Experiment 15
Bluetooth Attacks

The goal of this experiment is to attack a Bluetooth device by controlling its behavior through the Bluetooth device on your HAHA Board.

Instructor: Dr. Swarup Bhunia
TAs: Shuo Yang, Reiner Dizon, and Miles F Mulet

Theory Background

Bluetooth is an open standard for short-range radio frequency (RF) communication. Bluetooth wireless technology is used primarily to establish wireless personal area networks (WPANs). Bluetooth has been integrated into many types of business and consumer devices, including cell phones, laptops, automobiles, medical devices, printers, keyboards, mice, headsets, and, more recently, medical devices and personal devices (such as smart watches, music speakers, home appliances, fitness monitors, and trackers). This allows users to form ad hoc networks between a wide variety of devices to transfer voice and data. [2]

Several Bluetooth versions are currently in use in commercial devices. Bluetooth 4.0 (adopted June 2010) is the most prevalent. The most recent versions include Bluetooth 4.1 and Bluetooth 4.2. Bluetooth 4.1 (adopted December 2013) improved the strengths of the Basic Rate/Enhanced Data Rate (BR/EDR) technology cryptographic key, device authentication, and encryption by making use of Federal Information Processing Standard (FIPS)-approved algorithms. Bluetooth 4.2 (adopted December 2014) improved the strength of the low energy technology cryptographic key by making use of FIPS-approved algorithms, and provided means to convert BR/EDR technology keys to low energy technology keys and vice versa.

Bluetooth wireless technology and associated devices are susceptible to general wireless networking threats, such as denial of service (DoS) attacks, eavesdropping, man-in-the-middle (MITM) attacks, message modification, and resource misappropriation. They are also threatened by more specific attacks related to Bluetooth wireless technology that target known vulnerabilities in Bluetooth implementations and specifications. Attacks against improperly secured Bluetooth implementations can provide attackers with unauthorized access to sensitive information and unauthorized use of Bluetooth devices and other systems or networks to which the devices are connected.

1. Bluetooth Security Features

Five basic security services are specified in the Bluetooth standard:

- **Authentication**: verifying the identity of communicating devices based on their Bluetooth address. Bluetooth does not provide native user authentication.
- **Confidentiality**: preventing information compromise caused by eavesdropping by ensuring that only authorized devices can access and view transmitted data.
- **Authorization**: allowing the control of resources by ensuring that a device is authorized to use a service before permitting it to do so.
- **Message Integrity**: verifying that a message sent between two Bluetooth devices has not been altered in transit.

- **Pairing/Bonding**: creating one or more shared secret keys and the storing of these keys for use in subsequent connections in order to form a trusted device pair.

1.1 Security mode and level

Low-Power Bluetooth security mode can have its own security requirements for each service. Low-Power Bluetooth also specifies each service request can have its own security requirements. Equipment to enforce the service following the appropriate security mode and level of security requirements. Low Power Security Mode 1 has a plurality of encryption related level. Level 1 does not specify the security, which means that no authentication and encryption. Level 2 requirements to unauthenticated paired encryption. Level 3 requirements with encrypted authentication. The low power security mode 2 has a signature associated with the data at multiple levels. Data signature provides a powerful data integrity, but not confidentiality. Level 1 requirements unauthenticated data signature matching. Level 2 requires an authenticated signature matching and data. If a service request and related services with different security model and (or) levels, and more powerful security requirements. For example, if any demand need security mode level 3, then Security Mode 1 Level 3 requirements be enforced.

2.Pairing and Link Key Generation

Essential to the authentication and encryption mechanisms provided by Bluetooth is the generation of a secret symmetric key. In Bluetooth version 4.0 and 4.1, pairing is performed using authenticated or unauthenticated procedures. In Bluetooth 4.2, Secure Connections can be used during pairing to authenticate devices.

For PIN/legacy pairing, two Bluetooth devices simultaneously derive link keys when the user(s) enter an identical secret PIN into one or both devices, depending on the configuration and device type. After link key generation is complete, the devices complete pairing by mutually authenticating each other to verify they have the same link key. The PIN code used in Bluetooth pairing can vary between 1 and 16 bytes of binary or, more commonly, alphanumeric characters. The typical four-digit PIN may be sufficient for low-risk situations; a longer PIN (e.g., 8-character alphanumeric) should be used for devices that require a higher level of security.

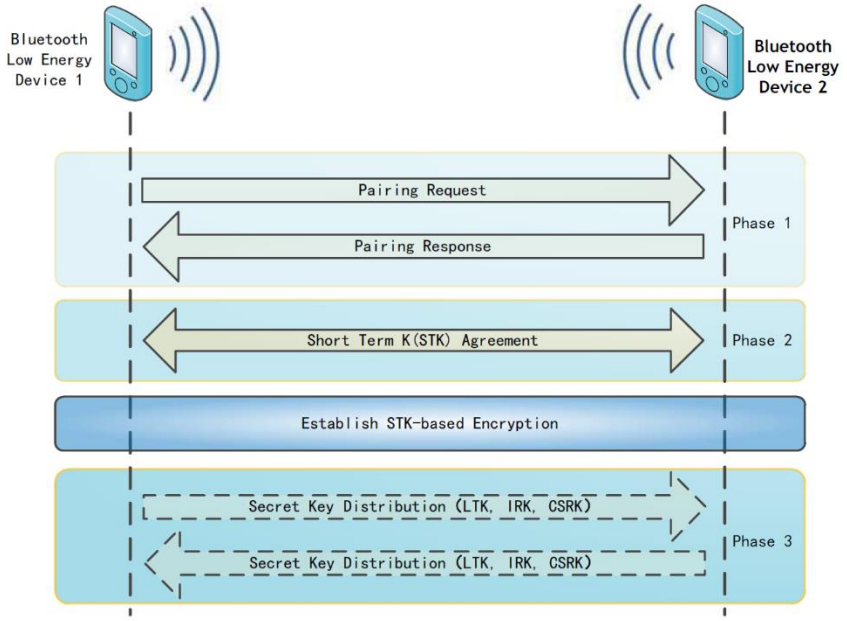

Figure 1 Pairing process of two LP Bluetooth devices.

The four association models offered in SSP (Secure Simple Pairing) are as follows:

- **Numeric Comparison** was designed for the situation where both Bluetooth devices are capable of displaying a six-digit number and allowing a user to enter a “yes” or “no” response.
- **Passkey Entry** was designed for the situation where one Bluetooth device has input capability (e.g., keyboard), while the other device has a display but no input capability.
- **Just Works** was designed for the situation where at least one of the pairing devices has neither a display nor a keyboard for entering digits (e.g., headset).
- **Out of Band** (**OOB**) was designed for devices that support a common additional wireless (e.g., Near Field Communication (NFC)) or wired technology for the purposes of device discovery and cryptographic value exchange.

3.Confidentiality, Authentication and Integrity

AES-CCM in the Low-Power Bluetooth is used to provide confidentiality, and each packet authentication and integrity. No separate authentication challenge / response steps and BR/EDR/HS used to verify whether they have the same LTK or CSRK of. LTK provide implicit authentication is used as input the encryption key, successful encryption settings. The same, although it does not provide confidentiality, but the success of the data signature remote device to provide implicit authentication holds correct CSRK. Versions of Bluetooth technology, in addition to many well-known vulnerabilities such as PIN too short, PIN management deficiencies, encryption key stream recycling vulnerability, but also there are some little-known loophole. These loopholes to bring the convenience of the user, and gave the attacker a shortcut. [3]

- Modes weak protection: “Just Works” associated mode during pairing provides MITM protection, which will result in unauthenticated link key. To obtain the highest level of security, Bluetooth devices in the SSP during MITM protection, and refused to accept the “Just Works” unauthenticated link key pairs generated upon request.
- Weak password generated: SSP ECDH key may be static or weak generation. Weak ECDH key SSP eavesdropping protection minimized, which allow the attacker can determine the secret link key. All equipment shall have a unique, strong generates ECDH key pair.
- Password static: static SSP key to MITM attacks facilitated. Key MITM protection during the SSP, even when you do not need to re-key Bluetooth devices, while still using the last connection key. Bluetooth devices for each pair is connected using the random, unique key. Allowed to fall back to any other security mode? Mode switching vulnerabilities: security mode devices connection does not support security mode 4 Bluetooth devices. Then, the worst case will fall device to return to the security mode, it provides no security authenticated connection.
- Key compromise: connection authentication attempts repeatable. Bluetooth devices need to include a mechanism to prevent unrestricted identity verification request. Bluetooth specification requires that the wait interval between attempts to exponential growth in continuous identity verification. However, it does not require the kind of waiting interval of authentication the suspect requests, so the attacker can collect a large number of suspected response (which the confidential link key encryption), may leak information about the secret link key information.

- Broadcast secret sharing: for broadcast encryption master key is shared between all the micro-network equipment. Secret key shared between two or more of the Parties to provide favorable conditions for the simulated attack.
- The vulnerability: encryption algorithm used for E0 stream cipher algorithm Bluetooth BR/EDR encryption more vulnerable. Bluetooth BR/EDR FIPS approved encryption stratified by application-level FIPS-approved encryption available. User information leaks if the Bluetooth device address (BD ADDR) is captured and associated with a user, privacy may be affected. Once (BD ADDR) associated with a specific user, the user's activities and locations may be tracked.
- Equipment certification attack: device authentication is simple shared key suspect in the response. One-way only suspect that the response to the authentication part of MITM attacks. Bluetooth provides mutual authentication, it should be used as a verification device and network legitimacy.

4.Bluetooth Threats

- **Bluesnarfing**. Bluesnarfing enables attackers to gain access to a Bluetooth-enabled device by exploiting a firmware flaw in older devices. This attack forces a connection to a Bluetooth device, allowing access to data stored on the device including the device's international mobile equipment identity (IMEI). The IMEI is a unique identifier for each device that an attacker could potentially use to route all incoming calls from the user's device to the attacker's device.
- **Bluejacking**. Bluejacking is an attack conducted on Bluetooth-enabled mobile devices, such as cell phones. An attacker initiates bluejacking by sending unsolicited messages to the user of a Bluetooth-enabled device. The actual messages do not cause harm to the user's device, but they may entice the user to respond in some fashion or add the new contact to the device's address book. This message-sending attack resembles spam and phishing attacks conducted against email users. Bluejacking can cause harm when a user initiates a response to a bluejacking message sent with a harmful intent.
- **Bluebugging**. Bluebugging exploits a security flaw in the firmware of some older (circa 2004) Bluetooth devices to gain access to the device and its commands. This attack uses the commands of the device without informing the user, allowing the attacker to access data, place phone calls, eavesdrop on phone calls, send messages, and exploit other services or features offered by the device.
- **Car Whisperer**. Car Whisperer34 is a software tool developed by European security researchers that exploits the use of a standard (non-random) passkey in hands-free Bluetooth car kits installed in automobiles. The Car Whisperer software allows an attacker to send to or receive audio from the car kit. An attacker could transmit audio to the car's speakers or receive audio (eavesdrop) from the microphone in the car.
- **Denial of Service**. Like other wireless technologies, Bluetooth is susceptible to DoS attacks. Impacts include making a device's Bluetooth interface unusable and draining the device's battery. These types of attacks are not significant and, because of the proximity required for Bluetooth use, can usually be easily averted by simply moving out of range.

- **Fuzzing Attacks**. Bluetooth fuzzing attacks consist of sending malformed or otherwise non-standard data to a device's Bluetooth radio and observing how the device reacts. If a device's operation is slowed or stopped by these attacks, a serious vulnerability potentially exists in the protocol stack.
- **Pairing Eavesdropping**. PIN/Legacy Pairing (Bluetooth 2.0 and earlier) and low energy Legacy Pairing are susceptible to eavesdropping attacks. The successful eavesdropper who collects all pairing frames can determine the secret key(s) given sufficient time, which allows trusted device impersonation and active/passive data decryption.
- **Secure Simple Pairing Attacks**. A number of techniques can force a remote device to use Just Works SSP and then exploit its lack of MITM protection (e.g., the attack device claims that it has no input/output capabilities). Further, fixed passkeys could allow an attacker to perform MITM attacks as well.

Experiment Set-up: Configuration

The hardware needed for this experiment include:

1. The HAHA Board and USB cable.
2. Your own laptop.
3. Your own mobile phone.

The software needed for this experiment include (install them in your own laptop):

1. Driver for the USB-to-UART serial converter MCP2221. [8]
2. Tera Term. [9]
3. TTLEditor. [10]
4. SmartData. It can be downloaded from Apple App store and Google App store.

Instructions and Questions

PART I Get started with the HAHA-BLE module

1. Connect the Bluetooth module (Microchip RN4870) to your laptop
   a) Install all the software listed in the previous page.
   b) Connect the HAHA Board to your laptop with a USB cable. Use the USB connecter J18 on the board. Turn on the power of the board.
   c) Make sure the switch S11 is on position “App”, and switch S10 is off.
   d) Press SW4, and the LED D14 should flash slowly.
   e) Use the TeraTerm softeware. Configure the serial port settings using a Baud rate of 115200.
   f) Enter Command mode by sending the command escape sequence $$$.
   g) Use commands in the table to finish the questions below.
      1) What’s the MAC address of your RN4870? What’s the original device name of your RN4870? Take a screenshot.
      2) Change the device name to HAHA_Groupxx. “xx” is your group number. Display the revised device information and take a screenshot.

ASCII Command	Description
$$$	Get into command mode
+	Echo on/off
D	Display critical information
S-,<string>	Set serialized device name

2. Connect the RN4870 to your mobile phone
   a) On your mobile phone, download and install SmartData by Microchip.
   b) Open a serial port terminal.
   c) Enter command mode ($$$). Enble echo (+).
   d) Enable UART transparent service (SS,C0).
   e) Reboot (R,1) for the configuration to take effect. (You don’t need to repeat d and e every time.)
   f) On your phone, turn on Bluetooth and open SmartData.
   g) Connect your phone with the RN4870. Answer the following questions.
      1) Send anything you want, including your group number, from the phone to the BLE module. Take a screenshot.
      2) Send anything you want, including your group number, from the BLE module to the phone. Take a screenshot.

PART II Hack into others' Bluetooth module

1. Connect your Bluetooth module to another one

Your TA will turn on a HAHA Board act as the target for you to attack. The MAC address for the TA's Bluetooth module is unknown to all. You should scan and find the correct MAC address, get connected, and send data to the target. The commands you may need are listed in the table below.

ASCII Command	Description
F	Start scanning
X	Stop scan
C	Connect to last bonded device
C,0,<address>	Connect to public <address>
K,1	Disconnect
B	Start bonding process

1) What is the MAC address of the target Bluetooth module?
2) After you get connected, send some words including your group number to the target. TA will record what is received and which groups succeed. The groups that finish this faster will get higher scores. After you finish sending data, disconnected immediately so that other groups can attack. Take a screenshot.

2. Brute-force the PIN and control the behavior of another Bluetooth module

Your TA will turn on the same HAHA Board for you to attack. The target's MAC address has been known. However, in order to have total control over it, you should have the correct 4-digit security pin code, which is unknown to you. You should guess from the 10^4 possibilities. If your guess is correct, after you get connected, you will be able to enter the Remote Command Mode. In this mode, change the IO status of the target so that your group number can be seen on the 7-segment display of the target board. The commands you may need are listed in the table below.

ASCII Command	Description
SP,<4 digit pin>	Set the pin code
!,1	Turn on remote command mode
!,0	Turn off remote command mode
|O,<hex8>,<hex8>	Set the output value of IO ports

It's not efficient to try every possible combination one after another by hand typing. Therefore, Use "Macro" to make the brute-force run automatically. You will write a very easy TTL script and run it in TeraTerm. In your script, there will be a loop repeating guessing the correct PIN. For every guessed PIN, try to enter the remote command mode. If the guessed PIN is incorrect, the connection between your Bluetooth and the

target will be disconnected automatically. The Macro language Tera Term Language (TTL) is friendly to beginners. You will need the commands listed in the table below to finish this task. Other commands can be found in [11].

TTL Command	Description	Example
send	Send data.	send 36 ; send “$” (ASCII code is 36)
sendln	Send data with new-line.	Sendln ’!,1’ ;send “!,1”
pause	Pause.	pause 10 ; pause for 10 seconds
int2str	Convert integer value to a string.	int2str key 1234 ; key will be a string “1234”
wait		

1) What is the 4-digit PIN code for the target?
2) Turn in your TTL macro script.
3) TA will record the behavior of 7-segment display of the target board to see which groups succeed in this attack. The faster a group finish this attack, the higher score the group will get.

References and Further Reading

[1] Oka, Dennis Kengo, et al. "Survey of vehicle IoT bluetooth devices." Service-Oriented Computing and Applications (SOCA), 2014 IEEE 7th International Conference on. IEEE, 2014.

[2] Padgette, John. "Guide to bluetooth security." NIST Special Publication 800 (2017): 121.

[3] Xu, Junfeng, et al. "Pairing and authentication security technologies in low-power Bluetooth." Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing. IEEE, 2013.

[4] https://threatpost.com/bluetooth-hack-leaves-many-smart-locks-iot-devices-vulnerable/119825/

[5] https://www.slideshare.net/CysinfoCommunity/attacking-and-crashing-iot-devices-via-bluetooth-le-protocol

[6] http://ww1.microchip.com/downloads/en/DeviceDoc/50002489A.pdf

[7] http://ww1.microchip.com/downloads/en/DeviceDoc/50002466A.pdf

[8] http://www.microchip.com/wwwproducts/en/MCP2221

[9] http://logmett.com/tera-term-the-latest-version

[10] http://logmett.com/ttleditor

[11] https://ttssh2.osdn.jp/manual/en/macro/syntax/index.html