Wearable Device RF Fingerprinting: an Experimental Study Using COTS Hardware

Artis Rušiņš[1*], Eduards Blumbergs[2], Deniss Tiščenko[1]
, Kirils Solovjovs[1] and Pēteris Paikens[2]
[1]*Institute of Electronics and Computer Science, 14 Dzerbenes st., Riga, Latvia*
[2]*Institute of Mathematics and Computer Science, University of Latvia, Raina blvd. 29, Riga, Latvia*
*Contact: artis.rusins@edi.lv

I. Introduction

In the past 7 years there has been sharp increase in number of connected wearable devices worldwide from 325 million in 2016 to 1105 million in 2022 [1]. This has led to growing concerns about privacy of the data collected by these devices. Wearable devices are becoming more sophisticated and can collect a wide range of data about the user and it is usually sent over using Bluetooth or Bluetooth Low Energy (BLE) standards which we are investigating. RF fingerprinting is one of the emerging techniques that can identify specific device by analyzing radio waveforms generated by target device and extracting unique features from it. However, wearable device RF fingerprinting is relatively unexplored research topic and thus it it important to investigate its effectiveness before offering countermeasures. In this paper we try to replacite results of previous RF fingerprinting attempts for Bluetooth devices.

In this paper we:

- Gather wearable device radio data in isolated environment for automated data capture
- Extract carrier frequency offset (CFO) and amplitude scaling factor fingerprints from the data
- Show that devices are distinguishable by these parameters
- Discuss the impact on wearable device privacy and security

II. Background

Wearable devices typically use short range Peer-to-Peer (P2P) communication standards such as Bluetooth, BLE, Wi-Fi or radio frequency identification (RFID). Due to its widespread adoption worldwide, this paper focuses on Bluetooth standard. Bluetooth is short range wireless communication standard that uses frequency hopping spread spectrum (FHSS) in the 2.4 GHz band to hop between 79 available radio channels. Each channel has a bandwith of 1 MHz.The hopping pattern is pseudo random and is determined by the master device clock and broadcast address (BD_ADDR). To mitigate privacy concerns, Bluetooth devices cryptographically anonymize and periodically rotate the broadcast address (BD_ADDR) of device, while allowing trusted (previously paired) devices to connect to it.

Previous research has shown that this anonymization of BD_ADDR can be rendered useless because it is possible to detect presence of device by its unique RF fingerprints [2]. CFO is the difference between the frequency at which radio transmission is supposted to happen and the frequency at which it actually happens. The ”scaling factor” is amplitude variations within a packet and is used to normalize amplitude to roughly [-1;1] before demodulation. Both CFO and amplitude scaling is usually done by Bluetooth chipset, and these values are not available to user, so we extract them manually.

III. Experimental setup

One of key challanges in extracting RF fingerprints is determining which radio packets are from our DUT and which come from nearby devices. To address this, we do all radio recording inside a radio frequency anechoic chamber. Our recording devices of choise are Ettus Research USRP B210 and B200 software defined radios, due to availability and good software support. They can both record at sampling rate up to 56 MHz. To capture all Bluetooth channels, we use 2 SDRs, with one of them recording lower end of the Bluetooth spectrum and the other one higher end.

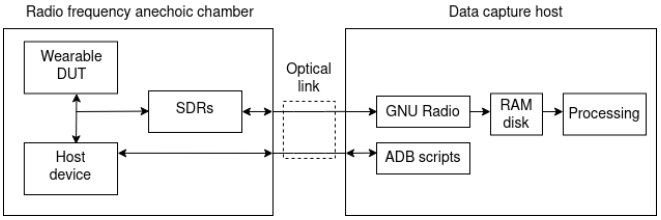

Fig. 1. Overview of capture setup

To capture whole communication process (advertising, pairing and data exchange), we use Android Debuging Bridge (ADB) [3] to enable Bluetooth on the smartphone and start pairing process inside closed chamber while the SDRs record everything. SDRs and smartphone are connected to the host PC via optical-to-USB converter because traditional copper wires can act as antennas and pick up unwanted signals outside the chamber.

SDRs use sample rate of 40 MHz, both data streams are then shifted in frequency, resampled and then added together to create 80 MHz wide signal that includes all Bluetooth channels. We then extract individual bluetooth packets by

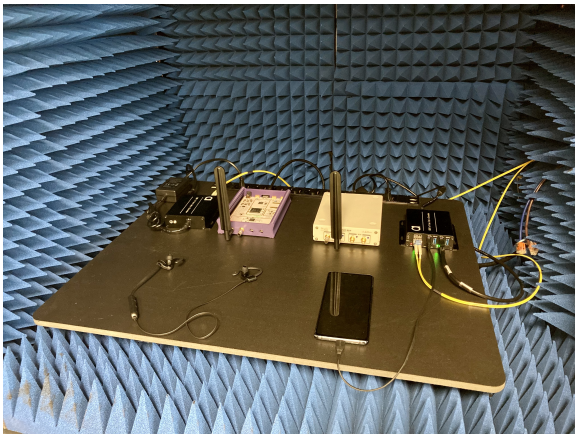

Fig. 2. Anechoic chamber with DUT, host device, USB to optical converters and SDRs

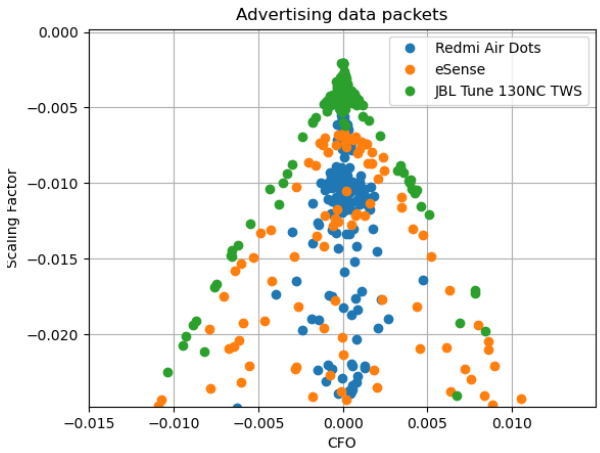

Fig. 3. CFO vs Scaling Factor

performing energy detection and FHSS dehoping using Sania Labs GNU Radio out-of-tree module [4]. To extract CFO and scaling factor values for each packet we use algorithm described by Mike Ryan of ICE9 Consulting LLC [5]. For every detected packet:

1) Separate all samples of negative and positive amplitude
2) $max = median(positive\_samples)$
3) $min = median(negative\_samples)$
4) $CFO = (max - min)/2$
5) $scaling\_factor = (max - CFO)/2$

In our experiments, we use Samsung Galaxy S20 FE smartphone as the host device and all earbuds as wearable DUT.

TABLE I
LIST OF AVAILABLE TEST DEVICES

Model	Type	Bluetooth version
Redmi Air Dots	Earbuds	5.0
JBL Tune 130NC TWS	Earbuds	5.2
eSense	Earbuds	5.0

IV. RF FINGERPRINTS

When plotting the CFO versus scaling factor for all wearable devices for Bluetooth advertising packets, they are visually distinguishable (see Figure 3). By examining the CFO and scaling factor values at different communication phases (advertising, pairing, data streaming), we observed that they do not change significantly for each device. The result is expected since these physical imperfections originate from the hardware components and are not affected by the overlaying protocols.

V. IMPACT

One of the first papers to focus on the importance of fingerprinting was "BlueSniff: Eve meets Alice and Bluetooth" [6]. In this paper the authors explored the potential impact of fingerprinting Bluetooth devices by exploiting the security offered by "undiscoverable mode" and obtaining the full BD_ADDR. The research work led to the development of the first open-source Bluetooth sniffer called "BlueSniff" and emphasized the necessity to tighten security measures to mitigate potential threats. The sniffer used Bluetooth fingerprints to identify and track devices. Although this paper did not demonstrated a practical example of a specific attack, it was one of the first steps to highlight the importance of Bluetooth sniffing capabilities by overcoming eavesdropping obstacles such as anonymized BD_ADDR. Our paper also shows that it is possible to track such devices.

VI. FUNDING

This work was funded by the Latvian Council of Science, project "Automated wireless security analysis for wearable devices", project No. LZP-2020/1-0395.

REFERENCES

[1] F. Laricchia, *Number of connected wearable devices worldwide from 2016 to 2022*, https://www.statista.com/statistics/487291/global-connected-wearable-devices/, Accessed: 24.04.2023.

[2] H. Givehchian, N. Bhaskar, E. R. Herrera, *et al.*, "Evaluating physical-layer ble location tracking attacks on mobile devices," in *2022 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2022, pp. 1690–1704.

[3] "Android Debug Bridge (adb)." (2023), [Online]. Available: https://developer.android.com/tools/adb (visited on 04/26/2023).

[4] "GitHub sandialabs/gr-fhss$_utils$." (2019), [Online]. Available: https://github.com/sandialabs/gr-fhss_utils (visited on 04/26/2023).

[5] "GitHub mikeryan/ice9-bluetooth-sniffer." (2022), [Online]. Available: https://github.com/mikeryan/ice9-bluetooth-sniffer (visited on 04/26/2023).

[6] D. Spill and A. Bittau, "Bluesniff: Eve meets alice and bluetooth.," *WooT*, vol. 7, pp. 1–10, 2007.