vs-BLE.html. Accessed 26 June 2018

Zegeye WK (2015) Exploiting Bluetooth low energy pairing vulnerability in telemedicine. In: International telemetering conference proceedings, international foundation for telemetering