RSSI-based Fingerprinting of Bluetooth Low Energy Devices

Guillaume Gagnon, Sébastien Gambs, Mathieu Cunche

RSSI-based Fingerprinting of Bluetooth Low Energy Devices

Guillaume Gagnon[1][a], Sébastien Gambs[1][b] and Mathieu Cunche[2][c]
[1]*Université du Québec à Montréal, Montréal, Canada*
[2]*University of Lyon, INSA-Lyon, Inria, CITI Lab., Lyon, France*
gagnon.guillaume.5@courrier.uqam.ca, gambs.sebastien@uqam.ca, mathieu.cunche@insa-lyon.fr

Keywords: Bluetooth, RSSI, Fingerprinting, Privacy, Unlinkability.

Abstract: To prevent tracking, the Bluetooth Low Energy protocol integrates privacy mechanisms such as address randomization. However, as highlighted by previous researches address randomization is not a silver bullet and can be circumvented by exploiting other types of information disclosed by the protocol such as counters or timing. In this work, we propose a novel attack to break address randomization in BLE exploiting side information that has not been considered before: Received Signal Strength Indication (RSSI). More precisely, we demonstrate how RSSI measurements, extracted from received BLE advertising packets, can be used to link together the traces emitted by the same device or re-identify it despite address randomization. The proposed attack leverages the distribution of RSSI to create a fingerprint of devices. An empirical evaluation of the attack on various scenarios demonstrate its effectiveness. For instance in the static context, in which devices remain at the same position, the proposed approach yields a re-identification accuracy of up to 99%, which can even be boosted by increasing the number of receivers controlled by the adversary.

1 INTRODUCTION

Bluetooth Low Energy (BLE) is a variant of the Bluetooth protocol tailored for application in resource-constrained devices. In particular, BLE has been embedded in a large number of devices such as smartphones and tablets, headphones, health and fitness trackers, tags, etc. In 2022, more than 3 billion BLE-enabled devices were shipped[1]. However, BLE, like other wireless networking technologies, is subject to security and privacy concerns such as breaking of protection mechanisms (Antonioli et al., 2019; Mariotto et al., 2019; Claverie and Lopes-Esteves, 2020) or allowing the exposure of personal information (Das et al., 2016; Martin et al., 2019; Celosia and Cunche, 2020a; Heinrich et al., 2021). In particular, tracking is a major privacy threat for the owner of wireless devices (Gruteser and Grunwald, 2005; O'Hanlon et al., 2014). To counter this threat, anti-tracking mechanisms were included in the first version of BLE (SIG, 2010) such as random addresses, which are unlinkable identifiers periodically renewed.

More recently during the pandemic, BLE has been used as the basis for contact tracing systems (Ahmed et al., 2020), raising another set of security and privacy issues (Vaudenay and Vuagnoux, 2020; Ludant et al., 2021). A key feature of many contact tracing protocols was the use of temporary identifiers that were rotated along with the BLE address.

Address randomization, used as a protection against tracking, has been under heavy scrutiny to identify potential weaknesses. In particular, a number of issues were identified in several implementations, in BLE but also Wi-Fi, that were leveraging the content of the frame (Vanhoef et al., 2016; Martin et al., 2017; Becker et al., 2019) or their timing (Matte et al., 2016). Those weaknesses have been partially fixed in a number of implementations (Fenske et al., 2021).

Another information available to an attacker is the RSSI (Received Signal Strength Indicator), which can be used to estimate the characteristics of the radio link but also for localization (Jianyong et al., 2014) and distance estimation (Larsson, 2015). Nonetheless, so far no study has analyzed how RSSI could be leveraged to defeat address randomization.

In this paper, we demonstrate how an adversary could defeat the BLE address randomization by leveraging RSSI measurements obtained from advertising messages periodically sent by BLE devices. More precisely, our contributions are the following:

[a] https://orcid.org/0009-0007-1717-7418
[b] https://orcid.org/0000-0002-7326-7377
[c] https://orcid.org/0000-0002-0066-8612
[1] https://www.bluetooth.com/2022-market-update/

- We show that RSSI measurements coming from BLE traffic can be used to fingerprint devices, thus completely circumventing other protection mechanisms such as address randomization.
- We propose a novel attack based on a machine learning approach in which the distribution of RSSI measurements is used as features to train a classifier that can link sets of RSSI traces, allowing us to uniquely identify devices.
- We present an evaluation of the attack using a dataset of RSSI measurements collected in a realistic environment under a diverse set of mobility scenarios, demonstrating its efficiency as well as a wide range of applicability.
- We study and discuss the factors influencing the success of the attack, thus characterizing the requirements for a successful attack.

2 BACKGROUND

2.1 Bluetooth Low Energy (BLE)

Bluetooth is a telecommunication standard operating on the 2.4 GHz frequency band, whose objective is to allow a standardized and short-range communication between a wide variety of electronic devices. Integrated into the Bluetooth 4.0 standard in 2010, BLE consumes 10 times less energy than regular Bluetooth and originally offered a speed of up to 1Mbit/s (SIG, 2010). This protocol was extended in 2016 with the adoption of version 5 of the specification, which quadrupled the theoretical transmission range in addition to offering a throughput of approximately 2 Mbps (SIG, 2016).

2.2 BLE Advertising

More specifically, BLE operates in the frequency range between 2400 MHz and 2483.5 MHz, with this range being divided into 40 channels of 2 MHz wide each. Among these channels, the three frequencies of 2402 MHz, 2426 MHz and 2480 MHz, also known respectively as channel 37, 38 and 39, are reserved to serve only as advertising channels (SIG, 2016). The advertising mechanism of BLE operates on the aforementioned advertising channels (37, 38 and 39), in which BLE devices broadcast or receive periodic unidirectional announcements, scan requests, scan responses and connections indications packets with previously unknown devices in the surroundings. In particular, this mechanism is used by unconnected devices to announce their presence in intervals as short as 20 ms (SIG, 2021) using advertising packets.

2.3 Address Randomization

Advertising packets periodically broadcast by BLE devices include a field called Advertising Address (AdvA) containing a Bluetooth device address (BD_ADDR), which is the unique 48-bit identifier of the transmitting device. Unfortunately, this is problematic as anyone within range can read this unique identifier when a device announces its presence over-the-air on advertising channels. This obviously led to significant privacy issues, as it becomes possible to track the movements of an individual through his device (Issoufaly and Tournoux, 2017).

For this reason, Bluetooth LE Privacy was introduced in version 4.0 to increase the difficulty for an adversary to track a device (Woolley, 2015; SIG, 2010). It gives manufacturers the ability to use random BD_ADDR addresses that will automatically change after an interval of their choosing, with a recommended maximum limit of 15 minutes (SIG, 2021). Several recent researches have empirically demonstrated that this limit is typically used by default in common devices running iOS, Android and Windows operating systems (Becker et al., 2019; Martin et al., 2019; Celosia and Cunche, 2020a).

2.4 Received Signal Strength Indication

In BLE, the RSSI is a measure of the power level at the receiver, which is quantified in decibel-milliwatts (*dBm*) on a logarithmic scale. More precisely, the RSSI is a value associated to each received frame and that is made available to the host by the Bluetooth controller. The RSSI value depends on multiple factors including the transmitting power, the gain of the antennas as well as the receiver-transmitter distance. In mobile operating systems, the RSSI can be obtained by mobile applications[2] and thus it can be easily collected by an adversary with a dedicated hardware or simply through a mobile application.

While, RSSI is mainly used to estimate the link quality between two devices, it can also be leveraged to adapt the transmission parameters (SIG, 2021, p.600). In addition, as the RSSI is highly influenced by the distance, it can be used to estimate distances between devices or to locate them with submeter precision (Pau et al., 2021). In particular, RSSI-based distance estimation has been used to design contact tracing applications during the COVID

[2]https://developer.android.com/reference/android/bluetooth/BluetoothDevice#EXTRA_RSSI

pandemic (Ahmed et al., 2020), with advanced distance estimation models having been designed for this (Gorce et al., 2020; Leith and Farrell, 2020).

3 RELATED WORK

The threat of physically tracking users of wireless devices has been investigated thoroughly over the past decade. While this issue applies to all types of wireless technologies, research has mainly focused on Wi-Fi (802.11), Bluetooth and BLE. A number of contributions have highlighted the feasibility of tracking users by collecting link-layer identifiers (O'Hanlon et al., 2014; Issoufaly and Tournoux, 2017). To counter this issue, address randomization has been introduced (Gruteser and Grunwald, 2005) and progressively integrated in Wi-Fi and BLE technologies.

Following this, address randomization has also been scrutinized, with a number of attacks having been published. More precisely, a first class of attacks aim at recovering a stable identifier of the device such as the real MAC address (Vanhoef et al., 2016; Martin et al., 2017; Martin et al., 2019). Another class of attacks aims at linking sequence of messages using various types of information. For instance, this linking has been performed using sequence numbers (Vanhoef et al., 2016) and other stateful elements (Martin et al., 2019; Becker et al., 2019; Celosia and Cunche, 2020a; Celosia and Cunche, 2020b; Ludant et al., 2021). Linking has also been done through fingerprinting, using optional fields (Vanhoef et al., 2016), physical layer information (Vo-Huu et al., 2016; Vanhoef et al., 2016; Hua et al., 2018; Nikoofard et al., 2023) or timing (Matte et al., 2016). In other situations, the implementation has been shown to be flawed, leading to side-channel that can be leveraged by an adversary to circumvent address randomization (Zhang and Lin, 2022). Some of those attacks especially affect the implementation of contact tracing protocols (Ludant et al., 2021). Beyond address randomization, wireless devices can be fingerprinted based on hardware imperfection (Yan et al., 2022; Shen et al., 2021; Givehchian et al., 2022), but those attacks often require specialized hardware.

In many wireless technologies, and especially in BLE, the RSSI has been leveraged for localization (Jianyong et al., 2014) and distance estimation (Larsson, 2015; Gorce et al., 2020). In the context of BLE, RSSI can be used for indoor localization (Jain et al., 2021) following a machine-learning based approach. Channel State Information (CSI) is a more detailed information that can be used for mobility tracking (Rocamora et al., 2020) with better performances than RSSI, but it requires specific hardware and is not supported in BLE (Iannizzotto et al., 2022). In contact tracing, RSSI has also been used to estimate distance (Leith and Farrell, 2020; Gorce et al., 2020), but this estimation is challenging because of the impact of environmental parameters.

4 SYSTEM AND ADVERSARIAL MODELS

4.1 System Model

We consider a setting in which multiple Bluetooth devices are in close proximity within the same physical location, which can be indoor or outdoor. The targeted devices will generally be mobile phones held in different positions with respect to the body of the user (at the ear, in the pocket, etc.). The Bluetooth functionality is assumed to be enabled but no further modification or configuration of the device is required.

Furthermore, these devices are not connected to each other and have never been paired previously via Bluetooth (*i.e.*, thus no information is leaked because of current or previous interactions of these devices). All these devices are assumed to be capable of receiving and transmitting announcement messages, scan requests, scan responses or any other messages that normally transit on the broadcast channels as defined in the BLE specification. In addition, the devices are capable of extracting the RSSI measurements of the signal when receiving messages from nearby devices.

4.2 Adversary Models

The objective of the adversary is to compromise the privacy of a BLE device despite the presence of an address randomization scheme. The adversary is assumed to be fully passive, in the sense that it does not actively transmit any messages to achieve his goal. Rather, he passively monitors the advertising channels and record the received advertising packets. In practice, this means that the receivers controlled by the adversary are following a classical BLE scanning procedure as described in the Bluetooth specifications (SIG, 2021, 4.4.3 Scanning state). We will distinguish between two attack scenarios.

Single receiver. In this first scenario, the adversary passively eavesdrops using a single device on transmissions on Bluetooth broadcast channels by neighboring devices in its close environment for a given time period. The adversary then attempts to use only

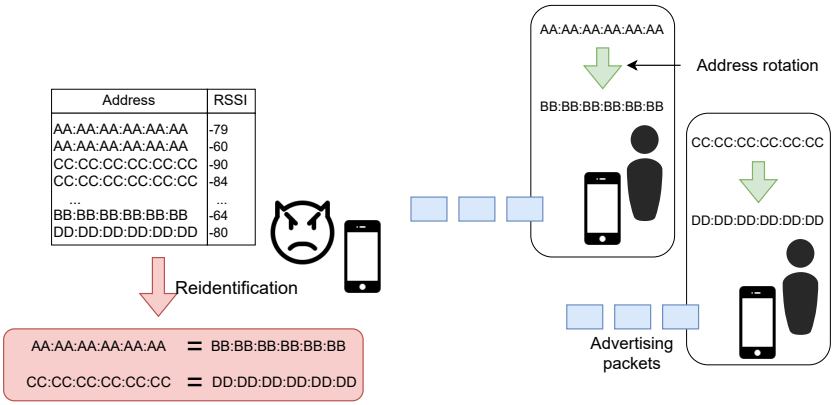

Figure 1: Adversary monitoring BLE advertising packets, leverages RSSI to re-identify devices using address randomization.

the RSSI values received from each of the neighbouring devices to re-establish the identity of the transmitters when they perform a BD_ADDR change.

Multi-receiver. The second scenario is one in which several receivers collude together (or equivalently the adversary controls several devices). In this context, the adversary could for instance own multiple synchronized stations allowing him to passively eavesdrop on transmissions from different physical positions within the same environment. The RSSI values collected by the different listening stations can then be combined to better re-identify the different transmitters that are in this environment when one of them undergoes a BD_ADDR change.

In both of these scenarios, the nature or content of the messages received is of no interest to the adversary, or rather the proposed attack only exploit the RSSI values and thus is agnostic to the content of the message. In this setting, an attack is considered successful if the adversary is able to link together all the messages originating from the same transmitter using only a sequence of RSSI values obtained through messages received from different transmitters. This would, for example, enable the adversary to link device identities in an attempt to determine how long people stay at a given location, or even associate other activities conducted by those people during that time.

5 RE-IDENTIFICATION ATTACK

We propose a new approach to re-identify a Bluetooth device after its BD_ADDR has changed. This novel attack defeats the address randomization mechanism but differs from previous work in that it only relies on the RSSI values computed by the adversary when receiving transmissions over BLE broadcast channels. These RSSI values are first passively accumulated by the attacker before conducting the attack.

More precisely, the proposed attack consists of using sequences of RSSI values to extract a device-specific fingerprint, called a *profile*, which represents the transmission patterns of each transmitter in a given environment. The RSSI profiles collected during a first-period frame are used to train a model, which can later be used to re-identify devices based on profiles collected during a different period.

5.1 RSSI Collection

The attack requires the collection of measurements from neighboring BLE devices by the adversary, which is done by monitoring BLE advertising packets and recording the RSSI and address associated. This collection is done in two phases, which will refer as the training and re-identification steps. The first phase corresponds to the collection of data for training the model while the data gathered during the second phase is used to perform the attack itself. As detailed later in Section 8.1, the strengths of our attack is that, 1) a small number of RSSI measurements is enough to achieve a high success rate for the attack, and 2) the advertising interval is in the order of *ms* for many BLE devices (Apple, 2022; Android, 2023).

Thus in practice, a high number of measurements can be collected even only during a few minutes, as most devices will not change their address during that period. As a result, the adversary will have for each phase a set of identifiers to which is associated a sequence of RSSI measurements.

5.2 Generation of RSSI Profiles

The sequence of RSSI associated to an identifier is used to create what we call a *profile*, which is effec-

tively a distribution of the RSSI values represented as an histogram. More precisely for each identifier, the adversary gathers the corresponding sequence of RSSI and compute their distribution as histogram by decomposing the range of possible values[3] in n_{bin} bins of equal sizes. Finally, this distribution is normalized, which effectively leads to profile being represented as a vector of dimension n_{bin}.

The resulting profile can be used as a fingerprint to identify a device. For instance, Figure 2 shows two examples of profiles corresponding to two distinct devices from the point of view of a single receiver. The difference between those distribution hints that they can possibly be used to fingerprint devices.

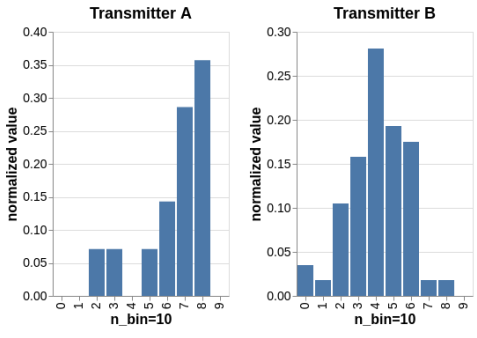

Figure 2: Two normalized profiles of transmitters $\mathcal{A}$ and $\mathcal{B}$ computed by the adversary I in the scenario Q1.

5.3 Generation of Feature Vectors

In our approach, we do not attempt to directly re-identify a profile but rather we will train a classifier at recognizing if two subsequent profiles correspond to the same identity (*i.e.*, device) and, if this is the case, to also output as predicted label the identity of the device. To use the terminology from the privacy literature, our aim is to conduct simultaneously a linking but also re-identification attack. As a consequence before it can be given to a learning algorithm, the data needs to be preprocessed to be represented in the form of a feature vector composed of two concatenated profiles (thus $2 \times n_{bin}$ elements) completed by a label.

Feature vectors are generated by crossing profiles from the same device (*i.e.*, to generate examples for the class associated to the identity of this device) and also distinct devices (*i.e.*, to generate examples for the negative class in which the two profiles are from different devices). A feature vector is thus created by concatenating two profiles and a label as follows:

$$v = \{h_i, h_j, \delta_{i,j}\} \tag{1}$$

in which h_i and h_j are two profiles of length n_{bin} corresponding to devices i and j while the label $\delta_{i,j}$ is computed as follows:

$$\delta_{i,j} = \begin{cases} i, & \text{if } i == j \\ \times, & \text{otherwise.} \end{cases} \tag{2}$$

Each of the two profiles represents a device while the label represent whether the profiles are coming from the same device or not, and in case they are, the label takes the value of the corresponding device identifier. A feature vector created with two profiles coming from two distinct devices is called a *non-matched* vector and is labeled with $\times$. In contrast, a feature vector created with two profiles from the same device, is called a *matched* vector and is labeled with i, the identifier of the device. The resulting dataset is composed of $n_{profile} = (n_{device} \times n_{split})(n_{device} \times n_{split} - 1)$.

5.4 Linkage Attack

The final step of our re-identification attack is the linkage attack. To realize this, a machine learning model is first trained using the data collected in the first phase of RSSI collection as described in Section 5.1. Afterwards using this model, the linkage attack takes as input a feature vector corresponding to two sequences of RSSI and outputs a label that can be the identifier *id* of a device (indicating matched profiles generated by device *id*) or $\times$ (indicating non-matched profiles coming from different devices).

The linking attack thus produces two elements of information. The first one is whether the feature vector is *matched/non-matched* while the second information, only meaningful in the case of *matched* vectors, is the identifier[4] of the device.

6 DATASET

To evaluate our attack, we have used a dataset of RSSI measurements collected by Inria in the context of the development[5] of a BLE-based contact tracing solution (Castelluccia et al., 2020). This data was obtained by running scenarios in a controlled environment with the participation of voluntary military personnel to fulfill the role of participants. The collected

[3]The range of possible values for the RSSI depends on the receiver characteristics. In our case, the values were comprised between -23 *dBm* and -104 *dBm*.

[4]Note that this identifier is just a pseudonym, and is not a stable identifier of the device.

[5]https://gitlab.inria.fr/stopcovid19

data was strictly limited to the dedicated devices involved in the experiment. In addition, as seen in Table 1 this data is solely composed of series of RSSI measurements and does not include any personal data. This dataset was gracefully shared by Inria with us to conduct this study but is not publicly available.

In more details, the dataset features measurements of BLE advertising packets emitted and collected by smartphones[6] carried by participants in various experimental scenarios that took place indoors and outdoors and in different types of spaces (free space, gymnasium, meeting room, warehouse, amphitheater and metro). Scenarios have a duration of 15 minutes and involve up to 30 participants/smartphones. During a scenario, each smartphone is both acting as an emitter and a receiver as it sends BLE advertising packets while performing BLE scanning.

Some of these scenarios are static with each participants being positioned on a grid. In those experiments, each participant stand still on a $0.5m \times 0.5m$ cell (see Figure 3). Other scenarios are dynamic and involved participant changing their position and gesture after remaining stationary for at least 3 minutes in a cell. Each scenario was also classified into two density categories: Low and High for respectively 0.15 and 0.3 participants per m^2.

For the sake of the experiments, the advertising packets emitted every second also included an identifier that unambiguously identified the source of that advertising packet. Using a dedicated application, each smartphone recorded the advertising packets received from other smartphones and logged the following information : time of arrival (in sec), RSSI (in *dBm*), transmitter identifier and phone states. Those data were later centralized and enriched with additional information such as the positions of the transmitters and the real distances between participants.

The resulting dataset is organized by scenarios and structured in *records*, in which each record corresponds to an advertising packet emitted by a device $\mathcal{A}$ and received by device $\mathcal{B}$. Each scenario is accompanied by a description including general information such as the nature of the environment (*i.e.*, indoor or outdoor) and the type of scenario (*i.e.*, static or dynamic). A record includes a timestamp, the identity of the sender and receiver, their coordinates, the RSSI value, the locked state, application state and screen states as well as the real distance between these two devices. A sample of records is presented in Table 1.

[6]iPhone 11 Pro Max, Samsung 10 SM-G973F and Samsung 10+ SM-G975F.

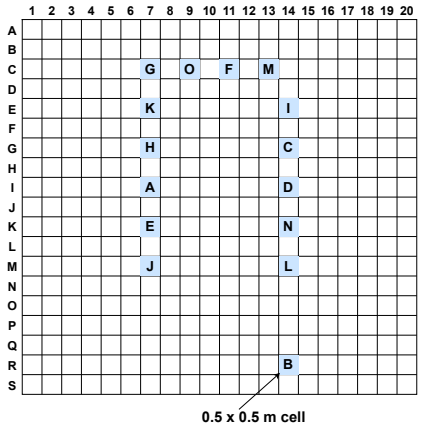

Figure 3: Experimental scenario Q1 emulating a meeting room in which users remain static. I and I & $\mathcal{L}$ were used for the static context attack with one and two receivers.

7 EVALUATION METHODOLOGY

7.1 Scenario Selection

The dataset we used (*cf.* Section 6) includes captures from multiple scenarios in different conditions. Amongst these scenarios, we discarded those with an insufficient number of records[7], devices or lacking details in the experimental protocol used for data collection. The resulting six selected scenarios cover a diverse set of situations, including static and dynamic scenarios involving between 15 and 30 devices. The details of the selected scenarios are shown in Table 2.

In the rest of the paper, the static scenario G3 will be used as the running example as it is an ideal case in terms of the number of messages received by a single receiver, with each transmitter having sent between 888 and 908 messages to the adversary $\mathcal{H}$. Note that transmitter $\mathcal{A}$ had to be removed from that dataset because it only has 505 RSSI measurements associated to it. In all other scenarios, the number of transmissions per device is much lower as can be seen from the *Nb. Records* column of Table 2.

7.2 Records Preprocessing

As the dataset was not originally collected to evaluate the performance of a re-identification attack, it needs to be preprocessed to prepare it to match the requirements of our adversary model.

First, we have removed the unnecessary fields to keep only the identity of the receiver(s), transmitters

[7]900 records is the maximum given the 15 minutes duration of our scenario and the transmission frequency being of 1/sec. at best.

Table 1: Examples of records from one of the scenario. Each record corresponds to an advertising packet and includes among other things, the sender's identity, the receiver's identity, a timestamp and the RSSI value.

RSSI	Timestamp	Rcv_position	Locked	Screen	State	Receiver	Transmitter	Trans_position	Distance
-69	2020-05-18 17:26:39	C3	FALSE	TRUE	active	A	O	D8	2.549
-79	2020-05-18 17:26:39	C16	FALSE	TRUE	active	I	O	D8	4.031
-74	2020-05-18 17:26:40	H2	FALSE	TRUE	active	B	O	D8	3.605

Table 2: Specifications of the selected scenarios.

Scenario	Context	Density	Nb. Devices	Nb. Records	Duration	Devices Types	Location	Details
G3	Static	Low	15	11694	15m09s	SM+, IP11	Indoor	Small hangar
Q1	Static	Low	15	8227	15m06s	SM+, IP11	Indoor	Meeting room
C1	Static	High	30	13821	15m14s	SM, SM+, IP11	Indoor	Gymnasium
H3	Static	Low	15	3411	15m05s	SM	Outdoor	Parking
E2	Dynamic	Low	15	12053	15m07s	SM+, IP11	Indoor	Small hangar
F1	Dynamic	Low	15	800	15m08s	SM+	Outdoor	Parking

and the RSSI values. The *timestamp* column can also be omitted since the records of each transmitter are ordered chronologically. To reflect the adversarial model, we assume that one of the nodes is playing the role of an attacker while the others are considered as possible targets. Given an adversary controlling the receiver $\mathcal{A}$, the data available to him for the attack is simply the subset of records having $\mathcal{A}$ as a receiver.

The slow advertising frequency in the dataset results in some devices having received less than 100 RSSI from their neighbors. To account for this, the adversaries are chosen by selecting those that have received the most advertising messages from their neighbors in that scenario. However, this is less likely to be an issue in a real-world setting in which a capture could be made every few tens of milliseconds (SIG, 2021, p.2749).

7.3 Sequences and Profiles Creation

In the dataset, devices keep the same identity during the duration of the scenario. To conduct our experiments, the set of records is sliced into n_{split} sequences of approximately equal size. As a result, each device is associated to a total of n_{split} sequences of RSSI measurements that can be used for training and evaluation. The temporal order of the RSSI values is preserved by the split. For each of those sequences, a profile is generated as described in Section 5.2, which means that a given adversary has for each target device a total of n_{split} profiles.

7.4 Training and Evaluation Design

To train and evaluate our model, a set of feature vectors is required. As described in Section 5.3, feature vectors are created by concatenating two profiles and a label. Thus, from the set of profiles previously generated (Section 7.3), we can combine profiles from the same device to create *matched* vectors and profiles from different devices to create *non-matched* vectors.

Note that for combinatorial reasons, the number of possible *non-matched* vectors far exceed the number of *matched* vectors. To avoid having an imbalanced dataset with a large majority of *non-matched* vectors, we enforce a *crossing random sample ratio* (CRSR). The CRSR parameter allows us to control the ratio between the *matched* and *non-matched* vectors in the following manner :

$$CRSR = \frac{\text{\#non-matched vectors}}{\text{\#matched vectors}} \tag{3}$$

To reflect a more challenging environment for the adversary, we consider a CRSR larger than one for the training phase. Indeed, in a real-world scenario, the number of *non-matched* vectors will increase faster than the number of *matched* as the number of devices grows. However, a balanced dataset is built for the evaluation phase.

For each scenario, the data available is divided into two distinct sets : the training set for which 75% of data is used (phase 1) and the evaluation set composed of the remaining 25% (phase 2). A balanced test set in which *matched* and *non-matched* are equally represented (CRSR = 1) is then extracted from the evaluation set. Although both evaluation sets are used for the assessment of our models, the balanced evaluation set is the one used for the results reported in the following sections. This allows us to use accuracy as a meaningful metric.

7.5 Machine Learning Setting

Following the previously described methodology, we have trained and tested several machine learning algorithms typically used for this type of problem. More precisely, we have selected two ensemble-based classification methods, namely HistGradient-

BoostingClassifier (HGT) and RandomForestClassifier (RF), as well as a k-nearest algorithm, KNeighborsClassifier (kNN), for which we used the library scikit-learn. The data was shuffled and we used a cross-validation with a stratified kFold in which $k = 5$.

7.6 Performance Metrics

In the context of the re-identification attack, while the underlying task corresponds to a multi-class setting, the analysis can be simplified by computing the following quantities related to the outcome of the classification:

- TP (true positive): a *matched* vector is correctly labeled as *matched* along the correct device id.
- TN (true negative): a *non-matched* vector correctly is labeled as *non-matched*($\times$ label).
- FP (false positive): a *non-matched* vector is incorrectly labeled as *matched*.
- FN (false negative): a *matched* vector is incorrectly labeled as *non-matched* or associated to an incorrect device id.

From these quantities, several metrics can be computed to quantify the performance of our attack scheme. More precisely, the accuracy is defined as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

To complement the accuracy results that are obtained with the balanced evaluation set, we have also considered the precision, recall and F1-score, that provide more detailed information on the performances in the context of multi-class task (with the TP, TN, FP, and FN being computed in the context of the class). They are defined as follows :

$$\text{Precision} = \frac{TP}{TP+FP} \qquad \text{Recall} = \frac{TP}{TP+FN} \quad (5)$$

$$\text{F1-score} = \frac{2 \times (Recall \times Precision)}{Recall + Precision} \quad (6)$$

For these three metrics, we have also reported their macro average, which is their unweigthed mean per label, as well as their weighted average, which is the support weighted mean per label.

8 EVALUATION RESULTS

8.1 Static Scenarios with a Single Receiver

The evaluation of our attack was first conducted using a static scenario with parameters n_{split} and n_{bin} set respectively to 15 and 10. As an illustration, Figure 4 presents the result of the attack applied to scenario Q1 with a CRSR of 5 and the algorithm HGT in the form of a confusion matrix displaying the predicted labels made by the algorithm against the true labels. A vast majority of the points falls on the diagonal reflecting a correct prediction. In addition, all the prediction errors are located in the last column, which corresponds to the situation in which the algorithm failed to recognize a *matched* vector and has classified it incorrectly as *non-matched*. In contrast, all *matched* predictions have correctly recognized the identity of the device. The performance of HGT presented in Table 3 is high with an accuracy of 0.99 and precision, recall and F1-score having an average value for the *matched* classes respectively equal to 1, 0.98 and 0.99.

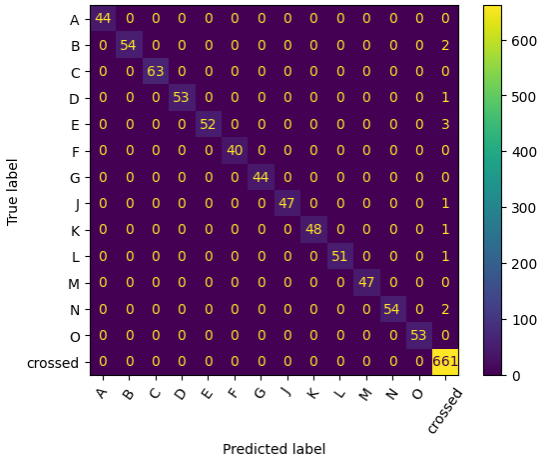

Figure 4: Result of the attack conducted by adversary I in static scenario Q1. The HGT algorithm was used and the assessment was performed using a balanced evaluation set.

Table 3: Balanced test classification report on the static scenario Q1 using I as the adversary with the HGT algorithm.

Class		Prec.	Recall	F1-score	# vectors
matched	A	1.00	1.00	1.00	44
	B	1.00	0.96	0.98	56
	C	1.00	1.00	1.00	63
	D	1.00	0.98	0.99	54
	E	1.00	0.95	0.97	55
	F	1.00	1.00	1.00	40
	G	1.00	1.00	1.00	44
	J	1.00	0.98	0.99	48
	K	1.00	0.98	0.99	49
	L	1.00	0.98	0.99	52
	M	1.00	1.00	1.00	47
	N	1.00	0.96	0.98	56
	O	1.00	1.00	1.00	53
	avg	**1.00**	**0.98**	**0.99**	661
non-matched	$\times$	0.98	1.00	0.99	661
Accuracy				0.99	1322
Macro avg		1.00	0.99	0.99	1322
Weighted avg		0.99	0.99	0.99	1322

Impact of the Learning Algorithm. We have compared the performances of three considered learning algorithms: HGT, RF and k-NN. In this evaluation, the n_{split} and n_{bin} were also set to 10 and 15 while CRSR was set to 5. The associated results are presented in Table 4 and demonstrate that HGT clearly outperforms other algorithms with an accuracy always above 98% while RF and k-NN are both below 79% and 65% accuracy for all single adversary attack. Moreover, the costs of data preprocessing and model training are minimal, making it easy for an adversary to conduct the attack almost in real-time on the field. More specifically, it takes less than 20 seconds for a standard laptop to preprocess a dataset containing up to 30 devices. In addition, it takes a maximum of 5 seconds to train the model on the preprocessed data.

Impact of CRSR. With the objective of analyzing the impact of the CRSR parameter (previously set to 5), we evaluated the performances of the three algorithms using a range of values for the CRSR from 1 to 12 using scenarios Q1 and G3. The results of the evaluation presented in Figure 5 shows that HGT has superior accuracy than the other algorithms. In addition, the CRSR has also clearly an impact on the accuracy, with focusing on HGT, the highest accuracy being obtained for values between 4 and 5. Overall, using HGT with a CRSR of 4 yield to an accuracy of 99% for both scenarios Q1 and G3.

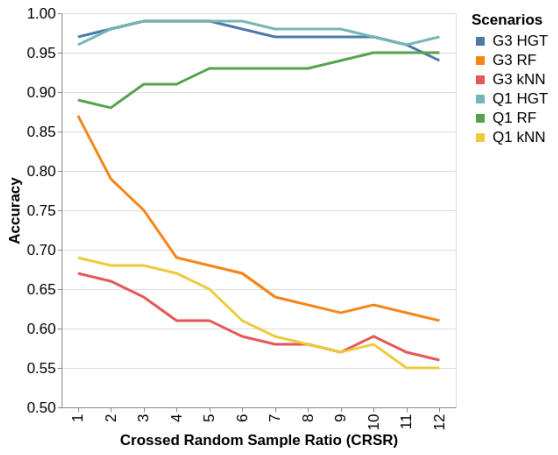

Figure 5: Effects of CRSR variations on performances for HGT, RF and kNN, using the balanced evaluation set of the G3 and Q1 static scenarios with $n_{split} = 15$ and $n_{bin} = 10$.

Impact of histogram resolution (n_{bin}) and sequences slicing (n_{split}). We have also studied how the parameters of the attack, n_{bin} and n_{split} (initially fixed respectively to 10 and 15), impact the performance of the attack. To realize this, using the scenario G3 data and the HGT algorithm, the accuracy of the attack is evaluated for all the combination of (n_{bin}, n_{split}) $\in [2..20]^2$ with a CRSR set to 5.

The results of this evaluation are presented in Figure 6. Overall, the accuracy increases with both n_{bin} and n_{split}. For instance, using the highest value 20 for those parameters leads to an accuracy of 1. Nevertheless, having both n_{bin} and n_{split} equal or above 10 is enough to yield an accuracy of 0.97 or above. Thus to obtain good performances, it is necessary to have a resolution of 10 values for the RSSI distribution and to have at least 11 sequences ($n_{split} \geq 11$). Note that the n_{split} parameter also controls the length of the sequences and on the considered scenario, having n_{split} between 10 and 20 implies that each sequence contains between 44 and 91 RSSI measurements.

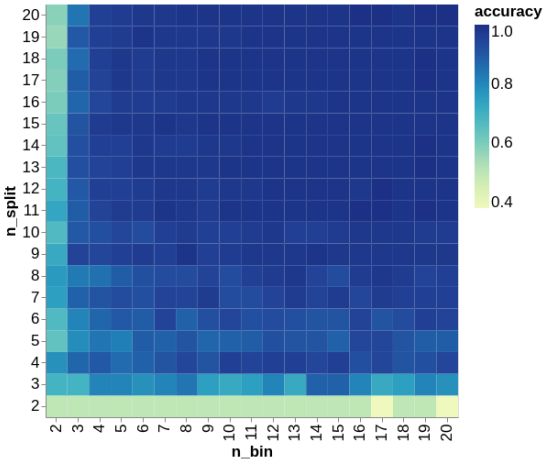

Figure 6: Effects of n_{split} and n_{bin} variations on HGT model performance with the balanced evaluation set of G3.

Impact of the quantity of RSSI measurements. Previous evaluations were using all the RSSI measurements available in a scenario. To evaluate the impact on attack performance of the amount of RSSI measurements available to the adversary, we restricted the number of records used. As can be seen in Figure 7, HGT shows again its superiority as having a minimum of 90 RSSI measurements per transmitter, resulting in only 6 RSSI by profile when using $n_{split} = 15$, is sufficient to achieve a 94% accuracy.

8.2 Static Scenario with Multiple Receivers

In the previous subsection, the adversary was assumed to be collecting RSSI measurements from a single receiver. In this section, we consider an adversary that controls multiple receivers. Note that

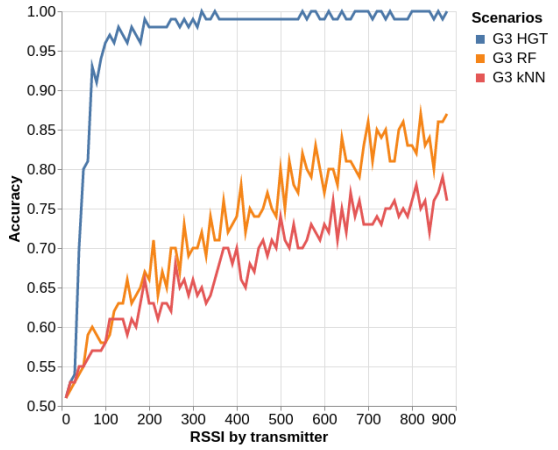

Figure 7: Effects of the number of RSSI on the performance of algorithms with $n_{split} = 15$ and $n_{bin} = 10$ on the balanced evaluation set of the static scenario G3.

having more than one receiver does not significantly change the attack framework. More precisely, instead of having a profile composed of a distribution of values coming from one receiver, the profile is just a sequence of distributions that each correspond to one receiver. Therefore, using k receivers, the resulting size of the profile is $k \times n_{bin}$.

We have performed a performance evaluation using the data from scenario Q1, in which we selected devices $\mathcal{I}$ and $\mathcal{L}$ as being the receivers controlled by the adversary. Furthermore, we set n_{bin} and n_{split} to 10 and 15 and we have relied on the HGT algorithm. The results of this evaluation are presented in Figure 8. The performance of the attack is high, with only two errors being made, resulting in an accuracy and average F1-score for true identities of 1.0. Adding a receiver increased the accuracy of all algorithms in all scenarios as shown in Table 4. Performance is likely to improve further with additional receivers.

Table 4: Static single and multi-adversary scenario accuracy using HGT, RF and kNN trained with $n_{bin} = 10$, $n_{split} = 15$, $CRSR = 5$ and assessed using a balanced evaluation set.

	Static Single-adversary			**Static Multi-adversary**		
Scenario	*HGT*	*RF*	*kNN*	*HGT*	*RF*	*kNN*
Q1	0.99	0.79	0.65	1.00	0.88	0.78
G3	0.99	0.69	0.61	1.00	0.79	0.67
C1	0.98	0.69	0.61	0.99	0.83	0.76
H3	0.98	0.65	0.57	0.99	0.85	0.75

8.3 Dynamic Scenario

Previous evaluations were using scenarios in which both the receiver (*i.e.*, the adversary) and the transmitters (*i.e.*, the targets) were not moving. We now assess the performance of our attack in a dynamic scenario in which both the targets and the adversary are

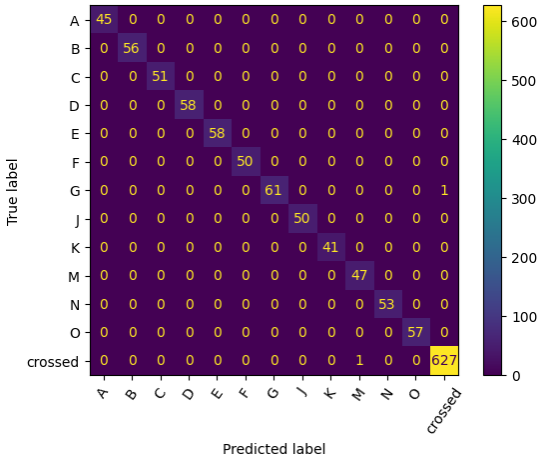

Figure 8: Result of the multi-adversary attack conducted from the perspective of receivers $\mathcal{I}$ and $\mathcal{L}$ in the static scenario Q1. The HGT algorithm was used and the assessment is performed using a balanced evaluation set.

moving. More precisely, we rely on the data from the scenario E2, in which participants were changing position up to five times during the scenario. In this experiment, device $\mathcal{H}$ was selected as the adversary as it was almost static. More precisely, it only changed its position once in the middle of the scenario to move to a neighboring cell. Training was done using data collected by $\mathcal{H}$ when it was located at its first position, while the evaluation was performed with the data collected at its second location. For all devices except $\mathcal{A}$, the performance is low as the adversary always failed to re-identify the targets. The fact that $\mathcal{A}$ was correctly re-identified can be explained by, although they both moved, $\mathcal{A}$ was always 2 meters away from the receiver $\mathcal{H}$. After further investigation, we hypothesize that the combination of a small variation in relative distance and the fact that $\mathcal{A}$ is much closer to the receiver, result in high and therefore more stable RSSI, which has a much smaller impact on the quality of the fingerprint of our RSSI profiles.

Table 5: Summary of dynamic single adversary scenarios accuracy of algorithms trained with $n_{bin} = 10$, $n_{split} = 15$, $CRSR = 5$, and assessed using a balanced evaluation set.

	Dynamic Single-adversary		
Scenario	HGT	RF	kNN
E2	0.53	0.54	0.52
F1	0.49	0.49	0.47

We also tested this approach on the F1 scenario, which in contrast to E2 took place outdoor. In this setting, we observed the same results, as none of the transmitters could be re-identified. The detailed results of our evaluations on the dynamic scenarios are presented in Table 5. Overall, those results show that

the relative position of the targets and receiver impact significantly the performance of the attack.

9 CONCLUSION

In this paper, we have introduced a novel attack to defeat address randomization using RSSI measurements of received BLE advertising packets. This attack has been evaluated using a dataset of RSSI measurements obtained from real smartphones. The results obtained showed that, in scenarios involving up to 30 motionless devices, a re-identification accuracy of 0.99 can be obtained, and that those performances can be further improved by increasing the number of receivers controlled by the adversary. However, the success of the attack decreased against mobile targets unless their relative movement is limited. We thus envision the following countermeasures.

Silent period. The results of our evaluation suggest that our attack is not as efficient against mobile targets as against static ones. Thus, the variations of the RSSI profile induced by a change of relative position might be one of the simplest yet effective countermeasure to reduce the risk of re-identification. Thus to prevent RSSI-based re-identification attacks in the context of address randomization, it is important to ensure that a device has moved significantly before using its new identifier. This requirement further motivates the need for a silent period (Leping Huang et al., 2005) before rotating the address. Indeed, during the silent period, the amplitude of the position change will necessarily be larger than without it, and the environment is also more likely to change (people moving around the target, device changing position, ...). In practice, this means than rather than having a fixed duration before changing the random address, a better strategy could be to coordinate this change with a location shift (*e.g.*, detected by the accelerometer).

Modifying the transmitting power. Since the attack relies on RSSI measurements, another natural countermeasure is to add some noise in those values, which could be achieved by randomly changing the transmission power of the Bluetooth controller. On Android, it is possible to change the advertising power between 6 power settings[8]. However, changing the transmission power will have a negative impact on the quality of service with lost packets and also potentially increasing the energy consumption.

[8]https://developer.android.com/reference/android/bluetooth/le/AdvertisingSetParameters

As future works, we envision adapting this attack against similar wireless technologies such as Bluetooth Classic or Wi-Fi (802.11). We would also like to improve the framework for dynamic scenarios, for instance by using multiple receivers or a continuous re-identification strategy rather than the two-phase one that we have currently used.

ACKNOWLEDGEMENTS

This research has been supported by the ANR PIVOT project ANR-20-CYAL-0002, PEPR Cybersécurité IPOP project and Inria MAGPIE associated team. Sébastien Gambs is supported by the Canada Research Chair program as well as a Discovery Grant from NSERC.

REFERENCES

Ahmed, N., Michelin, R. A., Xue, W., Ruj, S., Malaney, R., Kanhere, S. S., Seneviratne, A., Hu, W., Janicke, H., and Jha, S. K. (2020). A Survey of COVID-19 Contact Tracing Apps. *IEEE Access*, 8:134577–134601.

Android (2023). Android api reference - advertisingsetparameters.

Antonioli, D., Tippenhauer, N. O., and Rasmussen, K. B. (2019). The KNOB is Broken: Exploiting Low Entropy in the Encryption Key Negotiation Of Bluetooth BR/EDR. In *USENIX Security Symposium*.

Apple (2022). Accessory design guidelines for apple devices.

Becker, J. K., Li, D., and Starobinski, D. (2019). Tracking Anonymized Bluetooth Devices. *Proceedings on Privacy Enhancing Technologies*.

Castelluccia, C., Bielova, N., Boutet, A., Cunche, M., Lauradoux, C., Métayer, D. L., and Roca, V. (2020). ROBERT: ROBust and privacy-presERving proximity Tracing. Technical report.

Celosia, G. and Cunche, M. (2020a). Discontinued Privacy: Personal Data Leaks in Apple Bluetooth-Low-Energy Continuity Protocols. *Proceedings on Privacy Enhancing Technologies*, 2020(1):26–46.

Celosia, G. and Cunche, M. (2020b). Saving private addresses: An analysis of privacy issues in the bluetooth-low-energy advertising mechanism. In *Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MobiQuitous '19, pages 444–453. ACM.

Claverie, T. and Lopes-Esteves, J. (2020). Testing for weak key management in Bluetooth Low Energy implementations. In *SSTIC*.

Das, A. K., Pathak, P. H., Chuah, C.-N., and Mohapatra, P. (2016). Uncovering Privacy Leakage in BLE Network Traffic ofWearable Fitness Trackers. *ACM HotMobile*.

Fenske, E., Brown, D., Martin, J., Mayberry, T., Ryan, P., and Rye, E. (2021). Three years later: A study of mac address randomization in mobile devices and when it succeeds. *Proceedings on Privacy Enhancing Technologies*, 2021(3).

Givehchian, H., Bhaskar, N., Herrera, E. R., Soto, H. R. L., Dameff, C., Bharadia, D., and Schulman, A. (2022). Evaluating Physical-Layer BLE Location Tracking Attacks on Mobile Devices. In *2022 IEEE Symposium on Security and Privacy (SP)*.

Gorce, J.-M., Egan, M., and Gribonval, R. (2020). An efficient algorithm to estimate Covid-19 infectiousness risk from BLE-RSSI measurements. report, Inria Grenoble Rhône-Alpes.

Gruteser, M. and Grunwald, D. (2005). Enhancing Location Privacy in Wireless LAN Through Disposable Interface Identifiers: A Quantitative Analysis. *Mobile Networks and Applications*, (3).

Heinrich, A., Stute, M., Kornhuber, T., and Hollick, M. (2021). Who Can *Find My* Devices? Security and Privacy of Apple's Crowd-Sourced Bluetooth Location Tracking System. *Proceedings on Privacy Enhancing Technologies*, 2021(3).

Hua, J., Sun, H., Shen, Z., Qian, Z., and Zhong, S. (2018). Accurate and efficient wireless device fingerprinting using channel state information. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE.

Iannizzotto, G., Milici, M., Nucita, A., and Lo Bello, L. (2022). A Perspective on Passive Human Sensing with Bluetooth. *Sensors*, 22(9). Number: 9 Publisher: Multidisciplinary Digital Publishing Institute.

Issoufaly, T. and Tournoux, P. U. (2017). BLEB: Bluetooth Low Energy Botnet for large scale individual tracking. In *Next Generation Computing Applications*. IEEE.

Jain, C., Sashank, G. V. S., N, V., and Markkandan, S. (2021). Low-cost BLE based Indoor Localization using RSSI Fingerprinting and Machine Learning. In *WiSPNET*.

Jianyong, Z., Haiyong, L., Zili, C., and Zhaohui, L. (2014). RSSI based Bluetooth low energy indoor positioning. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*.

Larsson, J. (2015). *Distance Estimation and Positioning Based on Bluetooth Low Energy Technology*.

Leith, D. J. and Farrell, S. (2020). Coronavirus contact tracing: evaluating the potential of using bluetooth received signal strength for proximity detection. *ACM SIGCOMM Computer Communication Review*, 50(4).

Leping Huang, Matsuura, K., Yamane, H., and Sezaki, K. (2005). Enhancing wireless location privacy using silent period. In *IEEE Wireless Communications and Networking Conference, 2005*, volume 2. IEEE.

Ludant, N., Vo-Huu, T. D., Narain, S., and Noubir, G. (2021). Linking bluetooth le & classic and implications for privacy-preserving bluetooth-based protocols. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1318–1331. IEEE.

Mariotto, A., Heinrich, A., Kreitschmann, D., Noubir, G., and Hollick, M. (2019). A Billion Open Interfaces for Eve and Mallory: MitM, DoS, and Tracking Attacks on iOS and macOS Through Apple Wireless Direct Link. *USENIX Security*.

Martin, J., Alpuche, D., Bodeman, K., Brown, L., Fenske, E., Foppe, L., Mayberry, T., Rye, E., Sipes, B., and Teplov, S. (2019). Handoff All Your Privacy – A Review of Apple's Bluetooth Low Energy Continuity Protocol. *Proceedings on Privacy Enhancing Technologies*, 2019(4):34–53.

Martin, J., Mayberry, T., Donahue, C., Foppe, L., Brown, L., Riggins, C., Rye, E. C., and Brown, D. (2017). A Study of MAC Address Randomization in Mobile Devices and When it Fails. *Proceedings on Privacy Enhancing Technologies*, 2017(4).

Matte, C., Cunche, M., Rousseau, F., and Vanhoef, M. (2016). Defeating MAC Address Randomization Through Timing Attacks. In *ACM WiSec*.

Nikoofard, A., Givehchian, H., Bhaskar, N., Schulman, A., Bharadia, D., and Mercier, P. P. (2023). Protecting Bluetooth User Privacy Through Obfuscation of Carrier Frequency Offset. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 70(2).

O'Hanlon, P., Wright, J., and Brown, I. (2014). Privacy at the link layer. In *STRINT Workshop: A W3C/IAB workshop on Strengthening the Internet Against Pervasive Monitoring (STRINT)*.

Pau, G., Arena, F., Gebremariam, Y. E., and You, I. (2021). Bluetooth 5.1: An Analysis of Direction Finding Capability for High-Precision Location Services. *Sensors*, 21(11).

Rocamora, J. M., Wang-Hei Ho, I., Mak, W., and Lau, A. P. (2020). Survey of CSI fingerprinting-based indoor positioning and mobility tracking systems. *IET Signal Processing*, 14(7).

Shen, G., Zhang, J., Marshall, A., Peng, L., and Wang, X. (2021). Radio Frequency Fingerprint Identification for LoRa Using Spectrogram and CNN. In *IEEE INFOCOM*.

SIG, B. S. I. G. (2010). Bluetooth core specification 4.0.

SIG, B. S. I. G. (2016). Bluetooth core specification 5.0.

SIG, B. S. I. G. (2021). Bluetooth core specification 5.3.

Vanhoef, M., Matte, C., Cunche, M., Cardoso, L. S., and Piessens, F. (2016). Why MAC Address Randomization is Not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms. In *ACM AsiaCCS*.

Vaudenay, S. and Vuagnoux, M. (2020). Little Thumb Attack on SwissCovid.

Vo-Huu, T. D., Vo-Huu, T. D., and Noubir, G. (2016). Fingerprinting Wi-Fi Devices Using Software Defined Radios. In *ACM WiSec*, WiSec '16.

Woolley, M. (2015). Bluetooth technology protecting your privacy.

Yan, W., Voigt, T., and Rohner, C. (2022). RRF: A Robust Radiometric Fingerprint System that Embraces Wireless Channel Diversity. In *ACM WiSec*.

Zhang, Y. and Lin, Z. (2022). When Good Becomes Evil: Tracking Bluetooth Low Energy Devices via Allowlist-based Side Channel and Its Countermeasure. In *ACM CCS*.