Pairing Wi-Fi and Bluetooth MAC Addresses Through Passive Packet Capture

Edoardo Longo, Alessandro E C Redondi, Matteo Cesana

Pairing Wi-Fi and Bluetooth MAC Addresses Through Passive Packet Capture

Edoardo Longo, Alessandro E. C. Redondi, Matteo Cesana
DEIB, Politecnico di Milano
Milan, Italy
Email: {name.surname}@polimi.it

***Abstract*—The majority of smart devices used nowadays (e.g., smartphones, laptops, tablets) is capable of both Wi-Fi and Bluetooth wireless communications. Both network interfaces are identified by a unique 48-bits MAC address, assigned during the manufacturing process and unique worldwide. Such addresses, fundamental for link-layer communications and contained in every frame transmitted by the device, can be easily collected through packet sniffing and later used to perform higher level analysis tasks (user tracking, crowd density estimation, etc.). In this work we propose a system to pair the Wi-Fi and Bluetooth MAC addresses belonging to a physical unique device, starting from packets captured through a network of wireless sniffers. We propose several algorithms to perform such a pairing and we evaluate their performance through experiments in a controlled scenario. We show that the proposed algorithms can pair the MAC addresses with good accuracy. The findings of this paper may be useful to improve the precision of indoor localization and crowd density estimation systems and open some questions on the privacy issues of Wi-Fi and Bluetooth enabled devices.**

I. Introduction

The majority of smart mobile devices (smartphones, laptops, tablets, etc.) used nowadays are equipped with both Wi-Fi and Bluetooth wireless communication interfaces. The former type of interface is generally used for internet access, while the latter is the standard for building wireless personal area network (WPAN) and exchanging data over short distances to other devices or communicating with input/output wireless interfaces (keyboards, headphones, etc.).

Both the Wi-Fi and Bluetooth interfaces are identified by 48-bits addresses, known as the Media Access Control (MAC) addresses. Such addresses are unique worldwide, and are embedded into the network hardware during the manufacturing process, or stored in firmware, and designed not to be modified. These addresses are fundamental for link layer communications and are therefore contained in every frame transmitted by a mobile device on either the Wi-Fi or Bluetooth interface. Interestingly, there are some management frames which mobile devices transmit on the Wi-Fi and Bluetooth interfaces even without a proper association with a network and without any user data to deliver. For what concerns Wi-Fi, *probe requests* frames are used to collect information on the network served by access points in range. As for Bluetooth, *inquiry scan* frames are transmitted to discover available devices and their information. Both kind of frames are transmitted without encryption, may be easily captured with cheap off-the-shelves sniffers and a large body of literature has recently studied how to exploit such information to provide higher layer services. Indeed, capturing and analyzing wireless packets carrying a unique device identifier allows to track users in space and time, forming the basis for passive device localization, user behavior estimation, market analysis and many others [1].

ISBN 978-3-903176-05-8

Although in some devices the Wi-Fi and Bluetooth MAC addresses are assigned at the same time during the manufacturing process and are therefore consecutive [2], in the majority of the cases they are completely unrelated. Therefore, in this work we present a system capable of pairing a device Wi-Fi MAC address with its corresponding Bluetooth MAC identifier. We observe that pairing such two different identifiers, linking them to a single physical device may be used in several situations:

- In localization systems, a device can be tracked more precisely by fusing information coming from the two network interfaces [3], or switching between the two depending on the availability. The same applies to pedestrian flow estimation [4], [5] or crowd density estimation systems [6], [7].
- Due to the large spread of Wi-Fi sniffers and probe requests analysis systems [8], [9], device vendors are recently inserting privacy mechanisms to transmit Wi-Fi probe requests with a fake, randomized MAC address [10]. Since this is not done in Bluetooth, pairing the same Bluetooth MAC address with multiple randomized Wi-Fi MAC addresses belonging to the same device basically cracks this privacy mechanism.
- The mechanism can also be used as an adversary weapon: a malicious entity can double is effectiveness by committing blended attacks on both interfaces creating denial of services (DoS) attacks, battery drain attacks or exploiting other vulnerabilities. As an example, Moyers et al. [11] demonstrate that such attacks can accelerate battery depletion by as much as 18.5%.

The proposed system is composed of a network of cheap sniffers in charge of capturing packets, a python script that elaborates the data and different pairing algorithms which exploit the Received Signal Strength Indicator (RSSI) of the captured packets to link the two addresses. The rest of this paper is organized as it follows: Section II presents the reference scenario and the pairing algorithms; experiments are shown in Section III, while Section IV concludes the

manuscript and comments on ongoing and future work.

II. Scenario

The proposed system leverages a network of N synchronized sniffers deployed at fixed and known locations in the environment, equipped with both Wi-Fi and Bluetooth wireless interfaces and able to capture packets transmitted with both technologies. Assume M Wi-Fi and Bluetooth enabled mobile devices are present in the environment. For each packet transmitted by the m-th device on any interface and captured by the n-th sniffer, two pieces of information are extracted and stored in a local database: the Wi-Fi (w_m) or Bluetooth (b_m) address of the transmitting device, and the RSSI r_m (s_m) associated with the captured packet on the Wi-Fi (Bluetooth) interface. The system is operated according to discrete time slots: at the end of each slot, the local information from the N sniffers are periodically transmitted to a central controller, which constructs two $M \times N$ matrices $\mathcal{R}$ and $\mathcal{S}$ where each element $r_{m,n}$ ($s_{m,n}$) contains the average RSSI measurement of the overall Wi-Fi (Bluetooth) packets transmitted by device m and received by the n-th sniffer. Note that each row of $\mathcal{R}$ ($\mathcal{S}$) corresponds to a different Wi-Fi (Bluetooth) MAC address w_m (b_m). In the following, we present different algorithms for pairing each w_m with the corresponding b_m, starting only from the knowledge of matrices $\mathcal{R}$ and $\mathcal{S}$. Two different types of methods are presented: *signal mapping algorithms* and *location-based algorithms*.

A. Signal mapping algorithms

This first class of algorithms maps directly one matrix into the other and performs address pairing according to signal similarity.

1) RSSI normalization: The RSS indicator depends primarily on the transmitter-receiver distance and on the transmitter output power. In our scenario, the Wi-Fi and Bluetooth interfaces of a mobile device are colocated (hence the transmitter-receiver distances are the same), but the output power of the two interfaces may be different. A possible method to remove such difference is to normalize each row of $\mathcal{R}$ and $\mathcal{S}$ to its maximum RSSI value, so that it contains only values between 0 and 1. After normalization, rows of $\mathcal{R}$ and $\mathcal{S}$ can be compared through standard distance metrics (e.g. Euclidean distance, cosine similarity) for finding the best-matching pair. This algorithm is very simple and cheap to execute, and does not require any training.

2) RSSI-to-distance: A more complex approach consists in converting both Wi-Fi and Bluetooth RSS measurements in physical distances. Once converted to distance measurements, the rows of two matrices can be compared. In this work, the well known log-distance path loss model is used to convert a RSSI measurement s into a distance measurement d, that is:

$$r = r_0 - \alpha \log \frac{d}{d_0} \tag{1}$$

where r_0 is the RSSI measured at d_0 meters and α is the path loss exponent. In this work, r_0 and α are estimated a priori for each interface (Wi-Fi or Bluetooth) and independently for several device manufacturers using training data. In details, each transmitter device is located at known distances from a sniffer and RSSI data is collected. The log-distance path loss model is then estimated used least square fitting, as shown in Figure 1(a) and (b).

3) RSSI-to-RSSI: Another option to perform signal mapping is to learn a transformation from one domain to the other. Assuming again that Wi-Fi and Bluetooth signals propagate according to the log-distance path-loss model, we have:

$$r = r_0 - \alpha \log \frac{d}{d_0} \tag{2}$$

$$s = s_0 - \beta \log \frac{d}{d_0}, \tag{3}$$

where r_0, s_0 and α, β capture the difference output powers and the path loss exponents of the Wi-Fi and Bluetooth interfaces, respectively. After some algebra, it is easy to show that:

$$r = (r_0 - \frac{\alpha}{\beta} s_0) + \frac{\alpha}{\beta} s = \theta_0 + \theta_1 s \tag{4}$$

The last equation (4) shows that there exists a linear transformation between Wi-Fi and Bluetooth measurements corresponding to the same device. Therefore, we first learn the linear coefficients θ_0 and θ_1 for each device manufacturer using training data (see Figure 1(c)), and we use them to transform one matrix measurements into the other. Finally, the rows of the two matrices can be compared using standard distance metrics.

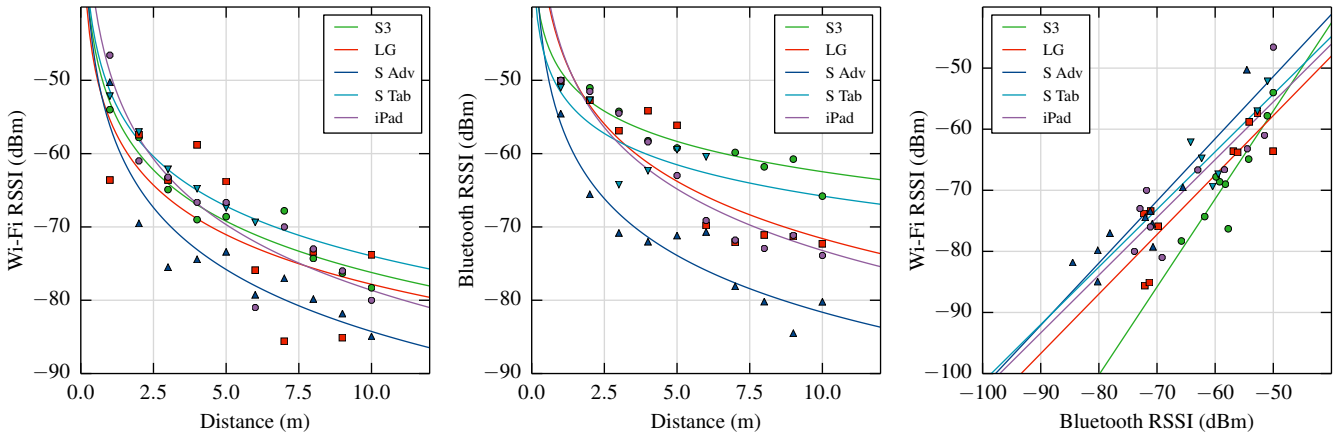
B. Location-based algorithms

The second class of algorithms performs first a localization process, in which each address is assigned to a position in a 2D coordinate system. Address pairing is then performed by linking to each Wi-Fi address its closest Bluetooth address in space.

1) Trilateration: This method builds on the *RSSI-to-distance* algorithm presented in Section II-A2, further processing the estimated distance measurements to obtain a 2D location for each captured device. First, each row of the Wi-Fi measurement matrix $\mathcal{R}$ is converted into distance measurements using the log-distance path loss models learned for each device manufacturer. Each row of the resulting distance matrix is composed of measurements $d_{m,n}$ capturing the distance from the m-th device to the n-th sniffer, whose 2D coordinates are $\mathbf{x}_n$. Localization can be performed solving a trilateration problem of the form:

$$\min_{\mathbf{x}_m} \sum_{n=1}^{N} (\|\mathbf{x}_m - \mathbf{x}_n\|_2 - d_{m,n})^2 \ \forall m, \tag{5}$$

using the gradient descent method. The process is repeated starting from the Bluetooth measurement matrix $\mathcal{S}$. Finally, address pairing is performed comparing the estimated locations.



(a) Wi-Fi RSSI vs distance (b) Bluetooth RSSI vs distance (c) Bluetooth RSSI vs Wi-Fi RSSI

Fig. 1. Models used for training the signal-mapping algorithms

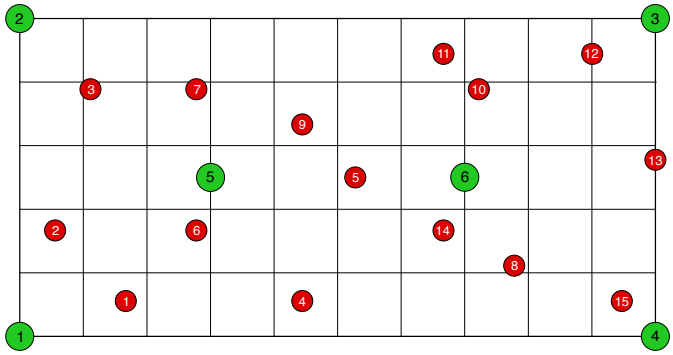


Fig. 2. Planimetry of the experimental environment. In green the six sniffers, in red the fifteen devices positions. The center of each cell, whose area is approximately one square meter, is used as reference point for training the *Fingerprint* algorithm.

2) Fingerprint: The fingerprint method consists of two phases. In a first off-line phase, the system is calibrated by sampling uniformly the environment at p spatial positions. A test device is placed at each position $\mathbf{x}_p$, and two fingerprinting databases are constructed by collecting the Wi-Fi and Bluetooth RSSI measurements received at the sniffers, along with a label indicating the test position. In the on-line phase, upon receiving the RSSI measurements from the sniffer, k-nearest neighbor search is used to retrieve the k closest entries (in signal space) and the corresponding labels. The final location $\mathbf{x}_m$ is retrieved computing a weighed average of the k closest position labels. Weights are chosen to be inversely proportional to the distance (in signal space) between the input fingerprint and the k closest in the database.

III. Experiments

In order to evaluate the performance of the proposed algorithms we carried out an experiment in a controlled scenario. Six sniffer devices are deployed in an indoor environment of approximately 50 square meters, as reported in Figure 2. Each sniffer is built on top of a Raspberry Pi 3 model B board, equipped with Wi-Fi and Bluetooth communication capabilities. The OS running on the machines is Raspbian Jessy version 4.9.24. The system is synchronized using NTP servers and it is remotely controlled through ClusterSSH over the Wi-Fi network. This facilitated the experimenter to have complete control over the whole system remotely. Each sniffer runs a Python script implementing a simple multi-thread architecture with two processes that respectively manage the Wi-Fi and the Bluetooth interfaces. The Wi-Fi sniffing process is handled by Airodump-ng, a Linux network utility used for packet capturing of raw 802.11 frames. The Bluetooth inquiry scanning is done in parallel by hcitool spinq and hcidump, provided by BlueZ, the Linux Bluetooth stack. The software is in charge to run the process, parse the data and upload it on a local MySQL database.

As for the mobile devices, we consider five different mobile devices: Samsung Galaxy S3 mini with CyanogenMode 12 (based on Android Lollipop 5.1.1), LG E450 with Android KitKat 4.1.2, Samsung S Advance with Android KitKat 4.4.2, iPad with iOS 10, and Samsung Galaxy Tab S2 with Android Nougat 7.0. The devices are placed randomly and their positions are changed three times to simulate 15 different positions. All the devices are connected to the same Wi-Fi network, the screens are always active and the Bluetooth interfaces visible. Data is collected for exactly 10 minutes for each device, capturing approximately a Wi-Fi (Bluetooth) packet every second for each device. Such data is used to build the two matrices $\mathcal{R}$ and $\mathcal{S}$.

The two matrices are processed with the algorithms presented in Section II. Each row of matrix $\mathcal{R}$, corresponding to the Wi-Fi address w_m is compared to all rows of matrix $\mathcal{S}$ according to each algorithm logic. Bluetooth addresses b_m are then sorted in ascending order of similarity using Euclidean distance. A top-k approach is then used to perform algorithms evaluation: for each Wi-Fi address w_m, we check if the ground-truth corresponding address b_m is in the first k positions of the ordered list of Bluetooth addresses. We

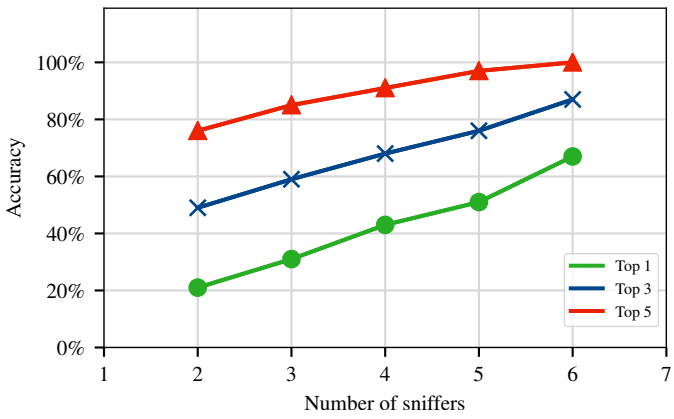


Fig. 4. Top-k {1,3,5} accuracy of the *RSSI-to-distance* algorithm using different sniffers configuration.

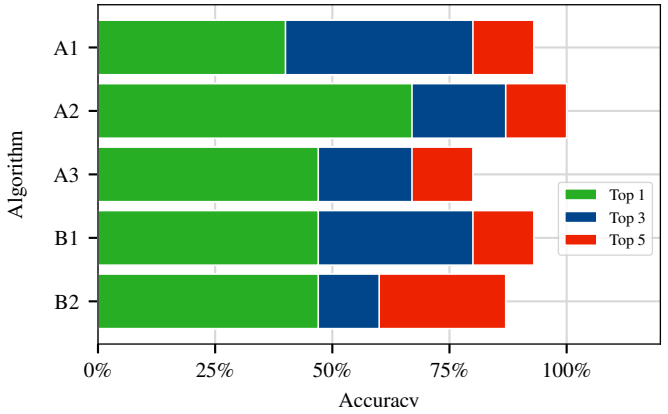


Fig. 3. Algorithms top-k accuracy. A1 - *RSSI Normalization*; A2 - *RSSI-to-distance*; A3 - *RSSI-to-RSSI*; B1 - *Trilateration*; B2 - *Fingerprint*. The percentage indicates the number of correctly paired addresses over the number of tested devices.

then evaluate top-k accuracy as the percentage of correctly paired addresses over all tested devices. Such an approach of measuring accuracy reflects particularly those scenarios in which the user is interested in obtaining a list of possible pairing devices rather than just one (e.g., an attacker who wants to direct its attack on a subset of Bluetooth interfaces possible matching a specific Wi-Fi address).

Figure 3 reports the top-k accuracy of all algorithms, when k is set to be in the range {1,3,5}. As one can see, all algorithms exceed 80% top-5 accuracy and 60% top-3 accuracy. The *RSSI-to-distance* algorithm performs best in all cases. It reaches 100% top 5 accuracy and 87% top 3-accuracy.

We also analyze the accuracy of the best algorithm (*RSSI-to-distance*) when changing the number of deployed sniffers N from 2 to 6. This corresponds to increasing the sniffer density from approximately 1 sniffer every 25 square meters up to 1 sniffer every 8 square meters. For a particular number of sniffers we show the mean accuracy, obtained by averaging the accuracy of the $\binom{6}{N}$ possible combinations of the 6 sniffers shown in Figure 2. As one can see from Figure 4 the number of sniffer devices used impacts on the accuracy of pairing almost linearly. However, with only 2 sniffers the top-5 accuracy of the *RSSI-to-distance* its still around 80%.

IV. Conclusions

In this paper, we have investigated the possibility of pairing a Wi-Fi MAC address and a Bluetooth MAC address of a mobile device in a controlled environment. First, we have presented the sniffers hardware and software architecture. Then, we have described different address pairing algorithms based on two different methodological approaches. The proposed system was tested in a controlled environment achieving promising results. As a future research direction, we plan to (i) validate the results in a real scenario, also dealing with non-static devices, (ii) investigate machine-learning based methods, including other features different from the RSS to perform pairing (e.g., the Bluetooth/Wi-Fi Round Trip Time or the MAC addresses themselves) and (iii) evaluate the performance increase of the pairing algorithms when applied as preprocessing step in localization and crowd estimation systems, or when applied with malicious intentions to attack a device identity or a device battery.

References

[1] A. E. Redondi and M. Cesana, "Building up knowledge through passive Wi-Fi probes," *Computer Communications*, vol. 117, pp. 1–12, 2018.

[2] J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, and D. Brown, "A study of MAC address randomization in mobile devices and when it fails," *CoRR*, vol. abs/1703.02874, 2017. [Online]. Available: http://arxiv.org/abs/1703.02874

[3] Z. Jindan, Z. Kai, K.-H. Kim, and P. Mohapatra, "Improving crowd-sourced Wi-Fi localization systems using Bluetooth beacons," *9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2012.

[4] F. M. Naini, O. Dousse, P. Thiran, and M. Vetterli, "Population size estimation using a few individuals as agents," in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 2499–2503.

[5] L. Schauer, M. Werner, and P. Marcus, "Estimating crowd densities and pedestrian flows using Wi-Fi and Bluetooth," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2014, pp. 171–177.

[6] J. Weppner, P. Lukowicz, U. Blanke, and G. Tröster, "Participatory Bluetooth scans serving as urban crowd probes," *IEEE Sensors Journal*, vol. 14, no. 12, pp. 4196–4206, 2014.

[7] D. Bullock, R. Haseman, J. Wasson, and R. Spitler, "Anonymous Bluetooth probes for airport security line service time measurement: the Indianapolis pilot deployment," in *89th Annual Meeting in Transportation Research Board*, 2010.

[8] M. V. Barbera, A. Epasto, A. Mei, V. C. Perta, and J. Stefa, "Signals from the crowd: Uncovering social relationships through smartphone probes," in *Proceedings of the 2013 Conference on Internet Measurement Conference*, ser. IMC '13. New York, NY, USA: ACM, 2013, pp. 265–276. [Online]. Available: http://doi.acm.org/10.1145/2504730.2504742

[9] M. Cunche, "I know your MAC Address: Targeted tracking of individual using Wi-Fi," in *International Symposium on Research in Grey-Hat Hacking - GreHack*, Grenoble, France, Nov. 2013. [Online]. Available: https://hal.inria.fr/hal-00858324

[10] M. Vanhoef, C. Matte, M. Cunche, L. S. Cardoso, and F. Piessens, "Why MAC address randomization is not enough: An analysis of Wi-Fi network discovery mechanisms," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. ACM, 2016, pp. 413–424.

[11] B. R. Moyers, J. P. Dunning, R. C. Marchany, and J. G. Tront, "Effects of Wi-Fi and Bluetooth battery exhaustion attacks on mobile devices," in *2010 43rd Hawaii International Conference on System Sciences*. IEEE, 2010. [Online]. Available: https://doi.org/10.1109/hicss.2010.170