A Basic Introduction to BLE Security

- Introduction:
- Security Issues Facing BLE:
- Pairing Overview:
- Pairing Methods for LE Legacy Connections (4.0, 4.1 and 4.2 devices):
- Pairing Methods for LE Secure Connections (4.2 devices only):
- Practical Considerations Concerning BLE Pairing Methods:
- Addressing:
- Conclusion:
- References:
- Appendix 1, Phase Three Keys and Values:
- Appendix 2, Relevant Links Concerning BLE Security:
- Appendix 3, Relation Between Pairing Methods and I/O capabilities:

Introduction:

Bluetooth Low Energy (BLE), is rapidly becoming one of the most common wireless standards in use today. Likewise, it is also becoming more commonly used in applications where sensitive information is being transferred. Thus, designers looking to integrate BLE into their products should be aware of the security features and limitations of this technology. This article seeks to give a basic overview of these features as well as give some insight into the theory behind them.

This article will focus on the BLE GAP (Generic Access Profile) Central and GAP Peripheral roles. The GAP Observer and GAP Broadcaster roles are typically used in applications with little to no security requirements and are thus not considered in this article.

For readers unfamiliar with the BLE standard, GAP is the layer of the BLE stack which determines the network topology of the BLE system. The GAP Central is typically the device which initiates the connection with the GAP Peripheral. Once the two devices are connected, they will perform a “pairing” process where they will exchange the information necessary to establish an encrypted connection. The devices may also perform a bonding process where the information from the pairing process is stored on the devices so that the pairing process does not have to be repeated every time the devices reconnect to each other.

Finally, this article will refer to BLE devices as 4.0, 4.1 or 4.2 devices in order specify which version of the Bluetooth spec that they are compliant with. Using these terms is somewhat incorrect as a Bluetooth Classic device that complies with Bluetooth Classic portion of the spec can also be referred to as a Bluetooth 4.x compliant device. However, since this article only focuses on BLE devices, the terms can be taken to refer only to devices that are compliant with the BLE portion of the respective spec.

Security Issues Facing BLE:

The main security issues with the pairing process and BLE in general are passive eavesdropping, man in the middle (MITM) attacks and identity tracking.

Passive eavesdropping is the process by which a third device listens in to the data being exchanged between the two paired devices. The way that BLE overcomes this is by encrypting the data being transferred using AES-CCM cryptography. While AES encryption is considered to be very secure, the key exchange protocols that BLE uses can introduce some severe security vulnerabilities which would allow an attacker to decrypt the data. Thus, the method by which the keys are exchanged, referred to as the “pairing method” or “association model”, has a great effect on the security of the connection.

MITM attacks are when a third device, which we will call the malicious device, impersonates the other two legitimate devices, in order to fool these devices into connecting to it. In this scenario, both the GAP Central and GAP Peripheral will connect to the malicious device which in turn routes the communication between the two other devices. This gives the legitimate devices the illusion that they are directly connected to each other when in fact their connection has been compromised. This setup not only allows the malicious device to intercept all the data being sent, but also allows it to inject false data into the communication or remove data before it reaches its intended recipient. A more detailed explanation of MITM attacks can be found on Wikipedia: https://en.wikipedia.org/wiki/Man-in-the-middle_attack. As with passive eavesdropping, the pairing method used determines how resilient the BLE connection will be to MITM attacks.

Identity tracking is where a malicious entity is able to associate the address of a BLE device with a specific user and then physically track that user based upon the presence of the BLE device. The way BLE overcomes this is by periodically changing the device address. A more detailed explanation of this process can be found in the Addressing section of this article.

Pairing Overview:

Pairing is the process by which two BLE devices exchange device information so that secure link can be established. The process varies somewhat between the BLE 4.2 devices and the older 4.1 and 4.0 devices. The processes are detailed in the sections below.

4.0 & 4.1 devices:

The pairing process for 4.0 and 4.1 devices, also known as LE Legacy Pairing, uses a custom key exchange protocol unique to the BLE standard. In this setup, the devices exchange a Temporary Key (TK) and use it to create a Short Term Key (STK) which is used to encrypt the connection. How secure this process is depends greatly on the pairing method used to exchange the TK, thus each pairing method is described in detail later in this article. The pairing process is performed in a series of phases shown below.

Phase One: This phase begins when the initiating device sends a 'Pairing_Request" to the other device. The two devices then exchange I/O capabilities, authentication requirements, maximum link key size and bonding requirements. Basically all this phase consists of, is the two devices exchanging their capabilities and determining how they are going to go about setting up a secure connection. It is also important to note that all data being exchanged during this phase is unencrypted.

Phase Two: Once phase one is complete, the devices generate and/or exchange the TK using one of the pairing methods. From there, the two devices then exchange Confirm and Rand values in order to verify that they both are using the same TK. Once this has been determined, they will use the TK along with the Rand values to create the STK. The STK is then used to encrypt the connection.

Phase Three: This phase is an optional phase that is only used if bonding requirements were exchanged in phase one. In this phase, several transport specific keys are exchanged. A full list of these keys and their functions can be found in appendix of this article.

4.2 devices:

BLE 4.2 devices are fully backwards compatible with BLE 4.0 and 4.1 devices, this means that 4.2 devices are capable performing the exact same pairing process as 4.0 and 4.1 devices. However, BLE 4.2 are also capable of creating what are known as LE Secure Connections. Instead of using a TK and STK, LE Secure Connections use a single Long Term Key (LTK) to encrypt the connection. This LTK is exchanged/generated using Elliptic Curve Diffie Hellman (ECDH) public key cryptography which offers significantly stronger security compared to the original BLE key exchange protocol.

In LE Secure Connections, both phase one and phase three of the pairing process are exactly the same as they are in LE Legacy connections. Thus, the only differences occur during phase two of the pairing process.

The way phase two works in LE Secure Connections is as follows. Both devices generate an ECDH public-private key pair. The two devices will then exchange their public keys and then start computing the Diffie-Hellman key. One of the pairing methods is then used to authenticate the connection. Once the connection is authenticated, the LTK is generated and the connection is encrypted.

Pairing Methods for LE Legacy Connections (4.0, 4.1 and 4.2 devices):

Just Works™:

In this method, the TK is set to 0. Thus, it is very easy for an attacker to brute force the STK and eavesdrop on the connection. Likewise, this method also offers no way of verifying the devices taking part in the connection and thus it offers no MITM protection.

Out of Band (OOB) Pairing:

In this method, the TK is exchanged using a different wireless technology such as NFC. The main advantage to this method is that a very large TK can be used, up to 128 bits, greatly enhancing the security of the connection. If the OOB channel is protected from MITM attacks, then it can be assumed that the BLE connection is also protected from MITM attacks as well. Likewise, as long as the OOB channel is immune to eavesdropping during the pairing process, then the BLE connection will also be immune from passive eavesdropping. Of the three legacy pairing methods (Just Works™, Passkey and OOB), OOB pairing is by far the most secure provided that the OOB channel employs sufficient security methods.

Passkey:

In this method, the TK is a 6 digit number that is passed between the devices by the user. The way this number is transferred can vary. One example would be to have one of the devices generate a random 6 digit number and display it on a LCD display. The user would then read the number and enter it into the other device using a keypad.

If an attacker is not listening in during the pairing process, then the passkey method gives fairly good protection from passive eavesdropping. However, if an attacker is present during the pairing process and is able to sniff the values being exchanged, then it is fairly trivial to brute force the TK and use it to derive the STK and decrypt the connection. The passkey method is generally considered to be secure from MITM attacks provided that the attacker is not able to obtain the passkey via some means other than the BLE connection. However, there is at least one theoretical MITM attack that is able to succeed without advanced knowledge of the passkey as detailed in the whitepaper: Bypassing Passkey Authentication in Bluetooth Low Energy by Tomas Rosa. For this reason, BLE applications that require the highest level of security should use either the OOB or Numeric Comparison pairing methods

Pairing Methods for LE Secure Connections (4.2 devices only):

Just Works™:

Once the devices exchange their public keys, the non-initiating device will generate a nonce, which is essentially a random seed value, and then use it to generate a confirmation value Cb. It then sends the Cb along with the nonce to the initiating device. At the same time, the initiating device generates its own nonce and sends it to the non-initiating device. The initiating device then uses the non-initiating device's nonce to generate its own confirmation value Ca which should match Cb. If the confirmation values match, then the connection proceeds.

By virtue of the ECDH key exchange, the Just Works™ pairing method in LE Secure Connections has substantially more resilience to passive eavesdropping compared to the same method in LE Legacy Connections. However, since this method does not give the user a way to verify the authenticity of the connection, it is still vulnerable to MITM attacks.

Out of Band (OOB) Pairing:

In OOB pairing, the public keys, nonces and confirmation values are all exchanged via a different wireless technology such as NFC. As in LE Legacy connections, OOB pairing only provides protection from passive eavesdropping and MITM attacks if the OOB channel is secure.

Passkey:

In this method, an identical 6 digit number is input into each of the devices. The two devices use this passkey, the public keys they exchanged earlier, and a 128 bit nonce to authenticate the connection. This process is done bit by bit for every bit of the passkey. One device will compute a confirmation value for one bit of the passkey and reveal it to the other device. The other device will then compute its own confirmation value for the first bit of its passkey and reveal it to the first device. This process continues until all the bits of the passkey has been exchanged and verified to match.

Due to the process detailed above, the passkey method for LE Secure Connections is much more resilient to MITM attacks than it is in LE Legacy connections.

Numeric Comparison:

This pairing method follows the exact same procedure as the Just Works™ pairing method, but adds another step at the end. Once the devices confirm that the confirmation values match, then both devices will independently generate a final 6 digit confirmation value using both of the nonces. They both then display their calculated values to the user. The user then manually checks that both values match and ok's the connection. This extra step allows this pairing method to provide protection from MITM attacks.

Practical Considerations Concerning BLE Pairing Methods:

A major hurdle to using BLE in secure applications has been that the most secure pairing methods have significant disadvantages in other areas. OOB pairing requires the device to have additional circuitry, raising the cost of the device and the designer must also guarantee that the OOB channel is secure, which can be a significant design challenge in and of itself. Numeric comparison requires each device to have a display, raising device cost, as well as have the user manually verify the codes match, which is detrimental to the user experience. Therefore it is reasonable to assume that most devices will use the passkey method or Just Works™, which means that most devices will have some degree of vulnerability. Designers working on products with high security requirements, such as medical devices, should consider other wireless protocols if OOB pairing or Numberic Comparison cannot be implemented in their designs.

Addressing:

Each BLE device is identified using a device address. These addresses are similar to the MAC addresses used in other communications protocols however, it is usually possible to change BLE device addresses at will. Due to this similarity, it is common to see BLE device addresses referred to as BLE MAC addresses.

BLE currently supports four different types of addresses all of which are 48 bits in length.

- Public IEEE Format- Purchased through the IEEE Registration Authority, these addresses are manufacturer specific. The 24 most significant bits of this address are the Organization Unique Identifier (OUI), aka company ID, and are assigned by the IEEE. The 24 least significant bits are free for the company to modify. Since these addresses do not change, they offer no protection from identity tracking.
- Random Static- These addresses are either burned into the device's silicon during manufacture or generated when the device power cycles. If the device generates a new address every power cycle and the user power cycles the device on a regular basis, then this address type offers some protection from identity tracking. Otherwise, the protection this address type offers is limited.
- Random Private Resolvable- This addressing method can only be used if the Identity Resolving Key (IRK) is exchanged between the two devices during the bonding process. With this method, the device will use the IRK to translate its device address in to a random address that appears in the advertisement packet. A second device that also possesses the IRK is then able to convert the random address back into the real address and identify the first device. In this method, the device will periodically generate a new random address based off the IRK, providing significant protection against identity tracking.
- Random Private Non-Resolvable- In this addressing method, the device address is simply a random number and a new device address can be generated at any time. If new addresses are generated fairly often, then this method offers significant protection against identity tracking.

Conclusion:

BLE offers several features for securing communication between devices, each with its own advantages and limitations. As designers look to implement BLE into their designs, it is important that they understand the specific security threats facing BLE and how BLE's security features help to mitigate them.

References:

Bluetooth Core Specification, ver. 4.1, Bluetooth SIG, December 2013

Bluetooth Core Specification, ver. 4.2, Bluetooth SIG, December 2014

Gibbs, J. (2014, August 13). *Increasing Wireless Security in Bluetooth Low Energy*. Retrieved August 01, 2016, from http://eecatalog.com/IoT/2014/08/13/increasing-wireless-security-with-bluetooth-low-energy/

Rosa, T. (2013, May 23). *Bypassing Passkey Authentication in Bluetooth Low Energy*. Retrieved August 01, 2016, from https://eprint.iacr.org/2013/309.pdf

Ryan, M. (2013). *Bluetooth: With Low Energy Comes Low Security*. Retrieved August 01, 2016, from https://www.usenix.org/conference/woot13/workshop-program/presentation/ryan

Appendix 1, Phase Three Keys and Values:

Long Term Key: If the two devices are bonded, this key is used to encrypt future links so that the pairing process does not have to be repeated.

EDIV and RAND: Values used to create and identify the LTK.

Connection Signature Resolving Key (CSRK): Signs transmitted data and verifies signatures on received data.

Public IEEE Address: See Addressing section above.

Random Static Address: See Addressing section above.

Identity Resolving Key (IRK): Used to resolve the Private Resolvable address.

Appendix 2, Relevant Links Concerning BLE Security:

BLE Core Specification Download Page

Bluetooth Sig's LE Security Page

Appendix 3, Relation Between Pairing Methods and I/O capabilities:

As mentioned in the Pairing method section of this article, certain pairing methods require the devices to have some sort of I/O capabilities. The Bluetooth specification lays out these requirements in the following charts which were taken from: *BLUETOOTH SPECIFICATION Version 4.2 [Vol 3, Part H] 2.3.5 Pairing Algorithms*

	Initiator				
Responder	DisplayOnly	Display YesNo	Keyboard Only	NoInput NoOutput	Keyboard Display
Display Only	Just Works Unauthenti-cated	Just Works Unauthenti-cated	Passkey Entry: responder displays, ini-tiator inputs Authenti-cated	Just Works Unauthenti-cated	Passkey Entry: responder displays, ini-tiator inputs Authenti-cated
		Just Works (For LE Legacy			Passkey Entry (For LE Legacy Pairing):

Display YesNo	Just Works Unauthenticated	Pairing) Unauthenticated	Passkey Entry: responder displays, initiator inputs Authenticated	Just Works Unauthenticated	responder displays, initiator inputs Authenticated
		Numeric Comparison (For LE Secure Connections) Authenticated			Numeric Comparison (For LE Secure Connections) Authenticated

	Initiator				
Responder	**DisplayOnly**	**Display YesNo**	**Keyboard Only**	**NoInput NoOutput**	**Keyboard Display**
Keyboard Only	Passkey Entry: initiator displays, responder inputs Authenticated	Passkey Entry: initiator displays, responder inputs Authenticated	Passkey Entry: initiator and responder inputs Authenticated	Just Works Unauthenticated	Passkey Entry: initiator displays, responder inputs Authenticated
NoInput NoOutput	Just Works Unauthenticated	Just Works Unauthenticated	Just Works Unauthenticated	Just Works Unauthenticated	Just Works Unauthenticated
Keyboard Display	Passkey Entry: initiator displays, responder inputs Authenticated	Passkey Entry (For LE Legacy Pairing): initiator displays, responder inputs Authenticated	Passkey Entry: responder displays, initiator inputs Authenticated	Just Works Unauthenticated	Passkey Entry (For LE Legacy Pairing): initiator displays, responder inputs Authenticated
		Numeric Comparison (For LE Secure Connections) Authenticated			Numeric Comparison (For LE Secure Connections) Authenticated