Fingerprinting connected Wi-Fi devices using per-network MAC addresses

Abhishek Kumar Mishra, Samuel Pélissier, Mathieu Cunche

Fingerprinting connected Wi-Fi devices using per-network MAC addresses

Abhishek Kumar Mishra[1], Samuel Pélissier[1], and Mathieu Cunche[1]

INSA-Lyon, Inria, University of Lyon, CITI Lab.
{abhishek-kumar.mishra,samuel.pelissier,mathieu.cunche}@insa-lyon.fr

Abstract. Wi-Fi stands out as one of the most prominent and widespread wireless technologies in use today. Smartphones and various other Wi-Fi-enabled devices employ management frames called probe-requests to discover nearby networks.
In this study, we reveal that it is possible to fingerprint based on the probe-requests they emit while connected to a network. Leveraging distinctive features of probe-request bursts we use a Random Forest-based approach to successfully fingerprint devices. This demonstrate that device randomizing their MAC addresses between networks can still be tracked. Through an assessment conducted on a real-world measurement comprising Wi-Fi devices with diverse operating systems, and spanning a month duration, we demonstrate that our model fingerprints individual devices with ∼40% accuracy with 1 burst and perfect re-identification if two or more bursts are available.

Keywords: Wi-Fi, Probe-requests, Privacy, Persistent MACs, Fingerprinting, Measurement

1 Introduction

The proliferation of Wi-Fi enabled devices has facilitated applications such as user trajectory tracking and pedestrian flow estimation [5, 6]. Conversely, there is a growing concern regarding user privacy stemming from issues related to user anonymity and device traceability through Wi-Fi sensing [7].

Contemporary devices equipped with Wi-Fi capabilities utilize the active scan method, a prominent technique within the Wi-Fi protocol standard, to discover nearby networks. During these active scans, mobile devices emit management frames known as probe-requests to locate nearby Access Points (APs). Intercepting probe-requests is relatively straightforward and can be leveraged to track users thanks to the MAC address exposed in those frames.

To mitigate obvious privacy issues induced by MAC-based device tracking, vendors have implemented countermeasures such as MAC address randomization [9]. As a result, non-connected devices change the address in their probe request frequently (e.g. at every probe burst or every 15 minutes). Randomizing addresses while a device is not connected is straightforward. But as soon as the

device connects to a network, it must keep the same address all along the session to communicate with the access point. As a result, the address can only be changed at each connection, by using a *per-network*[1] random address [12].

Devices in the connected state probe using their per-network MAC address, either to find better networks or respond to location-based service requests. This study demonstrates that the probing behaviour of connected Wi-Fi devices can be used to fingerprint them and thus defeat *per-network* MAC addresses randomization. Furthermore, we show that the observation of only a handful probe-requests observation is enough to derive a unique fingerprint. Consequently, those devices, and their users, can be successfully tracked across their networks and session(cf. Section 3.2) despite address randomization schemes.

We examine various device-specific attributes, including the timing of bursts, the duration of advertised randomized MAC addresses, and the content of probe requests. This study represents the first attempt, to our knowledge, focusing solely on *per-network* MAC addresses and demonstrating successful fingerprinting of observed devices.

The contributions of this paper are as follows:

- We identify various information from passively captured probe-requests that exhibit distinct behavior for devices in the connected state.
- Using the described features, we present a machine-learning method that utilizes a device's per-network MAC addresses solely from its probe-requests to create efficient fingerprints.
- Through a month-long measurement involving various devices, we show that robust fingerprinting is attainable with just two or more bursts. Our findings suggest that devices should randomize their MAC addresses per burst even while connected to prevent tracking.

2 Background

In the upcoming section, we delve into the active scanning process in Wi-Fi, with a specific emphasis on probe-request messages. Our examination covers temporal patterns, content, and the use of random MAC addresses.

2.1 Wi-Fi Active Scanning

Devices with Wi-Fi capabilities utilize active scanning to identify nearby wireless networks, commonly known as APs [1]. In the active scanning process, mobile devices explore accessible networks by sending out management frames known as probe-request frames.

When an AP detects a probe-request frame that matches its Service Set Identifier (SSID) or advertises a wildcard SSID, it responds by sending a probe-response frame. The probe-response is a unicast message directly addressed to

[1] In Android, the randomized MAC address is bound to the SSID, while iOS binds it to the BSSID.

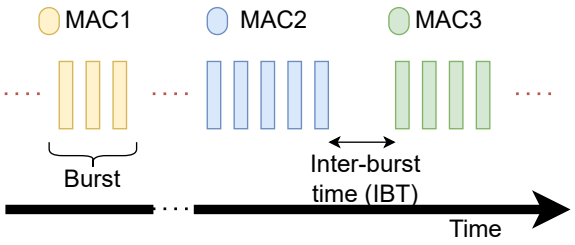


Fig. 1: Wi-Fi active scanning.

the requesting client. Upon receiving probe-response frames from nearby access points, the client can evaluate its choices and select a network to connect to based on criteria such as signal strength, security settings, and user preferences.

To conserve energy, devices periodically broadcast probe-request frames. Figure 1 depicts the active scanning process over time for a Wi-Fi device. Mobile devices repeatedly send probe-requests on available channels to receive responses from all nearby access points. Each device conducts multiple rounds of active scanning across the available channels.

The information element (IE) field in a probe-request frame enables devices to communicate their capabilities and connection preferences, crucial for the association process. The content within IE fields may be potentially unique to a specific device or its current state [10].

2.2 Per-network MAC randomization

To counter tracking issues, vendors have implemented address randomization while performing active scanning: the address field in the probe-request is periodically changed for a random value. As illustrated in Figure1, this change can done as often as every burst. This randomization can be straightforwardly applied when the device is not connected to a network; but as soon the device connects to an access point, it must keep the same address for the duration of the connection.

Nevertheless, to provide a minimum level of protection, vendors have enforced *per-network* random addresses: a distinct random address is used for each network [12]. With the current implementation, all the frames, including the probe requests, are using the same address while the device is connected.

3 Dataset and Threat model

In this section, we begin by examining the dataset under investigation, followed by a description of the threat model we consider. We then build upon the described threat model to introduce an attack successfully linking per-network MAC addresses. Finally, we introduce a new scheme for randomizing MAC addresses efficiently in Wi-Fi.

3.1 Dataset

We utilize an extensive dataset, named `UJI dataset` for the investigation of per-network MAC addresses, released in 2024. The `UJI dataset` is the first long-term public probe request trace which also contains randomized MAC addresses [2]. The dataset contains more than 1.4 million (1,410,834) probe-requests, originating from a variety of Wi-Fi-enabled devices.

The dataset was collected at the University of Jaume I, Spain. It captures a realistic office scenario with a dynamic environment that features up to 30 individuals frequently entering and leaving the office Wi-Fi network.

The office space is rectangular, measuring approximately 16.71m in length and 10.76m in width. The dataset was created in March 2023 to include regular work weeks, weekends, and special events. The collection period coincided with a local holiday week as well capturing an actual real-world monitoring scenario of a workplace that an adversary might be keen on tracking.

3.2 Threat model

The attacker's aim is to successfully fingerprint a device to track its user across network connections. We consider a passive attacker with the ability to passively collect Wi-Fi frames in a targeted area, potentially spanning extended periods. In this scenario, we assume target devices are unmodified, and the attacker lacks physical access to them. The attacker simply listens to emitted probe-requests on various frequency bands.

To collect the data, the attacker can be located in an area where the targeted devices connects to a known network. Alternatively, the attacker can use techniques such as Evil-Twin attacks to trigger a connection [4] even if there is no known network in range.

4 Fingerprinting connected Wi-Fi devices

In this section, we first extract the potential MACs that are per-network from the set of all MACs observed in probe-requests. Then, we select features to fingerprint the devices emitting them. We conclude by selecting a machine-learning-based model that successfully fingerprints Wi-Fi devices.

4.1 Separating per-network MACs

`UJI` dataset contains probe-requests sent by devices in both connected as well as non-connected states. In this work, we only focus on the per-network randomized addresses that were transmitted by devices when already connected to the network.

We separate per-network MACs from the pool of observed ones utilizing two thresholds on the minimum sojourn time. We set the bound of 6 hours (T) to cover cases when devices change their per-network MACs every day (e.g. in `Android` under the developer mode [2]).

[2] https://source.android.com/docs/core/connect/wifi-mac-randomization-behavior

We apply the threshold T over our dataset and illustrate the behavior of per-network MAC addresses in Figure 2. Our analysis reveals the presence of 927 persistent per-network addresses.

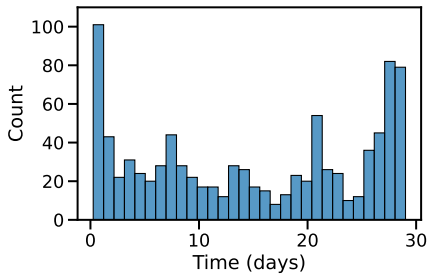


Fig. 2: Per-network MACs

4.2 Characterizing per-network MACS

We start our investigation by considering five features (cf. Table 1) that characterize the behavior of a smartphone's probe-request burst, which could be classified into three broad categories:

Metric	*Feaure*	*Notation*
Time-based	Mean burst duration	μ^{T_b}
	Mean inter-burst time	μ^{IBT}
Content-based	Mean num. of present IE fields	$\mu^{N_{ie}}$
Behavior-based	Sojourn time of burst's MAC	T_{mac}
	Mean hour of probing	μ^{H}

Table 1: Considered features.

1. Time-based features: We select two such features.

- *Mean burst duration (μ^{T_b}):* μ^{T_b} measures the average duration for which bursts are observed at the receiving sniffer for a particular MAC address.
- *Mean inter-burst time (μ^{IBT}):* The mean time gap between two successive probe-request bursts from a device is denoted by μ^{IBT}.

2. Content-based features: We choose the *mean num. of present IE fields ($\mu^{N_{ie}}$)*. The probe-requests have IE element fields that contain information about

the device's capabilities and preferences. Out of the 256 specific elements that a smartphone could advertise, many of them are not included in practice. $\mu^{N_{ie}}$ denotes the average number of non-empty IE fields for a random frame chosen from each of the bursts with a particular MAC.

3. Behavior-based features: We select two behavior-based features from the extracted bursts.

- *Sojourn time of burst's MAC (T_{mac}):* T_{mac} denotes the duration for which a particular MAC address is observed.
- *Mean hour of probing (μ^H):* μ^H is the average hour of the day when a particular per-network MAC is seen in the dataset.

As depicted in Figure 3a, distinct differential probe-request burst behavior is observed across various per-network MAC addresses in the dataset. Bursts are sent with distinct temporal and content-wise behavior for different states for all five features.

For instance, the figure shows that bursts display varied frequency and duration across devices. Probing in the associated state is also quite discriminating in terms of content as a variety of advertised IE fields is available. The behavior of user devices in terms of the time of the day at which they start probing also varies widely. These findings hold across all per-network MACs that are there in `UJI dataset`. The differential behavior can be attributed to the heterogeneity of devices, their states, as well as the user itself. Subsequently, significant variance in the data motivated us to study the unique trends in burst behavior for fingerprinting devices using per-network MAC addresses.

4.3 Model and Feature selection

Per-network MAC prediction can be analyzed as a multi-class classification problem. The input is the features extracted from a particular probe-request burst, while the output associates it with a certain per-network MAC address. For efficiently using features from the set $\{\mu^{T_b}, \mu^{IBT}, \mu^{N_{ie}}, T_{mac}, \mu^H\}$, we utilize a Random Forest (RF) based model. RF is fast, robust to outliers, can identify non-linear patterns, and, does not suffer from overfitting even if more trees are appended [3].

5 Fingerprint's performance

In this section, we first define our evaluation methodology before showcasing the robustness of device fingerprinting using per-network randomized MAC addresses.

5.1 Evaluation Methodology

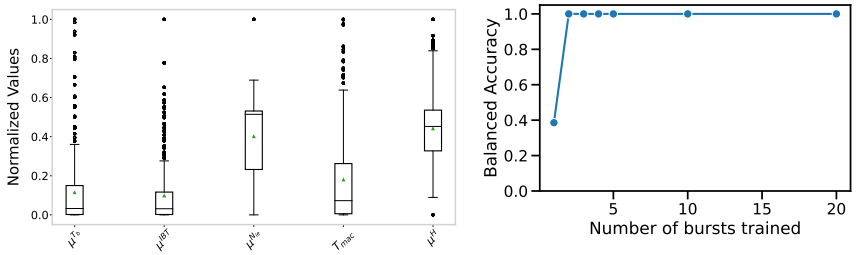
We split the probe-requests from each device seen in the dataset into individual bursts by separating frame sequences that have an inter-frame duration longer

than 1s. We only consider bursts with multiple frames containing the already selected MAC address. We compute the time, content, and, behavior-based metrics to train the model. The model takes bursts (b_n) for training as well as the input for prediction.

We train the model on bursts with per-network randomized MACs in `UJI` dataset. To obtain a robust model against unseen data bursts in the dataset are split into two subsets: the first `p` bursts are used for training, and the remaining bursts are only exploited during the testing phase. We train the model on `UJI dataset` and utilize the trained model to predict per-network MAC addresses on respective datasets. We use the `scikit-learn` [8] Python library[3], which provides the implementation of the Random Forest model.

5.2 Results

We use balanced accuracy as the measure to evaluate the effectiveness of the extracted fingerprints. The balanced accuracy is defined as the average recall obtained in each of the per-network MAC predictions. We observe in Figure 3b that beyond two observed bursts, we can perfectly fingerprint each of the per-network MAC addresses seen in our dataset. For a single burst, the accuracy drastically drops to ∼40%.



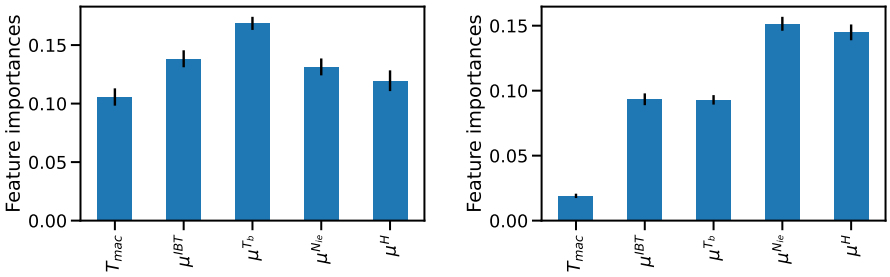
(a) Distinguishing behaviour of features (b) Accuracy of device fingerprints.

Fig. 3: Investigating features and accuracy.

We next look at the impact of each feature on device fingerprinting utilizing *permutation feature importance* [4]. This approach evaluates the influence of individual features on a model's statistical performance. It involves randomly shuffling the values of a single feature and measuring the resulting decrease in the model's performance. This manipulation of the feature-target relationship exposes the degree to which the model relies on that particular feature.

As we illustrate in Figure 4, the relative importance of each feature is uniform for short-term training of 5 bursts (selected to have a stable contribution of

[3] https://scikit-learn.org/stable/index.html (version 1.3.2)
[4] https://scikit-learn.org/stable/modules/permutation_importance.html



(a) Short-term training (5 bursts). (b) Long-term training (20 bursts).

Fig. 4: Permutation feature importance.

all features). The time-based features (μ^{T_b} and μ^{IBT}) do contribute more to fingerprinting accuracy. On the other hand, when undergoing long-term training of 20 bursts, user behavior-based features like the mean hour of probing (μ^{H}) naturally contribute highly towards the achieved balanced accuracy.

5.3 Discussion

Based on the presented results, we conclude that even in the associated mode, MAC addresses must change every burst to protect users from getting tracked. With a per-burst identifier, adversaries will find it hard to obtain effective fingerprints for a particular device.

Per-network MAC address randomization in the connected state could be implemented using a mechanism similar to the one currently employed for non-connected MAC randomization. However, this implementation may not be entirely straightforward due to concurrent data traffic along with the probe requests. Thus, careful consideration and testing are necessary to ensure seamless integration without disrupting existing functionalities, while protecting against known attacks (e.g. replay) [11].

6 Conclusion

We demonstrate the possibility of precisely fingerprinting connected Wi-Fi devices solely through passive monitoring of Wi-Fi probe-requests. Devices in the connected state often broadcast a *per-network* randomized MAC address, making them susceptible to fingerprinting and subsequent tracking. By analyzing probe-request burst behaviors, we identify key features that differ among devices. Our Random Forest based model achieves high classification accuracy in fingerprinting when observing a device in the connected state for two or more bursts. Therefore, we advocate for the extension of per-burst MAC randomization to the case of connected devices.

References

1. Ieee standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications - redline. IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016) - Redline (2021)
2. Bravenec, T., Torres-Sospedra, J., Gould, M., Fryza, T.: Uji probes revisited: Deeper dive into the dataset of wi-fi probe requests. IEEE Journal of Indoor and Seamless Positioning and Navigation (2023)
3. Chaudhary, A., Kolhe, S., Kamal, R.: An improved random forest classifier for multi-class classification. Information Processing in Agriculture **3**(4), 215–222 (2016)
4. Fenske, E., Brown, D., Martin, J., Mayberry, T., Ryan, P., Rye, E.C.: Three years later: A study of mac address randomization in mobile devices and when it succeeds. Proc. Priv. Enhancing Technol. (2021)
5. Huang, B., Mao, G., Qin, Y., Wei, Y.: Pedestrian flow estimation through passive WiFi sensing. IEEE Transactions on Mobile Computing **20**(4), 1529–1542 (2021). https://doi.org/10.1109/TMC.2019.2959610
6. Koh, Z., Zhou, Y., Lau, B.P.L., Yuen, C., Tuncer, B., Chong, K.H.: Multiple-perspective clustering of passive Wi-Fi sensing trajectory data. IEEE Transactions on Big Data pp. 1–1 (2020). https://doi.org/10.1109/TBDATA.2020.3045154
7. Mishra, A.K., Carneiro Viana, A., Achir, N., Palamidessi, C.: Public wireless packets anonymously hurt you. In: 2021 IEEE 46th Conference on Local Computer Networks (LCN). pp. 649–652 (2021)
8. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. the Journal of machine Learning research
9. Vanhoef, M., Matte, C., Cunche, M., Cardoso, L.S., Piessens, F.: Why mac address randomization is not enough: An analysis of wi-fi network discovery mechanisms. In: ACM on Asia CCS. p. 413–424. ASIA CCS '16, Association for Computing Machinery, New York, NY, USA (2016)
10. Vanhoef, M., Matte, C., Cunche, M., Cardoso, L.S., Piessens, F.: 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. pp. 413–424 (2016)
11. Zhang, Y., Lin, Z.: When good becomes evil: Tracking bluetooth low energy devices via allowlist-based side channel and its countermeasure. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. p. 3181–3194. CCS '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3548606.3559372, https://doi.org/10.1145/3548606.3559372
12. Zúñiga, J.C., Bernardos, C.J., Andersdotter, A.: Randomized and Changing MAC Address state of affairs. Internet-Draft draft-ietf-madinas-mac-address-randomization-12, Internet Engineering Task Force (Feb 2024), https://datatracker.ietf.org/doc/draft-ietf-madinas-mac-address-randomization/12/, work in Progress