A SECURITY RISKS ANALYSIS OF CONSUMER BLUETOOTH LOW ENERGY DEVICES

b09902003 陳宇柔 b09902009 劉峻瑋 b09902017 李安傑 b09902135 賴豐彰

{b09902003, b09902009, b09902017, b09902135}@csie.ntu.edu.tw

Abstract

Used by mobile phones, game controllers, personal computers, etc, Bluetooth technology has become an important part of this modern society. As Bluetooth technology gains its popularity and becomes widely used by users, security vulnerabilities are also increasing, hazarding the privacy of users' personal information. In this paper, we set commercial Bluetooth devices as our target, aiming to test whether these devices still contain the vulnerabilities addressed in previous research works or have been updated after the problems were fixed in more recent versions. After understanding the protocols used in different versions of Bluetooth, we explored the vulnerabilities in each security protocol and tried to conduct attacks based on the devices we have on hand. With the results of our experiments and the possible further attacks we have in mind, we conclude that the tested devices have not secured sessions keys appropriately during the key exchange, making them vulnerable to eavesdropping. We also conduct a proof of concept to demonstrate the impact of physical conditions to actual Bluetooth attacks. At the end, we propose some mitigation users can adopt to enhance security and protect their privacy.

1. INTRODUCTION

Nowadays, Bluetooth technology has been widely used for connecting electronic devices, considered to be cheap, reliable, and power efficient. Offering two radio options, the Bluetooth Classic and the Bluetooth Low Energy (LE), Bluetooth technology provides developers with a versatile set of solutions to meet the ever-expanding needs for wireless connectivity and to fit for different purposes. Designed to meet the unique needs of developers around the world, the Bluetooth LE and Bluetooth Classic radios support products streaming high-quality audio between smartphones and speakers, transferring data between devices, or sending messages between nodes, etc.

As the Bluetooth technology gains its popularity, some researchers indicate in their work that there exists some security vulnerabilities and some attacks against this technology have been proposed. Meanwhile, the recent versions of Bluetooth also came up with more advanced security protections and measures, trying to enhance its security. In this work, we aim to conduct a survey to find out to what extent do products that are available on the market still fall vulnerable to the existing attacks.

Instead of the Bluetooth Classic that has existed from the beginning, we focus our discussion on the Bluetooth Low Energy (LE) in this work for the following reasons. Firstly, as the name suggests, Bluetooth LE radio is designed for very low power operation, which can provide users with longer operation time for battery-powered products. Secondly, with over 40 channels in the 2.4GHz ISM frequency band for data transmission, the Bluetooth LE radio provides developers with flexibility to build products requiring unique connectivity of their market. Lastly, apart from its device communications capabilities supporting large-scale device networks, Bluetooth LE is also used for indoor location services, allowing devices to determine the presence, distance, and direction of another. Because of the wide use of Bluetooth LE today, we decide to focus our study on this promising technology.

In this paper, we first briefly introduce the security features of Bluetooth in Section 2. Then, in Section 3 and Section 4, we explain in detail the security mechanisms used, address the vulnerabilities, and describe the experiments we conducted, for BLE 4.0-4.1 and BLE 4.2-5.1, respectively. In Section 5, to test the impact of physical conditions, we discuss the effective distance that makes Bluetooth attacks possible. In Section 6, we provide some mitigation that people could adopt to help reduce the chances of coming under Bluetooth attacks. Last, we will summarize related works in Section 7 and give conclusion and future works in Section 8.

2. BACKGROUND

Embedded in the fabric of our lives, connecting us to one another, Bluetooth technology indeed has a huge impact in our lives, making the task of securing the connection extremely essential. The security properties of a BLE connection are defined primarily through the selected se-

curity mode, security level and the used pairing method. BLE distinguish two security modes, subdivided into several security levels. In security mode 1 the connection is encrypted and authenticated using AES128-CCM in all levels but level 1, which provides no security. In security mode 2 the connection remains unencrypted, but data is signed on a higher layer of the Bluetooth protocol using a special key in order to verify the authenticity of the data. Each connection will select a security mode, a security level, and a pairing method. We focus on the weaknesses in pairing protocols.

In Bluetooth connection, a trusted relationship needs to be established between two devices, called "pairing". The process is done by exchanging shared secret codes, PINs, where a master device can pair with one or more slave devices. More precisely, the initiating device first sends a pairing request to another device, then the two devices would exchange capabilities, determining how to set up a secure connection. Then, a temporary key, TK, would be generated, which, along with the random value, would later be verified and confirmed by the two devices. Then, a shared secret would be created using the TK and the random values, and would be used to encrypt the connection.

Bluetooth Low Energy was first introduced in version 4.0. The pairing process of BLE 4.0 and 4.1 is called BLE Legacy, where the TK is used to create a short term key, which is later used to create the long term key. In BLE Legacy, three methods are used to share the TK, Just works (TK set to 0), Out of Band (TK up to 128 bits), Passkey (TK being a 6-digit number).

Starting from Bluetooth 4.2, a different pairing process is used, called LE secure connections pairing (LE-SC), where elliptic curve Diffie-Hellman (ECDH) is used for key exchange. Similar to BLE Legacy, LE-SC also supports multiple methods to share the TK. In addition to the three supported in BLE Legacy, there is a new one called Numeric Comparison in LE-SC, which requires both devices to have a screen to display a 6-digit number.

3. ANALYSIS OF BLE 4.0-4.1

The communication of BLE 4.0-4.1 is protected by AES-CCM mode encryption, which is an encryption algorithm widely accepted. Hence, we believe that after the master device and the slave device exchange the secret key, the transmission process should be secure. The main security issue comes from the fact that the key exchange is not encrypted, resulting in the risks of losing the key of AES to an eavesdropper. Once the eavesdropper gets the key used in AES, he can get the messages and contents of the communication easily, and can also use the key to add fake packets to the communication.

Fig. 1. The setting of our experiments.

No.	Time	Source	Destination	Protocol	Length	Info
2064	23.072771	Master_0x692e9467	Slave_0x692e9467	SMP	47	Sent Pairing Confirm
2065	23.073169	Slave_0x692e9467	Master_0x692e9467	LE LL	26	Empty PDU
2066	23.079811	Master_0x692e9467	Slave_0x692e9467	LE LL	26	Empty PDU
2067	23.080040	Slave_0x692e9467	Master_0x692e9467	SMP	47	Rcvd Pairing Confirm
2068	23.087311	Master_0x692e9467	Slave_0x692e9467	SMP	47	Sent Pairing Random
2069	23.087709	Slave_0x692e9467	Master_0x692e9467	LE LL	26	Empty PDU
2070	23.102311	Master_0x692e9467	Slave_0x692e9467	LE LL	26	Empty PDU
2071	23.102541	Slave_0x692e9467	Master_0x692e9467	SMP	47	Rcvd Pairing Random
2072	23.109812	Master_0x692e9467	Slave_0x692e9467	LE LL	49	Control Opcode: LL_ENC_REQ
2073	23.110225	Slave_0x692e9467	Master_0x692e9467	LE LL	26	Empty PDU
2074	23.117312	Master_0x692e9467	Slave_0x692e9467	LE LL	26	Empty PDU
2075	23.117542	Slave_0x692e9467	Master_0x692e9467	LE LL	39	Control Opcode: LL_ENC_RSP

Fig. 2. The packets captured by the sniffer.

There are three pairing methods in BLE Legacy, Just Works, Out of Band, 6-digits Passkey. Out of Band uses wireless techniques other than Bluetooth to conduct the key exchange, having its security level based on the security of the technique used, and thus is not the target of our study. Just Works simply set TK to 0, allowing passive attackers to compute the short term key once getting information such as the random values, addresses of the two devices, further breaking the encryption. As for Passkey, although the TK is set to a 6-digit passkey, attackers can still easily confirm their guesses with the computation of 10^6 hash values. After confirming the TK, they can further compute the short term key.

To check if this problem in the pairing process still exists, we bought a Bluetooth sniffer to listen to a mouse with BLE Legacy Just Works. We use a Bluetooth mouse(MSI M98, using BLE Legacy for pairing) to draw a graph on the laptop, and then listen to the Bluetooth communication.

After recording the communication, we exported the file as a JSON file, and then used Python to reproduce the graph, below is our result.

Although encrypted with AES, BLE is still insecure do the the lack of encryption mechanisms in the pairing process. Attackers can get the original contents of communication by eavesdropping on the pairng process.

4. ANALYSIS OF BLE 4.2-5.1

BLE 4.2-5.1 is not much different from the old versions, still using AES-CCM mode encryption to protect the communication. The main different is that the new version of BLE uses ECDH for key exchange, and added Numeric Comparison method for the pairing process. Us-

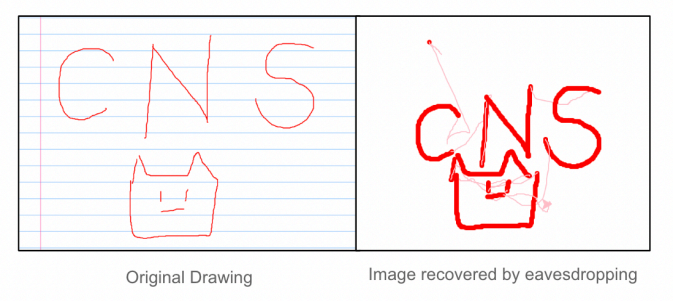

Fig. 3. The track of the Bluetooth mouse is eavesdropped by successfully recovering the key.

ing ECDH in key exchange solves the problem of passive eavesdroppers. In the pairing method Just Works, an eavesdropper can only see what the two devices exchanged, but cannot compute the shared secret, thus unable to decrypt the communication. However, the Just Works pairing method still lack the mechanism to check the public key, making it hard to detect man-in-the-middle attack and giving the attackers the opportunities to eavesdrop or modify the contents. To solve this problem, the 6-digits Passkey, numeric comparison in the pairing process can help the two devices make sure the received public key and the generated shared secret are coherent. But this type of communication is limited to the connection where both devices is capable of entering numbers or displaying numbers, which is unfeasible to many Bluetooth devices, such as keyboards, mice, smart home devices, etc. To verify the mechanisms in the new versions of BLE, we looked for some devices using BLE Security Connection. The device we used is Logi mouse M650 (using BLE 5.1), in which Just Works is used as the pairing method. We used the same method as the previous lecture to eavesdrop on the signals, and verified that this kind of communication actually uses ECDH to exchange keys, and cannot be decrypted easily.

4.1. Downgrade

To achieve downgrade attack, we first need to perform MitM, and use blocker, an adapter using the same frequency but stronger signals, to block the connection between the host and the device. Then, we would make the host pair to a fake device we set up by spoofing. Taking advantage of the fact that error messages do not appear on the host when using certain versions of Bluetooth, we can send error "Send Pin or Key Missing" to achieve downgrade attack. In this case, the contents of the communication would be transmitted in plain texts, and thus allow attackers to obtain some information and send unusual data to the host. To perform more attacks, we can make use of Bluetooth's inability to detect data inconsistency to break the connection between the fake device and the host, and then make the host connect back to the victim device, causing a deadlock in the Bluetooth connection due to data inconsistency and thus leading to denial of service. In addition, we can also send the MAC address and IRK (Identity Resolution Key, used to identify the paired device) to our fake mobile, and thus bypass the whitelist of the victim device.

Distance	Success Rate	Result
< 0.1 m	4/4	Can easily get the STK
1m	3/4	Sometimes lose the packet
3m	1/4	Sometimes success
10m	0/4	Too far to get signal

Table 1. Success Rate of recovering keys w.r.t. distance.

5. EFFECTIVE DISTANCE FOR BLUETOOTH ATTACKS

5.1. Attack with Different Distances and Physical Barrier

In the section 3, we discussed how a passive attacker can recover the message using the encryption key by simply eavesdropping on the pairing process. Now we want to further discuss under what physical conditions can the attackers get the signals and recover the key. To discuss the effective distance for Bluetooth attacks, we tried to perform the eavesdropping experiment we did in Section 3 in four different distances between the laptop and the mouse, and repeatedly testing if our sniffer can successfully eavesdrop the pairing process and then calculate the STK used in the encrypted communication. We did this experiment four times for each distance, and got the following results.

We found that with the distance in about 1 meter, the sniffer can easily obtain the STK. However, when the distance up to 3 meters, it becomes harder for the sniffer to eavesdrop without losing any important packet transmitted in the pairing process, thus making it almost impossible for sniffers to listen to the traffic at a distance of more than 10 meters from the pairing devices.

We later discovered that our mouse actually still works within the distance of 10 meters. We believe that the effective distance for Bluetooth communication is in fact not shorter than this, but our sniffer's ability to receive packets is not as strong. It is possible that by adding antennas, the range of which the sniffer can receive signals would increase, and thus results in larger attacking zone.

Aside from the distance factor, we also tested on the impact of the wall's separation. As a result, it does not make much difference. Bluetooth transmits data in the

2.4GHz ISM frequency band, which is not too high that it can work through physical barriers.

5.2. Attack on Packets with Errors

While conducting the experiments, we noticed that sometimes due to the distance or physical barriers, the content of a packet captured by our sniffer may be wrong, which leads to the consequence that the shared secret cannot be computed correctly. However, during the pairing process, the two devices would exchange a value to confirm, making sure the random value used to generate the secrets is coherent. We can make use of this confirm value to perform a brute-force search to obtain the pairing random value and recover the short term key. Here we designed a simple program that can recover an error byte in the packet, and can be extended to recover multiple bytes.

Algorithm 1: One-Byte Correction

```
Function correct_one_byte:
    k ← key; r ← random (in bytes); n ← info;
    if c1(k, r, n) ≠ confirm then
        for i = 0 to 15 do
            for b = 0 to 255 do
                r′ ← r[: i] + byte(b) + r[i :];
                if c1(k, r′, n) = confirm then
                    return r;
                end
            end
        end
    end
    return r;
```

The function "c1" is the function in BLE document[1] released by the Bluetooth company, which is used to calculate *confirm* for the random values of either the master device or the slave device. *Confirm* is a combination of many different arguments from the pairing process during the communication, and the key used is the TK mentioned in the previous sessions. This is a simple brute-force algorithm, and can be easily extended to search for any number of bits.

5.3. Defence Proposal Based on the Effective Distance

The pairing process is the most sensitive part of the BLE communication. Luckily, because of the Trust-On-First-Use property, once we can securely connect our devices at the first time, the connection can be protected by the AES encryption in all the future use. Based on this fact, it occurred to us that the manufacturers could develop a low-intensity signal mode for the pairing process, which can effectively shrink the attack range of the eavesdropping attack. If the distance in which the attackers can access the signal during the pairing process can be restricted to 3 meters or less, the users can pair their devices in a safer place, such as their own houses, to make the connection attack-free. When transmitting other packets, the device can go back to using signals of normal intensity to maintain its availability for communication from further distances.

6. DEVICES ON THE MARKET

When we were looking for suitable devices to be our tested devices, we noticed that most of BLE devices on the market neither show which BLE version the devices use nor whether the Bluetooth firmware can be updated. The only way for usual customer like us to confirm the version is to sniff packets via BLE dongle and compare the formats of the packets with pairing protocols of each version.

In this research, we tested four BLE devices in total: Logitech M590 bluetooth mouse, MSI M98 bluetooth mouth, Logitech M650 bluetooth mouse, and Xiaomi Yeelock bluetooth lock. Among the four devices, only Logi devices have formal product manual, giving information regarding bluetooth connection. M590 uses Logi Unifying and M650 uses Logi Bolt. Only the recently published Logi Bolt has a public specification with security-related information, showing that Logitech M650 uses security mode 1, level 2 when via bluetooth. The BLE version and pairing method of other three devices use is confirmed by sniffed packets. M650 use BLE 4.2 with ECDH as method for key exchange. M590 and M98 uses BLE Legacy with Just Work TM. Yeelock uses BLE 5.1.

7. RELATED WORKS

In this section we review how we gain knowledge about BLE technology. We started from exploring the history and the current development of BLE technology, then identifying vulnerabilities of security protocol in each BLE version, and at last studying some particular attacks and security issues on some specfic BLE devices.

To begin with, Gomez et al.[2] described the main features of BLE and investigated the impact of various parameters on its communication performance. They also briefly mentioned security issues such as the possibility of eavesdropping during key exchange in BLE 4.0 and below due to non-encrypted key. We then found that Bon [3] wrote an short but concise introduction on BLE 4.x security. He focused on how pairing method is designed and their corresponding flaws, including Just Work TM,

Out of Band (OOB) Pairing, Passkey, and BLE 4.2 and above Numeric Comparison. This article gave us some insights on what types of attacks to which each pairing methods are subjected and session key might be recovered. It made us decide to focus this project on pairing security. Given lots of major and minor reversion on paring protocols, Cäsar et al.[4] gave an systematic overview that covers the security and privacy in different specifications with introduction of each version and corresponding vulnerability. Their detailed survey on listing previous works on known weakness and attacks of different types (sniffing BLE, MitM attacks, weakness in pairing protocols, ... etc.) have been a great guide for this project.

Next we surveyed on what kinds of BLE devices could be our potential targets for experiments. Lonzetta et al.[5] presents an overview of Bluetooth in IoT (Internet of Things) on its security, vulnerabilities, and real-life examples of exploits. Some examples of commercial products includes automobiles, medical devices, smartwatch and smart bracelet, smartphones, smart homes. Among all devices, we chose Bluetooth mice as our target device because it has small numbers of possible operations, is simple to pair and disconnect process, and is under our budgets. Although we did not notice any research that performs attacks on some specific Bluetooth mice, we found that Klostermeier and Deeg [6] had a research project on analyzing three popular Blutooth keyboards (1byone Keyboard, Logitech K480, and Microsoft Designer Bluetooth Desktop, Model 1678). The project showed that secret keys are stored on the keyboards and can be easily extracted by an attacker with physical access. The 1byone keyboard does not require authentication when pairing to a Windows 10 host, and the communication of the Microsoft Designer Bluetooth keyboard can be decrypted if an attacker passively eavesdrops on the pairing process. In case of our Bluetooth mice, as we do not have enough skills to break down a mouse and see whether some secret pairing information are hard-coded, we decided to recover keys by passively eavesdropping on the pairing process using a well-implemented sniffer.

Last but no least, we research on how BLE devices can be attacked. Ruge et al.[7] emulated the firmware of Bluetooth-5.2-standards chips with qemu then fuzz them. During the greybox fuzz, the authors discovered that there are many heap-overflow issues for many chips especially Broadcom' s. They also found that there is a auto-pairing mechanism, which attackers can easily perform a zero-click attack accompanied with the not-reset memory. With these flaws, the authors can easily reach a RCE(remote code execution), which is a serious attack.

8. CONCLUSION

In this project we conducted an analysis on security vulnerability of consumer products that use BLE 4.0-4.1 and BLE 4.2-5.1. We identified security flaws in two BLE mice and successfully recovered the encryption keys by eavesdropping the pairing process. We showed that by performing the attack, the track of the mouse can recover in real-time by the attacker. To further test the physical conditions on how attackers can sniff pairing packets and perform the attack, we conducted an experiment to test the impact of distance and wall's separation on whether attackers can sniff and recover the packets. We also comment on the publicity of security information of consumer products, and give some mitigation to enhance the security of Bluetooth in daily usage. Different from related works, this project focus on the analysis of the conditions of real-world attacks to be performed on actual commercial products, instead of presenting a theoretical attack. All codes are available at `https://github.com/anj226/BLE_mouse_path_recovering`.

9. REFERENCES

[1] Bluetooth SIG Proprietary, "Specification of the bluetooth® system core specification 4.2," 2014.

[2] Carles Gomez, Joaquim Oller, and Josep Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," *Sensors*, vol. 12, no. 9, pp. 11734–11753, 2012.

[3] Matthew Bon, "A basic introduction to ble 4.x security," 2016.

[4] Matthias Cäsar, Tobias Pawelke, Jan Steffan, and Gabriel Terhorst, "A survey on bluetooth low energy security and privacy," *Computer Networks*, vol. 205, pp. 108712, 2022.

[5] Angela M. Lonzetta, Peter Cope, Joseph Campbell, Bassam J. Mohd, and Thaier Hayajneh, "Security vulnerabilities in bluetooth technology as used in iot," *Journal of Sensor and Actuator Networks*, vol. 7, no. 3, 2018.

[6] Matthias Deeg Gerhard Klostermeier, "Case study: Security of modern bluetooth keyboards," *SySS IT Security Research Project*, 2018.

[7] Jan Ruge, Jiska Classen, Francesco Gringoli, and Matthias Hollick, "Frankenstein: Advanced wireless fuzzing to exploit new bluetooth escalation targets," in *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020*, Srdjan Capkun and Franziska Roesner, Eds. 2020, pp. 19–36, USENIX Association.