Privacy in the Mobile World: An Analysis of Bluetooth Scan Traces

Heng Zhang
Purdue University
West Lafayette, IN, USA
zhan2614@purdue.edu

Amiya K. Maji
Purdue University
West Lafayette, IN, USA
amaji@purdue.edu

Saurabh Bagchi
Purdue University
West Lafayette, IN, USA
sbagchi@purdue.edu

Abstract

Bluetooth-enabled smartphones, wearable devices, as well as consumer electronics devices, are pervasive nowadays. Due to the low power consumption of Bluetooth hardware, users often leave Bluetooth enabled on their personal devices all the time. We find that even though the devices themselves may be protected against unauthorized connections, neighboring Bluetooth signals may still leak personal information. More specifically, a malicious smartphone application can easily obtain permission to perform Bluetooth scanning and then build a temporal trace of the number of active Bluetooth devices in the vicinity of a user. By collecting and analyzing data from 49 smartphone users over two weeks, we found that traces from different devices have little overlap and can, therefore, be used to identify a device with high likelihood. Moreover, Bluetooth advertisements from nearby devices can reveal what products the user may own making her susceptible to targeted advertisements. By comparing Bluetooth traces from multiple devices, the adversary can learn a user's location even if she does not give explicit permission to share her location. We also analyzed a public Bluetooth dataset to find similarities and differences with the conclusions drawn from our dataset. Our dataset has been publicly released for the scientific community.

ACM Reference Format:
Heng Zhang, Amiya K. Maji, and Saurabh Bagchi. 2020. Privacy in the Mobile World: An Analysis of Bluetooth Scan Traces. In *2020 Joint Workshop on CPS&IoT Security and Privacy (CPSIOTSEC'20), November 9, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3411498.3419962

1 Introduction

The powerful capabilities of mobile devices (computing, networking, sensing, location-awareness etc.) have led to the creation of a massive number of mobile applications over the last decade. These applications often provide enormous convenience in our daily lives and their demands are continuously growing. Modern mobile applications and wearable devices usually use Bluetooth to communicate over short-range, e.g., Android Auto, Apple CarPlay, or fitness devices such as Fitbit [7], etc. Because of substantial energy efficiency of Bluetooth compared to Wifi or 4G, some devices continuously advertise their existences once powered on. Therefore, we live in a world surrounded by Bluetooth signals, which motivates us to think: *Can a user's privacy be compromised because of the neighboring Bluetooth signals?* Even if a user does not connect to any Bluetooth device, apps installed on her device can easily scan Bluetooth advertisements from neighboring devices. *Can an adversary profile a user simply based on the traces of neighboring advertisements? What are the privacy concerns for such Bluetooth traces?*

To answer these questions, we conducted a measurement study and collected traces of Bluetooth advertisements from 49 users over a period of two weeks in Feb-Mar, 2019. Our data collection was consented by the users and was approved by IRB. The participating users were asked to install an Android app written by the authors which periodically collected and uploaded traces of Bluetooth advertisements and device locations. The collected traces, referred to as the campus dataset, were then analyzed to find evidence of privacy leaks. To further validate our findings from the campus dataset, we also analyzed another public Bluetooth dataset [8] (referred to as the external dataset) collected for 6 days in 2013 at a shopping mall. We found several similarities as well as differences across the datasets. We have made the campus dataset publicly available for continuing research [1].

Our data collection and app design were motivated by three key observations about Bluetooth usage on state-of-the-art mobile devices:

A. Minimal permission requirement to use Bluetooth in Android devices. An Android app that uses Bluetooth needs to request two permissions: `permission.BLUETOOTH` and `permission.BLUETOOTH_ADMIN`. Since Android 6.0, these two Bluetooth permissions are defined as normal which means the system automatically grants the app those permissions at install time. With them, an app can proactively discover nearby Bluetooth devices and request to connect to a neighbor. Since Bluetooth discovery may reveal the location of the user, Android requires either `ACCESS_FINE_LOCATION` or `ACCESS_COARSE_LOCATION` to be granted for the Bluetooth functions to work properly. Users are prompted to explicitly allow or deny those two locations permissions when an app is run for the first time. If a user grants either of those permissions, an app can subsequently use any Bluetooth features without notifying the user. This allows a malicious app (that wants to compromise Bluetooth privacy) to masquerade as a legitimate app. For example, an adversarial hiking app could acquire location permission as a legitimate requirement and then scan Bluetooth signals in the background without user's awareness.

B. Easy-to-read information broadcasted in the Bluetooth advertisement. Android uses `bluetooth.le.ScanResult` class to track the

CPSIOTSEC'20, November 9, 2020, Virtual Event, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8087-4/20/11…$15.00
https://doi.org/10.1145/3411498.3419962

information that is retrieved by Bluetooth scanning. From the ScanResult, an app can read various information about a neighboring Bluetooth device, such as, MAC address, name, signal strength, timestamp, whether the device is connectable, etc. By looking at frequencies of various advertisements, an adversary can learn about products that a user may own. For example, if a user frequently encounters a Bluetooth advertisement named "Ford SYNC", it will indicate either the user or her family may own a Ford vehicle. Then this information can be sold to Auto dealers. Since a lot of devices have to advertise easy-to-understand Bluetooth names (e.g. cars, audio devices, PlayStation, XBox, etc.), adversaries can easily associate a user with specific products.

C. Count of Bluetooth advertisements seen by a device over time depends on user movement and is distinguishable. People have different daily movement patterns and as a user moves from one location to another, her surrounding Bluetooth signals change. We note that for a specific place at a certain time, the nearby population does not fluctuate too much. For example, when a user goes to a cafe for lunch, the number of customers in the cafe is relatively stable during lunch hours on different days which means the count of Bluetooth signals observed by a user is also within a specific range. However, Bluetooth counts over a period of time (e.g. 1 week) seen by two different users (with different movement patterns) are distinguishale. We empirically validate this hypothesis.

Based on our analysis of the campus dataset and the external dataset, we find three potential sources of privacy leak from Bluetooth advertisements.

(1) The temporal pattern of Bluetooth advertisement count for each device is distinguishable. An adversary, in possession of Bluetooth traces, can identify a device with high degree of confidence even if the traces are anonymized.
(2) By looking at the frequency of Bluetooth advertisements, an adversary can associate a user with specific products and then send targeted product promotions.
(3) If the same advertisement appears in multiple traces with the same timestamp, we can infer that those devices were located within the Bluetooth range of the advertiser. If any of these devices share its location, all other devices' locations are also revealed (even if the users explicitly disabled their GPS sensors). Our campus dataset shows that such transitive disclosure of location is a real concern.

2 Background

Bluetooth devices operate in two modes, advertiser and scanner. The advertiser mode is a passive mode, where the device waits for incoming connections whereas the scanner is an active mode in which the device actively scans for nearby Bluetooth advertisements and initiates a connection to the advertising device. Bluetooth advertisements typically include device name, signal strength, list of profiles, manufacturer, and a 6-byte MAC address of the advertiser.

In Bluetooth versions later than 4.0 (BLE), the MAC address used in an advertisement can be one of four types [3]: i) Public static MAC address – provides no protection against identity tracking [12], ii) Random static address – updated every time a device reboots, providing some identity protection, iii) Random resolvable address – updated periodically but can be resolved by paired devices using an Identity Resolving Key (IRK); provides significant protection against identity tracking, and iv) Random non-resolvable address: the MAC address can be changed at any time, providing significant protection. Our results indicate that a device may be identified with scan traces even if no MAC addresses are present.

3 Related Work

Because of the popularity of Bluetooth and its use in private (short-range) communications, many studies talk about how to exploit Bluetooth protocol for attacks. The works from [2, 12, 13] focus on attacking the Bluetooth stack to steal messages in transit as well as sensitive personal data stored in smartphones. Aveek *et al.* [5] expose user privacy and identify users by attacking the BLE traffic between a wearable device and a smartphone. BLEB [9] discovers that BLE MAC address protection techniques are rarely used in today's Bluetooth products, so an adversary can easily track a person by reading the MAC address in BLE advertisements. The works above mostly exploit the Bluetooth stack for revealing sensitive data, our work passively collects Bluetooth advertisements of neighboring devices. By analyzing these Bluetooth advertisements, we show how an adversary can compromise user privacy.

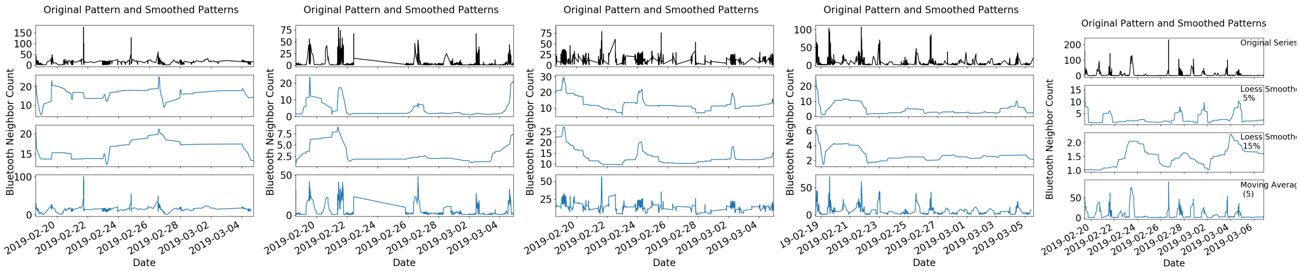
Kassem *et al.* [6] and Korolova *et al.* [10] discuss the feasibility of tracking/identifying a user by scanning Bluetooth signals. show that they can use different apps installed on the same mobile device to uniquely identify a user by comparing the nearby Bluetooth signal names obtained on each of the apps assuming those apps are constantly scanning. In the contrary, we rely on the number of Bluetooth signals. We also demonstrate how to use the Bluetooth signal number as well as using a time-series LSTM model to accurately predict this number in the future.

4 Experimental Setup

4.1 Data Collection

For the purpose of this study, we recruited 49 undergraduate and graduate students from ECE and CS departments at our university. The users were asked to install a custom Android application on their smartphones and run it in the background for 2 weeks. The application (developed by our team) periodically (every minute) scans local Bluetooth signals (advertisements) and uploads the data trace to a cloud server. To perform Bluetooth scanning, the app requires 3 permissions: permission.BLUETOOTH, permission.BLUETOOTH_ADMIN, and ACCESS_COARSE_LOCATION. During every scan, we first list the Bluetooth advertisements from nearby devices. Then for every advertisement, we record its [MAC Address, Device Name, RSSI, Time Stamp]. In addition, the location of the user (i.e. the scanning device) is also recorded as that may be used in other mobility based research. The users use their smartphones as usual and they have the freedom to kill our application, turn off the phone, or uninstall our application to quit the study at any time. We provided monetary incentives to the users according to the amount of data they contribute. However, there may still be some periods during which the cloud server did not receive data from the users. We use linear extrapolation to fill in the missing data.

In total, we received 197,070 data points from all users. The average number of data points received from each user is 4607.10, the median is 3384, and the maximum is 19407. To capture an average user and to eliminate the extremities, we choose users whose contributions lie between the inter quartile range (between 1st and 3rd quartiles) to do detailed analysis. There are 25 users in this range whose results are presented in Section 5.



(a) User 4 (b) User 6 (c) User 7 (d) User 9 (e) User 16

Figure 1: Bluetooth count traces for 5 representative users from the campus dataset. We show the raw traces as well as the smoothed variants (Lowess 5%, Lowess 15%, and moving average).

We also analyzed Bluetooth traces from a publicly available dataset presented in [8]. This dataset consists of traces from 25 employees at a shopping mall over a period of 6 days in 2013. In contrast to the campus dataset which had detailed advertisement information, this only included the timestamps and neighboring MAC addresses. We present the similarities and differences between the results from our campus dataset and the external dataset in Section 6.

4.2 Adversary Model

An adversary is someone who acquires Bluetooth scan traces from multiple devices over a period of time. Since many smartphone applications use Bluetooth, it is possible that some applications collect and save Bluetooth advertisements for future analysis. The adversary can either buy this dataset from app vendors or simply publish an app which masquerade as a useful tool (e.g. a hiking app or a workout app). We assume that the collected traces do not contain personally identifiable attributes, such as IMEI number, or phone number. This lets the suspicious app avoid detection by malware scanners. Moreover, the traces may have partial location information (i.e. some traces contain location but others do not) depending on whether the user enabled location services.

The goal of the adversary is two-fold—i) To profile the user (victim) about her Bluetooth usage and find what devices she may own. ii) To infer the location of the victim (assuming the victim does not have location service enabled) from the trace of a nearby Bluetooth device (which has location enabled). Such information can be used for marketing purposes (i) or for tracking a person's movements (ii).

5 Analysis of Campus Dataset

5.1 Uniqueness of Bluetooth Traces

From the Bluetooth scans we first count the no. of advertisements seen by each device at every timestamp. We refer to this time-series data of advertisement counts as the Bluetooth trace of a device. From the raw traces, we observed that devices may sometimes see spikes in advertisements. While this can be due to the user passing through a crowded area (e.g. a bus stop), it can also happen if bluetooth caches are not cleared in a timely manner. To reduce such sources of noise, we apply 3 different functions to smooth the raw traces. We show Bluetooth traces from 5 representative users in Figure 1. For each of these users, the figure shows the original raw data trace as well as the three smoothed traces. Loess stands for LOcalized regrESSion [4] which is a data regression method that combines the simplicity of linear regression as well as the flexibility of non-linear regression. Its advantage is that it can model complex dataset for which no theoretical models exist. By using `Python statsmodels` package, the Loess function is used to perform Loess regression. It has a control parameter, `fraction` between 0 and 1, that controls the degree of smoothing. The larger the fraction, the smoother the output model. We compare two degrees of smoothness, 5% and 15%, of Loess, and another smoothing method, moving average with the window width of 5. The moving average method simply calculates each y_i by taking the average of $x_{i+1}, \cdots, x_{i+k}$, where $k = 5$ in our test. From Figure 1, it can be seen that moving average does not always remove transient spikes, whereas, 15% Loess over smooths the data. 5% Loess smoothing provides a balance between representing the patterns in the Bluetooth traces and removing transient spikes.

The smoothed temporal patterns for the 25 users is shown Figure 2. It can be observed from the figure that each user's trace is distinguishable from the others and there is little overlap among any two traces. This indicates that a person's Bluetooth trace can be identified with a high degree of confidence even if it is anonymized. Indirectly, this also shows that each user in our dataset has a unique mobility pattern which puts them in the vicinity of different subsets of devices.

5.2 Forecastability

In this section, we show that the forecastability of the Bluetooth patterns in both the raw data and the 5% Loess smoothed data. Then we fit the smoothed data with a LSTM recurrent neural network model to see if we can predict the Bluetooth count. We use Root Mean Squared Error (RMSE) as the metric to measure the prediction accuracy. Intuitively, if the dataset has high predictability, then an adversary can easily profile the user. For example, we may be able to answer the questions: *At what time during the week does user X observe most Bluetooth advertisements?* This information can, in turn, be used to reveal the user's location as described in the next section.

Since the learning task is relatively simple, the neural network has a single LSTM layer with 4 hidden units. The network is shown in Figure 4. y_1, ..., y_n correspond to the predicted values of the Bluetooth signal count at consecutive timestamps. For each user, we divide their data into training and testing datasets in the proportion of 2 to 1. Figure 5 shows the prediction results from the raw dataset and Figure 6 shows the prediction results from the smoothed dataset for 5 representative users. Clearly, the raw dataset is less predictable compared to the smoothed data.

Quantitatively, we summarized the RMSEs for both raw data and smoothed data in Figure 7 for all 25 users. Besides RMSE, we also show the the sample entropy [11] for each user. Sample entropy can quantitatively measure the regularity and unpredictability of

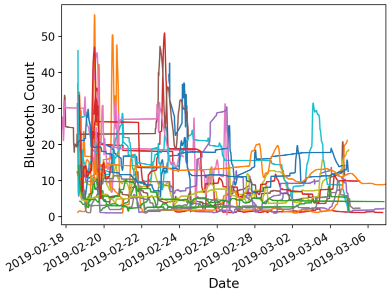


Figure 2: Temporal Bluetooth count patterns of 25 students in the campus dataset. Each line represents the smoothed pattern for one student. Each user has a unique pattern.

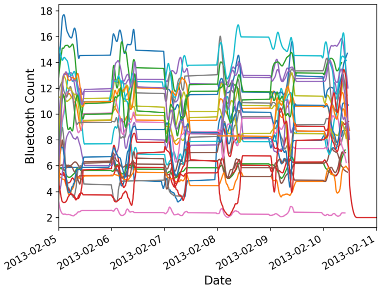


Figure 3: Temporal Bluetooth count patterns of 25 employees in the external dataset. Each line represents the smoothed pattern for one employee.

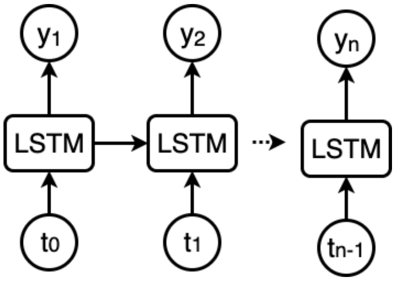
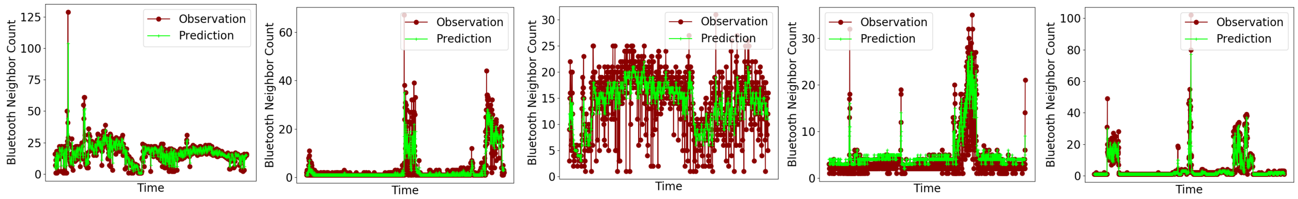
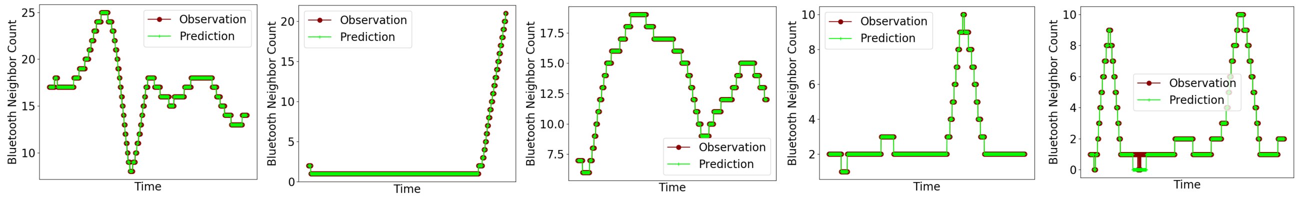


Figure 4: LSTM network for predicting Bluetooth counts. $t_0, ..., t_{n-1}$ are the timestamps and $y_1, ..., y_n$ are the predicted Bluetooth counts at each timestamp.



(a) User 4: RMSE=6.727 (b) User 6: RMSE=3.558 (c) User 7: RMSE=4.798 (d) User 9: RMSE=2.695 (e) User 16: RMSE=0.374

Figure 5: Predicted and the observed values from the raw data trace for the 5 representative users using LSTM model. Prediction errors are shown as RMSE values in the captions.



(a) User 4: RMSE=0.082 (b) User 6: RMSE=0.029 (c) User 7: RMSE=0.047 (d) User 9: RMSE=0.015 (e) User 16: RMSE=0.002

Figure 6: Predicted and the observed values from the smoothed data trace for the 5 representative users using LSTM model. Prediction errors are shown as RMSE values in the captions.

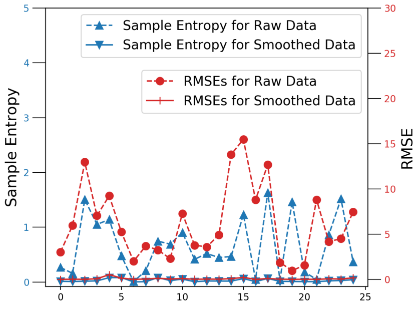


Figure 7: The sample entropy and the RMSE values of the 25 users in the campus dataset. Lower sample entropy means higher predictability.

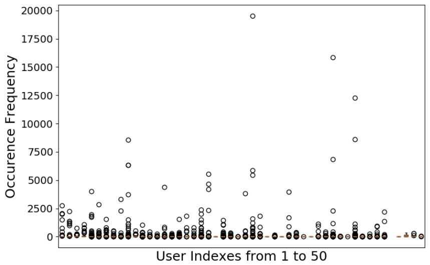


Figure 8: The frequency of different Bluetooth advertisement names observed by each user. Notice that the median frequency is close to 1.

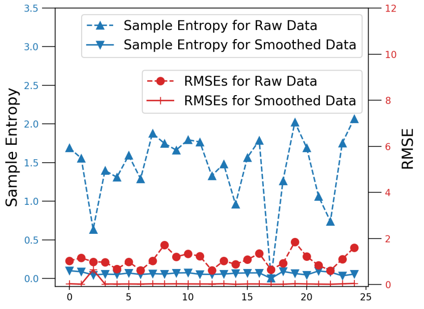


Figure 9: The sample entropy and the prediction RMSE values of the 25 users in the external dataset. Lower sample entropy means higher predictability.

a time series data. The lower the sample entropy, the better the forecastability of the data. We see from Figure 7 that the sample entropy values of the raw data are higher than those of the smoothed data traces. Therefore, the smoothed data is more predictable and still maintains the unique pattern of each user.

5.3 Revealing Personal Information

The frequencies of various advertisements are shown in Figure 8. Although the median frequency is very low, some advertisements appear with very high frequencies. If the frequencies of some advertisements are extremely high (e.g. an advertisement appears

10,000 times for a user), we can infer that the user either owns that device or stays near it. Some typical advertisement names with high frequencies such as `Mi Band 3`, `LE-BOSE`, `Apple Pencil`, and `Alta HR` tell the adversary that the user may have Xiaomi band 3, Bose headset, Apple pencil, or Fitbit AltaHR. By analyzing advertisement frequencies users can be associated with common products. This information can then be used to send targeted product promotions.

In the campus study dataset, we found 2865 unique Bluetooth advertisement names and 1140 pairs of the 49 users encountered at least one common Bluetooth advertisement. To match the advertisements from different traces (devices), we used the advertised device name, MAC address, and the timestamp. If all three fields match, then the advertisements must come from the same device. This also indicates that the devices corresponding to the matching traces were in the bluetooth range of the advertising device. For the pair of users who see common advertisements, if one user's location is exposed, an adversary can easily know the other person's location even if she does not enable location services. This can become a severe issue if a malicious app is installed in a large population like an urban setting. Then many people's location can be exposed without their awareness. Such transitive disclosure of location data is a serious privacy threat.

6 External Dataset

For the external Bluetooth dataset [8], we performed the same analysis as described in Sections 5.1 and 5.2. First, we smoothed the data with 5% Lowess smooting. The patterns are shown in Figure 3. Compared to the campus dataset, the shapes of user traces are more similar in the external dataset. This is because the shopping mall is a more controlled environment, so each employee encounters similar no. of visitors at the same time. In our campus dataset, the users were not limited to any geographic location. If the employees (in external dataset) had collected Bluetooth signals after hours, their traces would have differed more. The absolute Bluetooth counts are still different among users. This can be attributed to the fact that the employees may work at different indoor stores, therefore, they encounter different subsets of the visitors.

Next we calculate the forecastabilities and the RMSEs for each user in the external dataset. The results are summarized in Figure 9. One finding is that, the RMSE values for the shopping mall dataset on average is lower than those for the campus study dataset. This can be due to the fact that the population within the shopping mall was relatively stable. It is unlikely that the population in the shopping mall will change suddenly (and hence fewer spikes in the traces). Whereas, in our campus dataset, the locations of the users are spread over different areas in the town with varying populations (leading to many spikes in the traces). The sample entropy values on average are higher than the campus dataset. This is attributed to the fact that the data collection range is short (6 days) so weekly patterns do not show up.

7 Conclusion

In this paper, we explore how a person's privacy may be compromised by passively analyzing Bluetooth scan traces obtained from her smartphone. First, our results show that the Bluetooth traces from different devices can be easily distinguished, therefore, users can be identified with a high likelihood. Second, by analyzing the frequency of various advertisements, we can infer what products a user may own. Finally, the presence of common Bluetooth advertisements across traces from different devices may leak user location without her knowledge.

Although such privacy leaks are difficult to prevent, we believe one possible mitigation is to insert random noise in the collected advertisements. When an application performs scanning, along with the actual Bluetooth signals nearby, the system can inject other fake advertisements in the results. This will not affect normal application usage because, a user can still find the desired Bluetooth advertisement. But an analysis that relies on the count of Bluetooth devices (such as ours) will produce incorrect results. Note that this mitigation is vulnerable against techniques that decouple the noise from the actual pattern. We suggest that the noise injection randomly alternate between various distributions (e.g. Gaussian distribution, geometric distribution). In this way, denoising will be difficult without prior knowledge.

References

[1] 2019. *Two Weeks Bluetooth Low Energy Dataset.* https://github.com/purdue-dcsl/bluetooth-trace

[2] Wahhab Albazrqaoe, Jun Huang, and Guoliang Xing. 2016. Practical bluetooth traffic sniffing: Systems and privacy implications. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services.* ACM, 333–345.

[3] Inc. Bluetooth SIG. 2015. *Bluetooth Technology Protecting Your Privacy.* https://www.bluetooth.com/blog/bluetooth-technology-protecting-your-privacy/

[4] William S Cleveland. 1979. Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association* 74, 368 (1979), 829–836.

[5] Aveek K Das, Parth H Pathak, Chen-Nee Chuah, and Prasant Mohapatra. 2016. Uncovering privacy leakage in ble network traffic of wearable fitness trackers. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications.* ACM, 99–104.

[6] Kassem Fawaz, Kyu-Han Kim, and Kang G Shin. 2016. Protecting Privacy of {BLE} Device Users. In *25th {USENIX} Security Symposium ({USENIX} Security 16).* 1205–1221.

[7] Inc Fitbit. 2019. *Fitbit.* https://play.google.com/store/apps/details?id=com.fitbit.FitbitMobile&hl=en_US

[8] Adriano Galati and Chris Greenhalgh. 2013. CRAWDAD dataset nottingham/mall (v. 2013-02-05). Downloaded from https://crawdad.org/nottingham/mall/20130205. https://doi.org/10.15783/C7D30D

[9] Taher Issoufaly and Pierre Ugo Tournoux. 2017. BLEB: Bluetooth Low Energy Botnet for large scale individual tracking. In *2017 1st International Conference on Next Generation Computing Applications (NextComp).* IEEE, 115–120.

[10] Aleksandra Korolova and Vinod Sharma. 2018. Cross-App Tracking via Nearby Bluetooth Low Energy Devices. In *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy.* ACM, 43–52.

[11] Joshua S Richman, Douglas E Lake, and J Randall Moorman. 2004. Sample entropy. In *Methods in enzymology.* Vol. 384. Elsevier, 172–184.

[12] Pallavi Sivakumaran and Jorge Blasco. 2019. A Study of the Feasibility of Co-located App Attacks against {BLE} and a Large-Scale Analysis of the Current Application-Layer Security Landscape. In *28th {USENIX} Security Symposium ({USENIX} Security 19).* 1–18.

[13] Fenghao Xu, Wenrui Diao, Zhou Li, Jiongyi Chen, and Kehuan Zhang. 2019. BadBluetooth: Breaking Android Security Mechanisms via Malicious Bluetooth Peripherals.. In *NDSS.*