Hacking Electric Scooters with Bluetooth Low Energy

Peder Grundvold
pederg@stud.ntnu.no
TTM4137 Wireless Security Technical Essay

May 25, 2021

1 Introduction

In the last couple of years, electric scooters have had a massive increase in popularity. Companies like Bird, Lime, Voi, and several others are fighting for market shares all around the world. Many of the e-scooters rely on the Bluetooth low energy (BLE) communication standard. This standard has proven to have many security issues, including unsecure pairing, the possibility for passive eavesdropping and man-in-the-middle (MITM) attacks[14][13]. Lately, different security researchers have demonstrated several exploitations of these weaknesses[6][3][10]. This means that malicious actors can steal personal information from users, and in some cases also modify the behavior of the e-scooter while it is in use. Hackers can, in other words, physically harm users by removing brakes or accelerating the e-scooter in the wrong situation.

In this essay we will focus on the weaknesses in the BLE communication protocol that make these types of attacks possible. We will also give a brief overview of which e-scooter brands that might be vulnerable to this type of attack, and finally give a real life example of such an attack on a popular e-scooter model.

2 Background

2.1 What is Bluetooth Low Energy (BLE)

Bluetooth is a widely used technology standard for exchanging data between different devices. The "normal" version of Bluetooth, Bluetooth Classic, is maybe what most people are familiar with. This is used when connecting a wireless headset to your phone or a wireless keyboard to your computer. There is; however, another version that is not that well known.

Bluetooth Low Energy is used for periodic transmission of smaller packets and has a somewhat different use case than Bluetooth Classic. In the BLE topology there are two types of devices; central and peripheral. Central devices are similar to what we also would use in Bluetooth Classic, like phones, laptops, etc. Peripheral devices are often less complicated and will act as sensors. They send data to the central device and will not do any processing by themself. Examples of such devices can be pacemakers, fitness trackers and industrial sensors. When not sending, the BLE connection on these devices will remain in sleep mode and therefore this standard significantly reduces the overall energy consumption[4][2].

2.2 BLE pairing process

The most vulnerable part of the BLE standard is the pairing process, and therefore it is here a typical attack will occur. Bluetooth Low Energy has two different connections types; Secure and Legacy. Secure connection was introduced in Bluetooth 4.2 and implements better authentication and encryption with a more sophisticated cryptographic method using Elliptic-

curve Diffie-Helman. Because this obviously makes attacks more difficult, and Legacy is still widely used, we will not discuss Secure connection any further here.

With BLE it is up to the developer to choose what security level a product should implement. Figure 1 below shows the LE Legacy Pairing process with the most basic method, called Just Works. In this simple method the Temporal Key (TK) is set to zero and therefore the devices does not need any pre-shared knowledge.

First, both devices use the TK, a random value and some capability values to compute what is called confirming values. The function used is named Confirm Value Generation Function, or simply c1. Then each device sends their random value so that both devices can verify that the values match. If they match, the TK and both random values will be used to generate a 128 bit long Short Term Key (STK) that is then used to encrypt any further communication.

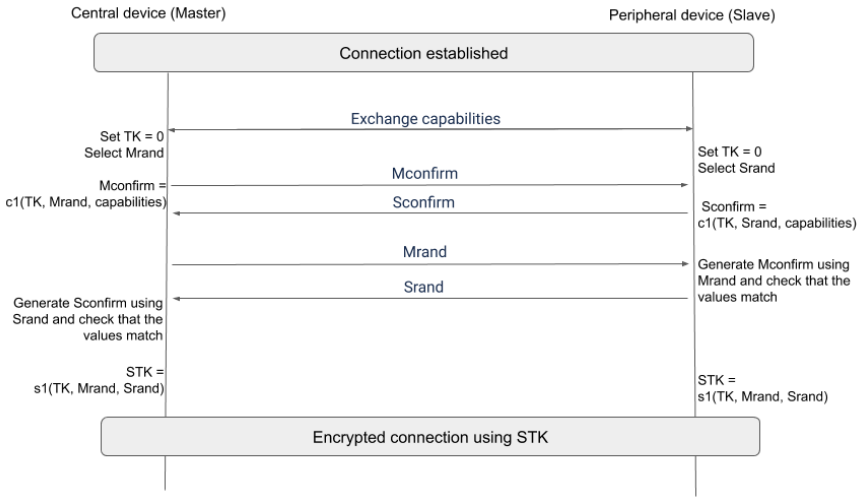

Figure 1: BLE Just Works pairing process

The obvious security flaw here is that the communication before generation of the Short Term Key (STK) is unencrypted. An attacker simply needs to listen to the traffic during setup and can then easily generate the STK using Mrand and Srand. In the other pairing methods, Passkey and Out of Band (OOB), TK is a 6-digit input from the user.[8].

3 Problem Discussion

3.1 BLE in e-Scooters

Scooter-sharing companies all follow the same simple usage model. A user downloads the app made by the e-scooter company and can then unlock any scooter from that company by simply scanning a QR code. Since the launch of the first scooter-share companies in California in 2017, the number of competing companies have just increased[5]. By 2020 here in Norway we have six different companies: Voi, Tier, Lime, Bolt, Ryde and Wind. But not all of them require Bluetooth. A quick search on Google Play and App Store tells us that the scooters used by Bolt and Voi require no Bluetooth permissions, but the rest require both the “access Bluetooth settings”- and the “pair with Bluetooth devices”-permission. Further testing shows that it is still possible to connect with BLE to Voi and Bolt even though their corresponding apps do not require this to work.

As mentioned earlier one of the principles with BLE is that authentication and encryption methods are decided by the developers. This gives great versatility because the level of security a device needs is not necessarily equal in all situations where BLE is used. The above mentioned ride-share companies do not manufacture their own scooters. They are bought from third-party producers and then rebranded with the company’s colors and logo. Because of the BLE standard’s principle of self-determinism it is not certain that the products bought have adequate security. This was the case with the Chinese produced scooter Xiaomi M365. A big American

scooter-share company named Bird [9] used this model in their fleet, and it has later been discovered that it had a serious security vulnerability which we will discuss now.

3.2 Real-life example - Xiaomi M365

In February 2019, the mobile security company Zimperium published a blog post revealing a weakness in the e-scooter Xiaomi M365[6]. Later there have been several other publications about this weakness[3][10]. As mentioned before, this e-scooter has been part of the fleet for different rideshare companies. It is currently not used by any of the rideshare companies operating in Norway, but is available for sale in several Norwegian electronic stores[11][12].

Let us then dive into this vulnerability. The Xiaomi e-scooter comes with a dedicated app that enables users to modify different features like Eco Mode or Cruise-Control. To do this the e-scooter's firmware is updated over a Bluetooth Low Energy connection initiated from the user's app. Access to this is protected by a password. Because of this, looking back at the BLE pairing process, we see that in this case, the cryptographically flawed method Just Works is not used, and it should therefore be considered safe? However, this was not the case.

While initiating a connection using the dedicated app from Xiaomi was secured with a password, initiating a BLE connection from another source was not. This can easily be achieved, for example by using the network security tool Bettercap[1]. With this connection it is possible to overwrite the firmware with custom values. That is; without being limited to the functions defined in the app. However, to do this the attackers have to write in bytes, and without knowing the exact bytestring for each function this would be close to impossible. One solution for the attacker is to decompile the entire Xiaomi app, but there is an easier way. The attackers can simply download the app and perform a Man-in-the-Middle (MITM) attack on themself when connecting to the e-scooter. While connected, unlock the e-scooter from the app and listen to the data sent between the app and the e-scooter over BLE. Now the attackers can go to any Xiaomi M365 e-scooter, connect to it (with Bettercap i.a.), send the same bytestring as was seen during the MITM attack and the e-scooter will be unlocked.

Zimperium actually also decompled the app and was able to find the corresponding byte strings for the brake- and gas command. They then made a proof of concept (PoC) app that could be used to hijack Xiaomi M365 scooters and run those commands over a BLE connection. Due to safety concerns that PoC is not available, but the one for unlocking is available here[7].

4 Conclusion

Electric scooter ride-sharing is an ever growing market. The competition on price and market shares is fierce and companies are frequently started and shut down. Therefore, when these companies are buying e-scooters from third-party producers, security is maybe not their highest priority. Combine this with the fact that all of these e-scooters use Bluetooth Low Energy to some degree or at least allow for some kind of BLE connection. We have seen that BLE can be a very insecure protocol if not implemented correctly. Can we then be sure of, as opposed to the case with Xiaomi, if all of these companies and their suppliers implement adequate security in their solutions?

Furthermore, regarding Xiaomi M365, a model which is still for sale in Norway; how long was this vulnerability present, or is it maybe still there? A paper by Louis C. Booth et.al. says: *"Despite our scooter arriving several months after the exploit being released we were able to reproduce the hack in the exact way shown by Zimperium"*[3].

With all of these known issues, the lack of focus on e-scooter security is truly worrying. The public should be better informed about the risks, and companies has to be held more responsible if something happens. In conclusion, as a user keep this in mind; be careful with what ride-sharing companies you trust. At least, I know I will.

References

[1] *bettercap*. https://github.com/bettercap/bettercap.

[2] *BLE Security Fundamentals*. http://dev.ti.com/tirex/explore/node?node=AOPOY.GDApakIOYjiwoY6A__pTTHBmu__LATEST. Accessed: 2020-10-12.

[3] Louis Cameron Booth and Matay Mayrany. *IoT Penetration Testing: Hacking an Electric Scooter*. https://www.kth.se/polopoly_fs/1.914061.1600688684!/Louis_and_Matay_final_scooter.pdf. Accessed: 2020-10-12. 2019.

[4] Carles Gomez, Joaquim Oller Bosch, and Josep Paradells. "Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology". In: *Sensors (Basel, Switzerland)* 12 (Dec. 2012), pp. 11734–53. DOI: 10.3390/s120911734.

[5] Andrew J Hawkins. *The electric scooter craze is officially one year old — what's next?* https://www.theverge.com/2018/9/20/17878676/electric-scooter-bird-lime-uber-lyft. Accessed: 2020-10-12. Sept. 2018.

[6] Rani Idan. *Don't Give Me a Brake – Xiaomi Scooter Hack Enables Dangerous Accelerations and Stops for Unsuspecting Riders*. https://blog.zimperium.com/dont-give-me-a-brake-xiaomi-scooter-hack-enables-dangerous-accelerations-and-stops-for-unsuspecting-riders/. Accessed: 2020-10-12. Feb. 2019.

[7] Rani Idan. *Mi365Locker*. https://github.com/rani-i/Mi365Locker. 2019.

[8] Karim Lounis and Mohammad Zulkernine. "Bluetooth low energy makes "just works" not work". In: *2019 3rd Cyber Security in Networking Conference (CSNet)*. IEEE, 2019, pp. 99–106.

[9] Mike Murphy and Alison Griswold. *Rebranded Chinese scooters are taking over San Francisco*. en. https://qz.com/1257198/xiaomi-makes-the-bird-and-spin-scooters-taking-over-san-francisco/. Accessed: 2020-10-12. Apr. 2018.

[10] Ronald Stoner. *The next electric scooter you ride could be hacked*. https://medium.com/@forwardsecrecy/the-next-electric-scooter-you-ride-could-be-hacked-7cba3dcc64a4. Accessed: 2020-10-12. June 2019.

[11] *XIAOMI MI M365 EL SPARKESYKKEL SVART - Power.no*. https://www.power.no/hjem-og-fritid/el-sparkesykkel/xiaomi-mi-m365-el-sparkesykkel-svart/p-768734/. Accessed: 2020-10-12.

[12] *Xiaomi Mi M365 Elsparkesykkel, svart EU-modell med to ekstra dekk og slanger (FBC4004GL)*. https://www.multicom.no/xiaomi-mi-m365-elsparkesykkel-svart/cat-p/c/p10264826. Accessed: 2020-10-12.

[13] Yue Zhang et al. "Bluetooth low energy (BLE) security and privacy". In: *Encyclopedia of Wireless Networks*. Cham: Springer International Publishing, 2019, pp. 1–12.

[14] Yue Zhang et al. "On the (in)security of Bluetooth Low Energy one-way Secure Connections Only mode". In: (2019). eprint: 1908.10497.