Bleach: From WiFi probe-request signatures to MAC association

Abhishek Kumar Mishra, Aline Carneiro Viana, Nadjib Achir

Bleach: From WiFi probe-request signatures to MAC association

Abhishek Kumar Mishra[a], Aline Carneiro Viana[a], Nadjib Achir[a,b]

[a] *Inria, 1 rue Honore d'Estienne d'Orves. Campus de l'Ecole Polytechnique, Palaiseau, 91120, France*
[b] *University Sorbonne Paris Nord, 99 Avenue Jean-Baptiste Clément, Villetaneuse, 93430, France*

Abstract

Smartphones or similar WiFi-enabled devices regularly discover nearby access points by broadcasting management frames known as probe-requests. Probe-request frames relay, as information, the MAC addresses of sending devices, which act as the device identifiers. To protect the user's privacy and location, probe-requests use a randomized MAC address generated according to the MAC address randomization protocol. Unfortunately, MAC randomization greatly limits any studies on trajectory inference, flow estimation, crowd counting, etc. To overcome this limitation while respecting users' privacy, we propose `Bleach`, a novel, efficient, and comprehensive approach allowing randomized MAC addresses to device association from probe-requests. `Bleach` models the frame association as a resolution of MAC conflicts in small time intervals. We use time and frame content-based signatures to resolve and associate MACs inside a conflict. We propose a novel MAC association algorithm involving logistic regression using signatures and our introduced time metric. To the best of our knowledge, this is the first work that formulates the probe-request association problem as a generic resolution of conflicts and benchmarks the association concerning several datasets. Our results show that `Bleach` outperforms the state-of-the-art schemes in terms of accuracy (as high as 99%) and robustness to a wide range of input probe-request datasets.

Keywords: WiFi, probe-requests, MAC address randomization, Frame Association
PACS: 0000, 1111
2000 MSC: 0000, 1111

1. Introduction

Modern WiFi-enabled devices find nearby networks using one of the prominent methods in the WiFi protocol standard called *active scan*. In active scans, mobile devices broadcast management frames called *probe-requests*, which could contain physical (true) MAC addresses that reveal their identity. Legacy devices transmit their true MAC address in probe-requests. To protect user privacy, the WiFi standard strictly recommends mobile devices change (randomize) their true MAC periodically. This reduces the correlation between probe-requests (with unique MAC addresses) and the emitters [1].

MAC address randomization disrupts the continuity and semantics of probe-requests and breaks the network data collection and analysis process. While this mechanism protects the user's privacy, it impacts the continuity and accuracy of crucial works and strategies relying on MAC addresses as user-device identifiers. Some of these works include user trajectory inference [2, 3, 4] and crowd flow estimation [5][6], bringing understanding of urban space usability, benefiting extended reality or pervasive computing applications, or improving traffic management or disaster response [7].

Further domains relying on the continuity of WiFi MAC addresses include network security and intrusion detection by identifying devices and monitoring MAC address patterns [8]. In location-based services, it improves indoor positioning accuracy and delivers personalized content [9]. User behavior analytics benefit from analyzing foot traffic and crowd dynamics for improved customer experiences and safety [10]. In smart cities and IoT, it manages connected devices and supports reliable data streams [11]. Personalized user experiences in smart homes and connected vehicles are enhanced through seamless connectivity [12]. Network management is optimized by understanding device mobility and efficiently allocating resources [13]. Law enforcement and public safety use it to track devices involved in criminal activities and monitor movements in sensitive areas [14]. It tracks patients and medical equipment in healthcare, ensuring connectivity and monitoring wearable health devices [15].

To address continuity and accuracy issues, recent research extensively explores *MAC address association*, which involves linking (associating) randomized MAC addresses emitted by a specific device. Current frameworks claim high accuracy in de-randomizing MAC addresses in their evaluation datasets. For example, [16] achieves over 80% accuracy in a shopping mall, while [17] reports up to 75% accuracy in laboratory settings. [18] achieves 67.6% to 80% uniqueness for 50 to 100 devices in music festivals and lab scenarios.

These frameworks extract *signatures* from probe-request frames, capturing unique device features for MAC association. Signatures are derived from: i) sequence numbers (SEQ) [19, 16], ii) fields like information elements (IE) [20, 18, 16], iii) timing patterns like inter-burst time (IBT) [17], and iv) RSSI values [16].

Despite their promise, these frameworks lack reliability across different validation datasets. We define *reliability* as the consistent ability to accurately identify MAC addresses from the same device regardless of the contextual scenario. Our observations show significant performance discrepancies in varied environments, highlighting the need for a more reliable approach and a robust algorithm.

Challenges arise in densely populated scenarios with frequent MAC address changes, reducing the effectiveness of current signatures. In response, we present `Bleach` (cf. Section 4), a novel framework that ensures robust association accuracy, efficient runtime, and deployment. It performs well even with numerous simultaneous MAC address changes.

`Bleach` operates through four steps: i) Partitioning input datasets that describe the MAC addresses of devices observed within a particular zone into what we term "MAC trails." These MAC trails are bursts of probe-request frames associated with specific MAC addresses. ii) Identifying and characterizing conflicts between appearing and disappearing MAC trails. iii) Extracting and evaluating signatures from these trails, ensuring their effectiveness across diverse datasets. iv) Implementing a novel MAC association algorithm to correlate randomized MAC addresses.

These steps ensure `Bleach`'s improved performance compared to the state-of-the-art. In summary, the major contributions of the paper are as follows.

1. In Sections 2 and 3, we investigate and identify reliability issues in current address association frameworks, emphasizing the need for awareness and improvement. Section 6 characterizes MAC association to enhance understanding and enable better comparisons among existing and future works.

2. We introduce metrics to evaluate signatures and identify effective time- and frame-based signatures for correlating probes with randomized MACs from a single device (cf. Section 7).

3. `Bleach` associates randomized probe requests based on obtained signatures and their distance metrics (cf. Section 8). We evaluate our framework across various scenarios with differing degrees of observed *conflicts*

(cf. Section 9), and we predict the performance of association frameworks on new datasets.

4. We demonstrate that `Bleach` is effective even in highly complex scenarios with extensive MAC address changes in the sniffing zone.

We plan to release the open-source code of our framework on usage demo and potentially a few anatomized datasets. Finally, we conclude the work and discuss future perspectives in Section 10.

2. Background and State of the Art

In the subsequent section, we will explore the fundamentals of WiFi active scanning and MAC randomization within current WiFi standards. Additionally, we will review the existing literature on association frameworks.

2.1. WiFi Active Scanning and MAC randomization

WiFi-enabled devices use active scanning to locate nearby wireless networks, or *access points* (APs) [21]. During active scanning, devices send *probe-request* frames. APs respond with probe-response frames if the probe-request matches their Service Set Identifier (SSID) or a wildcard SSID. These unicast responses help the device evaluate available networks based on signal strength, security settings, and user preferences.

Probe-request frames are periodically broadcasted to conserve energy. Fig. 1 shows the active scanning process over time, where devices send probe-requests on available channels and receive responses from accessible APs, performing multiple rounds of active scanning.

Active scanning rounds last a few seconds, depending on the number of known access points and non-busy channels. As shown in Fig. 1, multiple rounds contain *bursts* of probe-requests captured by the sniffer, with the MAC address of individual probes within a burst remaining consistent. However, the MAC address is likely to change (randomize) in subsequent bursts, a process known as MAC randomization [Section 12.2.10, [21]]. The longer it takes for a device to discover a network, the more probes will circulate, increasing the number of randomized MACs.

The number of bursts advertising a certain MAC address varies and depends on the manufacturer and the device's state. The inter-burst time (IBT)

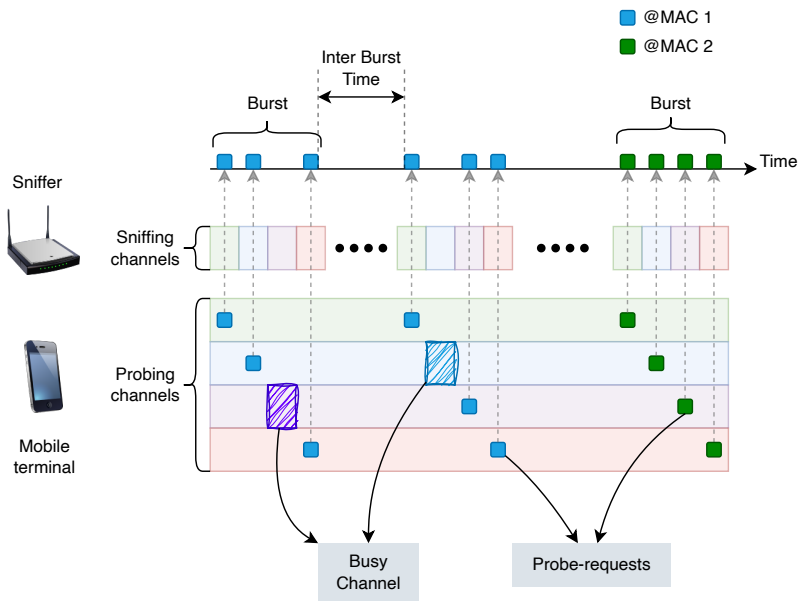


Figure 1: A device's randomised probe-requests.

between successive bursts also varies by manufacturer. Most modern WiFi devices use randomized addresses instead of their physical or *true* MAC address, while legacy devices may still broadcast their true MAC addresses.

As discussed in the subsequent subsection, the literature suggests that randomized MAC addresses in probe-requests can be correlated to the sender device using various attack methodologies, a process called MAC address association.

2.2. Issues in MAC randomization

The current implementation of MAC randomization is susceptible to certain vulnerabilities, that lead to MAC association, arising from:

1. **Inference from temporal behaviour:** The time-interval patterns between probe-requests with randomized MACs could help point them to specific devices.

2. **Inference from spatial behaviour:** If randomized MAC addresses are frequently observed broadcasting a certain SSID or near a specific location, they can be linked together.

3. **Behavior of MAC addresses:** The inconsistency or specific patterns in the implementation of MAC address randomization can give hints for finding links between randomized probe-requests and a user device.

4. **Inference from contextual information:** The contextual information like the name of devices and IP addresses can help associate randomized MAC addresses.

5. **Content of probe-requests:** The information contained in probe-requests, such as SSIDs that the user-device is willing to connect helps link various randomized MACs from the same device.

6. **Behavior of users:** We can investigate the behaviour of users such as their times when they connect their devices. Moreover, users' frequent visits could potentially assist in MAC association

Manufacturers progressively change and adapt their MAC address randomization methods. This might lead to changing the effectiveness of MAC association frameworks across device populations. Next, we look at the related works in the MAC association before checking their effectiveness and identifying the current shortcomings in Section 3.

2.3. MAC association literature

For MAC associations, current solutions explore different avenues to understand and manipulate the associations between randomized MAC addresses. Address association in the literature relies on two primary approaches: i) identifying vulnerabilities (*leaks*) in system design or protocols, and ii) extracting device-specific *signatures* from the probe-request transmission. We compare literature association frameworks and the novel framework introduced in this paper, `Bleach` in Table 1. In the following, we investigate the works mentioned in Table 1 based on the adopted strategy:

- Information leaks: Early studies on address association leverage *information leaks* in protocols or system designs to establish links between randomized MAC addresses and specific devices. For instance, both [20] and [22] engage in reverse engineering of probe-request Universally Unique Identifier-Enrollees (UUID-Es) to identify true MAC addresses using precomputed hash tables. In [20], the authors exploit certain devices' auto-connect feature to connect to SSIDs with popular but potentially malicious names, which might inadvertently reveal their true MAC address. Some devices assign consecutive MAC addresses

Table 1: MAC association frameworks' comparison.

Framework	Information leaks	Signatures				Evaluation	
		Information element	Temporal	Sequence number	RSSI	Many contextual scenarios	Benchmarks
[20], [22], [23]	✓					✗	✗
[20], [18], [24], [25]		✓				✗	✗
[20], [26], [27]			✓			✗	✗
[16, 28]		✓		✓	✓	✗	✗
[29]		✓	✓	✓	✓	✗	✗
Bleach		✓	✓	✓		✓	✓

for BLE and WiFi, creating an opportunity to unveil the WiFi MAC address [23]. It is important to note that these methods are not universal and depend on vulnerabilities in the system design, which manufacturers typically rectify once brought to their attention.

- Signatures: The second approach relies on identifying *signatures*, which consist of metrics extracted from probe-request transmissions to differentiate devices. Recent frameworks use four key metrics for generating these signatures.

The first metric analyzes the information element (IE) field in probe-requests to uniquely fingerprint devices [20, 18, 24, 25, 16]. This field contains information about device capabilities or SSIDs, with specific combinations of IE fields selected to maximize signature effectiveness [20].

The second metric examines the temporal characteristics of probe-requests, such as inter-arrival times, to distinguish between devices with randomized MAC addresses. Distinct patterns in temporal behavior arise from manufacturer disparities [30, 31, 17, 32, 33].

The third metric uses the consistency of sequence numbers to differentiate between randomized MAC addresses that change within a specific timeframe. This relies on the range of sequence numbers broadcasted by different devices [20, 26, 27, 16].

The final metric involves utilizing RSSI vectors captured by different sniffers. Devices with changing MAC addresses will have similar RSSI vectors, aiding in address association [16, 28]. However, RSSI measurements are volatile in time and position [34] and are not reliable as device-specific signatures.

There is a lack of literature on effectively combining these signature metrics. [16, 28] combine IE, sequence number, and RSSI, but the resulting signatures do not achieve high association accuracy in various scenarios (cf. Section 3).

[29] examines a combination of attributes related to the content and length of optional fields within transmitted frames. They use density-based clustering algorithms, such as DBSCAN, OPTICS, and HDBSCAN, to group frames sent by the same device. However, the study does not explore the effectiveness of different signature metrics.

Evaluations in controlled semi-anechoic and bus scenarios involve around 30 devices in proximity, with mean accuracy in bus scenarios reaching 75%. The limited number of devices and the short duration of dataset collection (approximately 30 minutes) suggest the algorithm primarily handles a small number of simultaneous randomized MAC address changes.

Remaining challenges: It is essential to choose and integrate signature metrics comprehensively. We need to assess the resulting association framework's performance in situations where device populations vary significantly and there is a high frequency of MAC address changes. With this integration, the framework may become more reliable, particularly in environments with a high concentration of devices (cf. Section 3).

2.4. Datasets used in MAC association literature

Besides MAC address association frameworks, it is essential to examine their evaluation methodology and, more precisely, the datasets used in these studies. Most of the works rely on an evaluation using datasets that are gathered in controlled environments, such as laboratories [17, 20, 18, 24, 25, 26]. In addition, most of these datasets are not public. Unfortunately, utilizing self-generated datasets, particularly those limited in scale, not only raises concerns about replicability but also introduces a level of lack of reliability when applied to a new input dataset featuring a substantial number of devices. The alternative is to use public datasets obtained from large-scale collection campaigns. We can rely on *Sapienza* [35] and *HongKong* datasets [16].

The *Sapienza* datasets, gathered in 2013, contain five different contextual scenarios: *university*, *mall*, *train station*, *vatican1*, and *politics1*. Each dataset was compiled by 5 researchers using their laptops in various environments, including outdoors, indoors, and mixed settings (see Table 2). These datasets are non-randomized, requiring a transformation for evaluating association solutions. For example, in [17], for each set of m consecutive bursts, a random

MAC address can be generated and substituted to the true MAC address in the dataset. The recorded correspondence between the true MAC address and the random MAC establishes the *ground truth*. We also introduce a measure, γ, which denotes the percentage of devices (true MACs) in the dataset for which we apply the above randomization procedure. We consider two cases: a half-randomized trace ($\gamma = 50\%$) and a fully randomized trace ($\gamma = 100\%$).

Name	Nature	# Probe-requests	Duration	# Sniffers
University	Outdoor	1M	6 Weeks	5
Mall	Indoor	331560	6 hours	5
Trainstation	Mixed	190941	6 hours	5
Vatican1	Outdoor	589278	6 hours	5
Politics1	Indoor	564900	6 hours	5
HongKong	Indoor	5M	1 day	21

Table 2: Used datasets

On the other hand, the *HongKong* dataset, used in [16], contains a day of probe-request data collection. This was conducted on an entire floor of a large shopping mall in Hong Kong in 2021. It represents a dense scenario featuring a substantial volume of probe-requests containing true and randomized MAC addresses. However, it lacks a *ground-truth*. The data was collected using multiple WiFi sniffers implemented on commercial WiFi Access Points, which captured probe-requests. These sniffers operated on channel 1 of the 2.4 GHz band.

`Bleach` uses all the datasets in Table 2 for the framework evaluation, enabling the accuracy to be evaluated in varied contextual scenarios, with and without *ground-truth*.

3. Pitfalls in MAC association literature

We illustrate the limitations of existing frameworks by assessing two case studies from the literature, denoted as follows: 1) `Infocom21` [16] and ii) `WiSec16` [17]. These case studies were selected as they cover all major association methodologies, including the use of sequence numbers (SEQ), information elements (IE), timing information from the received frames, and signal strength (RSSI).

3.1. Assessment methodology

In the `Infocom21` study, the authors utilize the *HongKong* dataset. To assess the effectiveness of their approach, they include frames transmitted by a small

number of older devices that advertise the true MAC address. This approach allows them to address the absence of *ground-truth* for randomized MAC addresses in relation to the originating device. Infocom21 uses IE, sequence number, and RSSI as metrics for their signature.

In WiSec16, the authors use a probe-request dataset collected in a controlled indoor laboratory environment and focus exclusively on frames advertising the true MAC addresses. After that, they introduce randomization by assigning a new random MAC address to the true MAC after every four bursts of probe-requests per device in the trace. WiSec16 uses temporal information from probe-request bursts to infer signatures that are eventually utilized for association.

We assessed their performances using the same dataset traces from the public *Sapienza* datasets for a more fair comparison between the two approaches. As in WiSec16, we randomize the traces while retaining the *ground-truth* information. We selected three trace scenarios: *Vatican 1*, *Trainstation*, and *Mall*. The chosen datasets reproduce indoor and outdoor scenarios. *Mall* dataset is equivalent to the shopping mall dataset condition utilized in Infocom21. All chosen scenarios denote public spaces with large populations with mobility conditions varying from static low to highly mobile devices. *vatican 1* dataset captures an event in a public square while the other two scenarios are places of visit with considerable human mobility.

We consider the two major signature components used by Infocom21 in their association framework: i) IE and ii) SEQ. The remaining third component, RSSI, requires the overhead of placing a large number of sniffers close to each other in the sniffing zone to obtain an effective vector of observed RSSIs for a particular probe-request. Moreover, the RSSI component contributes as low as 10% in the final association accuracy of Infocom21.

Infocom21 uses *discrimination accuracy* as a metric to evaluate the potential of signatures in associating MAC addresses. It is defined as the ratio of the number of correct associations in the randomly selected probes. The authors consider the accuracy metric for 1000 frames for the association. For each of the selected frames, they only consider the frames received during the period between its reception and τ seconds before that. We choose this period τ as 600s (maximum utilized period in Infocom21) to test the limits of the framework in scenarios with high conflict.

In the WiSec16 approach, timing information, precisely the inter-arrival time between received frames, is used to link randomized probe-requests. The authors assess the accuracy of their association framework using a limited lab-

oratory dataset where they artificially introduce "ground-truth" associations.

These controlled indoor data collections involve a relatively small number of devices (only 100) implementing MAC randomization. Such conditions are unlikely to pose a significant challenge to the effectiveness of the employed signatures, resulting in relatively higher accuracy. Furthermore, devices in a laboratory setting typically exhibit longer sojourn times compared to outdoor environments, which aids in correctly deducing the timing signatures.

In light of these considerations, we comprehensively evaluated `WiSec16` across all selected scenarios to thoroughly examine its robustness and performance.

3.2. Literature shortcomings

Framework	Scenario	Signature	Parameter	Accuracy
`Infocom21`	Mall	IE	$\tau = 600$	42%
`Infocom21`	Mall	SEQ	$\tau = 600$	9%
`WiSec16`	Mall	Timing	$\gamma = 50\%$	22%
`WiSec16`	Mall	Timing	$\gamma = 100\%$	9%
`Infocom21`	Trainstation	IE	$\tau = 600$	59%
`Infocom21`	Trainstation	SEQ	$\tau = 600$	11%
`WiSec16`	Trainstation	Timing	$\gamma = 50\%$	24%%
`WiSec16`	Trainstation	Timing	$\gamma = 100\%$	13%
`Infocom21`	Vatican1	IE	$\tau = 600$	40%
`Infocom21`	Vatican1	SEQ	$\tau = 600$	8%
`WiSec16`	Vatican1	Timing	$\gamma = 50\%$	20%
`WiSec16`	Vatican1	Timing	$\gamma = 100\%$	10%

Table 3: Case studies: `Infocom21` [16] and `WiSec16` [17].

Tab. 3 shows that discrimination accuracy achieved by `Infocom21` varies significantly. Both signature components, IE and SEQ suffer significant drops and instability in the obtained accuracy. *Train-station* dataset has considerable mobility, a medium-crowded scenario, and the possibility of the presence of APs, which decreases the number of probes sent by a device. *Mall* dataset represents an indoor scenario with relatively larger crowds and, consequently, a higher density of observed probes. In contrast, *Vatican1* dataset shows a very crowded outdoor environment comprising the general audience and nearby commuters listening to the pope in St. Peter's Square. This leads to a high number of probe-requests captured by the sniffer unit of time, potentially lowering the association accuracy.

The discrimination accuracy demonstrates notable variations across different scenarios, as evident in Table 3. The discrimination accuracy is relatively high in scenarios with lower population density, such as the *Train-station*. However, in highly crowded outdoor settings like *Vatican1*, the accuracy of signature components drops notably. It's worth noting that the decrease in accuracy is more pronounced for SEQ compared to IE.

A similar pattern is observed in the accuracy results obtained for `WiSec16`, as illustrated in Figure 9. The framework displays sensitivity to changing scenarios and the degree of MAC randomization (γ) considered in the dataset. When examining a probe-request trace with complete randomization ($\gamma = 100\%$), the achieved accuracy plummets to just a few percentage points. In terms of data collection scenarios, once again, we observe a decline in accuracy in the *Vatican1* scenario, while the *Trainstation* scenario exhibits relatively better accuracy.

Henceforth, we identify that the existing studies encounter challenges related to subpar accuracy and vulnerability to diminished performance when applied to new input probe-request datasets. These shortcomings arise because these studies are typically evaluated either within controlled settings or in the absence of reliable *ground truth* data.

To address these limitations, it becomes essential to holistically integrate generic signature metrics, such as temporal information extracted from frames, alongside content-specific signatures like IE and SEQ. What needs to be improved in current research is the introduction of a generic methodology for gauging the effectiveness of signatures derived from the aforementioned metrics.

3.3. Paper positioning

We introduce a new MAC association framework named `Bleach`. Bleach leverages two pivotal concepts associated with signature effectiveness: *consistency* and *discrimination power*.

Consistency refers to the ability of a set of signatures to consistently and reliably associate probe requests, even in the presence of dynamic changes in MAC addresses over time. It helps us gauge the framework's stability and reliability.

Discrimination power assesses the framework's capability to accurately distinguish between different devices based on their probe requests. It measures how effectively the signatures can separate devices, especially in scenarios where multiple devices exhibit similar or identical behaviors.

Using the above concepts, we carefully choose our signatures and exploit them to introduce a novel MAC association algorithm. We formulate MAC association as the *resolution of conflicts* arising from observed MAC address changes over a certain time period, as discussed in Section 6 and detailed in the work by [36]. These conflicts represent situations where multiple MAC addresses could potentially be associated with the same device, and resolving them is essential for accurate device tracking.

As we delve into the specifics in Section 9.3, we examine the distribution of conflict sizes observed in various time periods within a new input probe request trace. This distribution serves as a valuable predictive indicator (benchmark) for the performance of the `Bleach` framework. `Bleach` demonstrates its robustness by performing effectively across a wide spectrum of conflict sizes encountered in the probe-request dataset, further emphasizing its adaptability and reliability in real-world scenarios.

It's noteworthy to point out that the concepts of *consistency* and *discrimination power*, which are further discussed in Section 7, represent the first instances in the literature where one can forecast the efficiency of any set of signatures. This capability enables the selection of the most effective metric from a vast array of potential signature metrics.

4. `Bleach` framework overview

The framework `Bleach` takes probe-request trace with randomized MAC addresses as input and yields a dictionary ($\mathscr{A}$) of randomized addresses (M_j) associated with particular devices (U_n). $\mathscr{A}$ can be represented as:

$$\mathscr{A} = \{U_1(M_i, M_j, ..., M_k), \ldots, U_n(M_a, M_b, ..., M_z)\}$$

It consists of **four** major steps as shown in Figure 2.

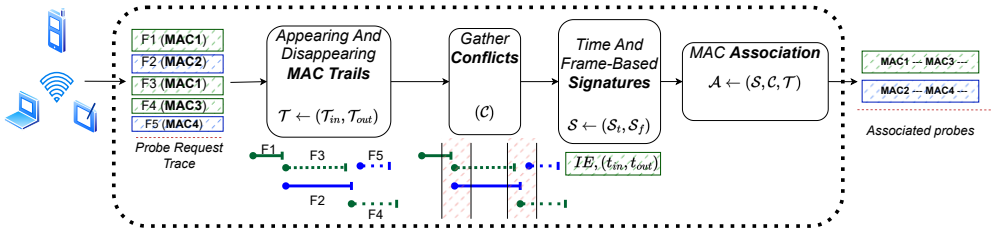


Figure 2: `Bleach` framework.

In the **first step**, we transform the input probe-request trace into a set of MAC address trails. Each MAC trail can be viewed as an instance of the appearance or the disappearance of a MAC address in the sniffing zone. This reduces the problem of MAC association to that of correctly associating each disappearing MAC trail from a device with an appearing trail from the same device. We detail the process of MAC trail generation in Section 5.

In the **second step**, we separate the trails into disjoint subsets comprising *conflicts* ($\mathscr{C}$) (cf. Sec. 6). The conflict denotes the set from which a disappearing MAC trail could be possibly associated with any of the appearing MAC trails present in the dataset within a period ($T_c^{\tau_i}$) from the end of the disappearing trail. We identify this period as the conflict period. The right value of the conflict period allows us to consider all potential associations while deciding to link the MAC address trail pairs.

Conflicts are caused by devices changing their MAC addresses or their entry/exit from the sniffing range. Any address association framework has to resolve conflicts to perform correct assignments between the disappearing and appearing MAC from individual devices. After obtaining conflicts of MAC address changes and a generic formulation of the MAC association problem, we take a step further toward the association itself. We need to obtain effective signatures for resolving conflicting MAC address trails.

In the **third step**, we define and extract the time and frame-based signatures ($\mathscr{S}_t, \mathscr{S}_f$) from the collected MAC trails (cf. Sec. 7). We consider two types of signatures in this paper: i) *time-based signatures*, which utilize the information from the temporal behavior of received probe-request frames, and ii) *frame-based signatures*, which use the control field information present in the captured frame itself to form effective signatures that have the potential of discriminating a device from the rest of the population.

Finally, in the **last step**, we introduce our novel MAC association algorithm capable of accurately resolving the conflicts observed in the input dataset. It uses extracted signatures ($\mathscr{S}$) to fingerprint and differentiate randomized MACs in each conflict duration to finally associate them (cf. Sec. 8).

The following sections detail each of the above-mentioned steps of the `Bleach` framework.

5. Step 1: Extracting MAC trails

We divide the input dataset into MAC address trails, tr^j. A MAC address trail (cf. Sec. 6 and 8) comprises a group of probe-requests sent from a device with a

particular MAC. For each MAC address (M_j) seen in the dataset, we extract two trails from it, as illustrated in Figure 3. One denotes the start of the M_j, which we label as an appearing trail (tr^j_{in}), while the other showcases the end of the advertisement of M_j, which we name as a disappearing MAC trail (tr^j_{out}).

Trails of both natures, though, contain the same bursts (b_n) of probe-requests emitted by the device, with the MAC address as M_j as described in Equation 1. Each burst contains varying number of probe-requests (p_m), i.e. $b_n = \{p_1, p_2, ..., p_m\}$.

$$tr^j_{in}, tr^j_{out} = \{b_1, b_2, ..., b_n\} \tag{1}$$

We consider the appearing MAC trail for association at timestamp $(tr^j_{in})^{start}$, while we consider the disappearing trail for subsequent association at the timestamp $(tr^j_{out})^{stop}$ as shown in Figure 3.

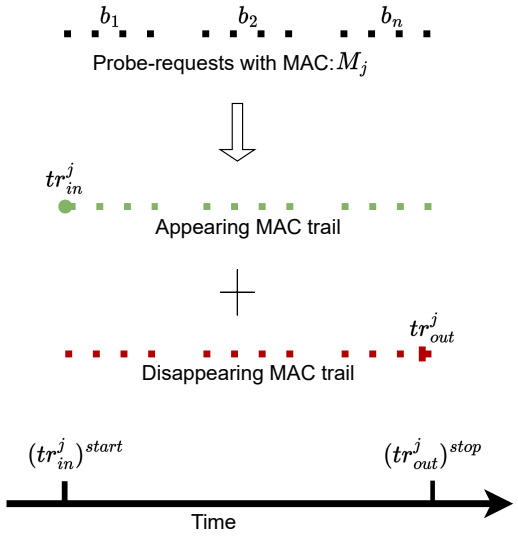


Figure 3: Breaking probe-request sequences (with address M_j) into MAC trails

This distinction in the nature of trails eases the formulation of the address association problem by simplifying it into the correct matching of each disappearing MAC trail (with address M_j) to an appearing MAC trail (with address M_k). Each trail additionally has its characteristic features describing its temporal characteristics (transmission duration, frequency of probes e.t.c.), its nature, and subsequently, the information about composing the signatures from probe-request groups. We denote the set of appearing trails in the dataset as $\mathcal{T}_{in}$ and the set of disappearing trails as $\mathcal{T}_{out}$.

6. Step 2: Obtaining conflicts

After the preliminary step of our framework `Bleach`, we have a set of appearing and disappearing MAC trails from the input dataset. In the second step, `Bleach` identifies MAC association as the resolution of *conflicts*. A preliminary idea of MAC conflicts in BLE is also discussed by [37]. We redefine it comprehensively with respect to WiFi probe-requests. Next, we describe the characteristics of conflicts and the methods to obtain them.

6.1. MAC conflicts

For each disappearing MAC trail (tr^{j}_{out}), we denote a time period ($T_c^{\tau_i}$) starting from the end of tr^{j}_{out}, called as *conflict periods* ($T_c^{\tau_i}$). We illustrate conflict periods in Figure4 where dotted lines in different colors represent different appearing and disappearing MAC address trails of devices in a $T_c^{\tau_i}$.

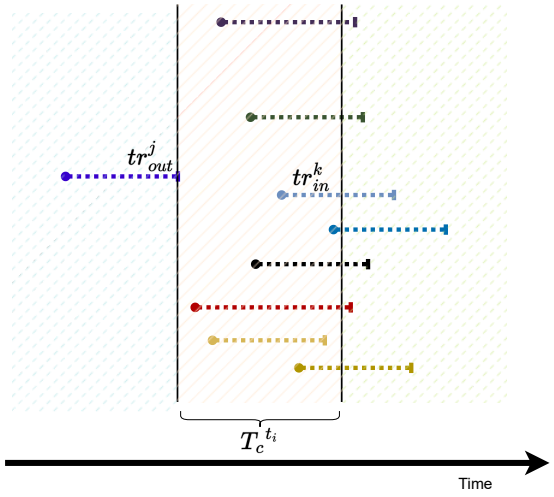


Figure 4: An illustration of conflict periods ($T_c^{\tau_i}$)

Formally, we define a *conflict* (cf. Figure 4) between a disappearing MAC trail, tr^{j}_{out} and an appearing MAC, tr^{k}_{in}, if the two trails satisfy the condition mentioned in Equation 2.

$$\mathscr{C} : tr^{k}_{in}, tr^{j}_{out} \mapsto (T_c^{\tau_i})^{begin} < (tr^{j}_{out})^{stop}, (tr^{k}_{in})^{start} \leq (T_c^{\tau_i})^{end} \tag{2}$$

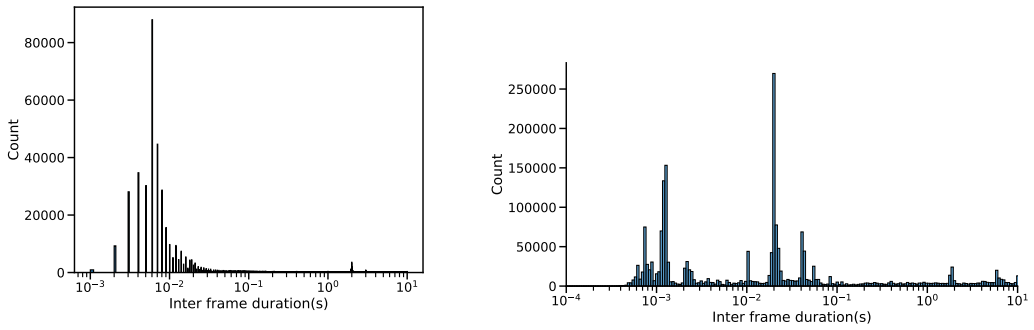
Here $(tr^{k}_{in})^{start}$ and $(tr^{j}_{out})^{stop}$ are the start and stop timestamps of the trails tr^{k}_{in} and tr^{j}_{out}. $(T_c^{\tau_i})^{begin}$ and $(T_c^{\tau_i})^{end}$ are the beginning and end timestamps of a particular conflict period $T_c^{\tau_i}$.

As we already know, each MAC trail (tr^j) consists of probe-request bursts. To isolate individual bursts from respective devices, we investigate burst-related parameters. Isolating and investigating individual and adjacent bursts is critical in choosing the right value of conflict period, $T_c^{\tau_i}$ as MAC addresses from a single device only change on a new burst of probe-requests. A small value of $T_c^{\tau_i}$ will cause `Bleach` to miss a potential correct association of a disappearing MAC trail to the appearing one as the new burst will start after the chosen $T_c^{\tau_i}$. A very large value would mean considering unnecessary associations, as it is highly unrealistic for the duration between two consecutive bursts from a device to be too big. These unnecessary associations lead to higher time complexity of the framework.

6.2. Choosing burst parameters

We define two parameters related to bursts: i) Burst duration (t_b) and ii) Conflict period ($T_c^{\tau_i}$). Understanding the burst duration is crucial as each MAC address trail (tr^j) in Figure 4 consists of sequences of bursts, with each burst containing multiple frames. Isolating bursts aids in the development of signatures as well (cf. Section 7.1.1). Conversely, $T_c^{\tau_i}$ enables the identification of conflicts that `Bleach` must consider when associating a disappearing MAC address that may have been randomized.

We analyze the histogram of inter-frame durations (IFS) observed in captured frames from the *HongKong* and *Sapienza* datasets. IFS represents the time difference between two consecutive probe-requests sent by a specific device, as observed by the sniffer.



(a) (b)

Figure 5: Inter-frame duration(s) (IFS) in Accumulation of all devices in a) HongKong dataset, b) Sapienza datasets

Figure 5a displays two prominent peaks in IFS bin counts: one in the millisecond range and a smaller peak in the second range. Similarly, Figure 5b, encompassing all scenarios in the Sapienza datasets, exhibits distinct peaks in the millisecond range and several smaller peaks in the second range. This phenomenon is expected as devices transmit probe-requests in bursts across different channels to solicit responses from nearby access points.

The IFS within bursts is typically short due to consecutive transmissions, whereas longer IFS values indicate intervals between bursts from the same device, suggesting new probing rounds occurring after a period of time (a few seconds). Therefore, frames with IFS less than 1 second likely belong to a single burst, indicated by the major peaks, while those between 1 to 10 seconds represent inter-burst times (IBT), as denoted by smaller peaks.

Consequently, we define the burst duration (t_b) as 1 second. The conflict period (T_c) is set to 10 seconds, enabling detection of MAC address changes associated with new bursts from devices within a conflict ($\mathscr{C}$).

7. Step 3: Obtaining signatures

Signatures ($\mathscr{S}$) are deductions from exhibited characteristics of a device or entity, which allows isolating it from the rest of the population. We propose and use two signatures extracted from captured probe-requests to associate randomized MAC addresses from a device.

In the following, we first present our choice of signatures for associating randomized WiFi MAC addresses inside `Bleach` framework. Then, we proceed to present details for computing the chosen signatures. Finally, we end the section by justifying the choice of considered signatures.

7.1. Chosen signatures

We choose i) **Time-based** signatures and ii) **Frame-based** signatures for our association framework. Time-based signatures utilize the timing-related information obtained from the frame reception by a sniffer from respective devices. These signatures are effective choices as they are generic and independent of the device type.

We combine the time-based signatures with the frame-based signatures. Frame-based signatures supplement the cases where the timing information from the frames is not representative of a device due to fewer probes or the high variability in timing information per user device. Next, we discuss the choice and effectiveness of these two signatures in detail.

7.1.1. Time-based signature

We already illustrate the behavior of IFS in Figure 5 when analyzing probe-request bursts. The properties of a burst could be extracted that are unique for an observed device in the dataset. We choose the timing information: mean inter-frame time (IFS) across individual probe-request bursts as the time-based signature ($\mathscr{S}_t$) for the device advertising a particular MAC address. Mean IFS is the average interval between subsequent probe-request frames received from a device inside a burst while considering all the bursts from that device. The idea is that the frequency of sending probes during an active scan of networks is likely to differ across devices while remaining unique for the same device. Hence,

$$\mathscr{S}_t = \mu^{IFS}$$

For the calculation of μ^{IFS} in a MAC trail, we take the mean of IFS values inside a burst while considering all observed probe-request bursts in the trail.

7.1.2. Frame-based signature

For frame-based signatures, we investigate the information elements (IE) [Section 9.4.2.1, [21]] contained inside a probe-request frame. This field depicts the abilities of the sending device, which is used for its negotiation with the access point. There are multiple IE fields referred to by their Element IDs, which range from 0 to 255 [21]. We look at around 500,000 frames from the HongKong dataset and investigate specific capabilities advertised by the probe-requests as a part of IE.

The inclusion of Information Elements (IEs) within the probe request is not obligatory, but they are necessary for specifying the supported functionalities of the device. Each device could send all the IEs or only a subset of them, depending upon the context where the WiFi device is situated, the manufacturer, etc. We take the maximum occurring elements of IE in the probes we investigate.

The top 8 most probable metrics that are likely to be consistent in terms of presence are shown in Tab. 4. Hence, we select frame-based signature as:

$$\mathscr{S}_f = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$$

We investigate the potential of this signature in being discriminative in Section 7.3.

Name	IE element	Percent occurrence
e_1	SSID	100
e_2	Supported Rates & BSS Membership	100
e_3	Extended Supported Rates	99.51
e_4	HT Capabilities	82.60
e_5	Vendor Specific	62.56
e_6	Extended Capabilities	54.52
e_7	Interworking	13.5
e_8	VHT Capabilities	2.43

Table 4: Most frequent IE elements

7.2. Computing MAC trail signatures

After obtaining the formulations for time and frame-based signatures, we proceed to finally give details for computing them for each MAC address trail in the input dataset (cf. Algorithm 1). In Alg. 1, we illustrate the application of the first step of `Bleach` too for completeness.

We first isolate/group probe-requests per MAC (M_j) to collect all the individual probe-requests bursts to advertise that address. Grouping into bursts takes into account the burst duration (t_b) that we calculated earlier (cf. Sec. 6). For each burst group with MAC M_j, we add an instance of appearing and disappearing MAC trails in ($\mathcal{T}_{in}$) and ($\mathcal{T}_{out}$) respectively.

For each MAC trail in tr_{in} and tr_{out}, we randomly select a representative frame for that trail (f_{in} and f_{out}). We use f_{in} and f_{out} for calculating the frame-based signatures ($\mathscr{S}_f^{f_{in}}$ and $\mathscr{S}_f^{f_{out}}$) of the considered MAC trail.

We finally obtain trail signatures ($\mathscr{S}[tr_{in}]$ and $\mathscr{S}[tr_{out}]$) as a tuple comprising of frame-based signatures and the mean inter-frame space (μ^{IFS}) of the considered appearing (tr_{in}) and disappearing trail (tr_{out}) respectively.

7.3. Evaluating chosen signatures

We have to formulate the *effectiveness* of a signature to ensure that the association is likely to be the correct one. The two factors that we identify as generic indicators for a signature's performance are: i) *Consistency* and ii) *Discriminating power*.

Consistency measures the ability of a signature to be uniform for a single entity across multiple instances of itself in the population. In our case, the population is the set of WiFi devices emitting probe-requests while a single

Algorithm 1 Computing Signatures

procedure COMPUTESIGNATURES($t_b, \mathscr{S}_f$) ▷ input variables
  $\mathscr{B} \leftarrow \phi$ // Dictionary of probe bursts
  $\mathscr{S} \leftarrow \phi$ // Dictionary of signatures
  $\mathscr{T}_{in} \leftarrow \phi$ // Appearing MAC trail
  $\mathscr{T}_{out} \leftarrow \phi$ // Disappearing MAC trail
  for $M_j \leftarrow \Sigma$ **do**
    $\mathscr{P} \leftarrow GroupProbes(\Sigma, M_j, t_b)$
    $\mathscr{T}_{in}, \mathscr{T}_{out} \leftarrow TrailMACs(\mathscr{P})$
    $\mathscr{B}[M_j] \leftarrow \mathscr{P}$
  end for
  for $tr_{in}, tr_{out} \leftarrow \mathscr{T}_{in}, \mathscr{T}_{out}$ **do**
    $f_{in}, f_{out} \leftarrow RandSamples(tr_{in}), RandSamples(tr_{out}))$
    $\mathscr{S}[tr_{in}] \leftarrow (\mathscr{S}_f^{f_{in}}, \mu^{IFS}{}_{tr_{in}})$
    $\mathscr{S}[tr_{out}] \leftarrow (\mathscr{S}_f^{f_{out}}, \mu^{IFS}{}_{tr_{out}})$
  end for
  return $\mathscr{S}$
end procedure

entity is a particular WiFi device. The multiple instances are multiple probe-requests/probe-request bursts with randomized MACs from the same device. The intuition is that the signature should not be volatile for a single device itself in the first place and should ideally be able to associate all MACs from a device. Hence, a high *consistency* value is essential for an effective signature.

Once a signature validates *consistency* per device, the second factor we should complement it with is the *discriminating power*. It implies that the signature values should be variable across devices in the dataet. Ideally, the larger the size of the range from which the device's signature exhibits its values, the higher the chances of it to be correctly associating randomized MAC addresses among those in the population. Multiple devices with similar signature values are likely to lower the accuracy with which a signature correctly associates addresses.

Limitations of Signatures: Time-based signatures face several limitations, including high variability in timing information across different devices and environments, insufficient data in scenarios with fewer probe requests, and susceptibility to external factors like network congestion or interference. Similarly, frame-based signatures have their own limitations: the inclusion of Informa-

tion Elements (IEs) within probe requests is inconsistent and not obligatory, leading to variability in the available data. Additionally, the IEs can vary depending on the manufacturer, device state, and context, affecting the uniqueness and reliability of the signatures. Furthermore, the selected IEs may not always provide sufficient discriminative power to distinguish between devices, especially in dense environments.

In the following, we acknowledge these limitations and show that our careful choice of signatures minimizes the impact and ensures high *consistency* and *discriminating power*.

7.3.1. Time-based signatures

We illustrate the *consistency* of the time-based signatures. We first compute the signatures for each probe-request burst by an individual device with MAC, M_j in the collected trace. We normalize the signature values between 0 and 1. Finally, the *consistency* in time-based signatures, $\mathscr{CS}$ for M_j is defined as:

$$\mathscr{CS}^{M_j} = 1 - \sigma\left(\frac{\mathscr{S}_t}{maximum(\mathscr{S}_t)}\right) \tag{3}$$

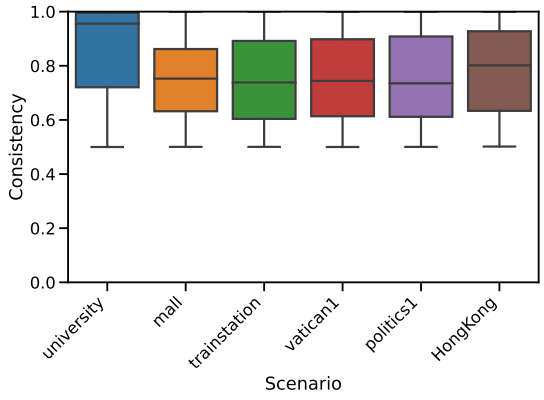


Figure 6: *Consistency* in time-based signatures

We look at the range of consistencies shown by observed MACs in multiple datasets. Figure 6 shows the results for the *consistency* of chosen time-based signatures. We could observe that $\mathscr{S}_t$ demonstrates high *consistency* in each of the scenarios. On average, the *consistency* is greater than 75%, and up to 100% for MAC addresses in the datasets. The stability of $\mathscr{S}_t$ across bursts from the

same datasets is essential to be considered an effective signature. We observe that for all scenarios, we achieve a high consistency, enforcing the stability of $\mathscr{S}_t$.

To finalize the mean IFS as the time-based signature, we also check its *discrimination power*. We looked at the difference between mean IFS for each pair of MAC address pairs observed in various datasets. Figure 7 shows that the difference in mean IFS takes a wide range of values in the interval (0, 0.2) seconds. This ensures the high discrimination power of $\mathscr{S}_t$ as a signature. Finally, the last observation is that the Mean IFS inside a burst is device-specific and similar across various datasets.

The mean IFS has a high *consistency* with respect to a particular device while is variable over a large range of values when considering different devices. This affirms the ability of the signature to discriminate the MAC from a device from the rest of the population.

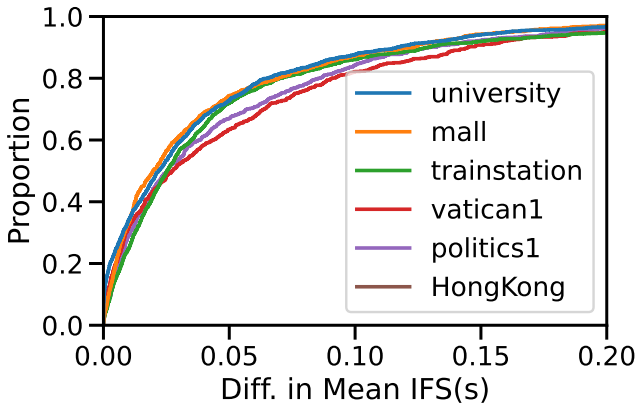


Figure 7: Difference in mean IFS.

7.3.2. Frame-based signatures

To compare multi-dimentional frame-based signatures ($\mathscr{S}_f$), we define a similarity metric ($\mathscr{Z}$) which demonstrates and validates its *Consistency* and the *Discriminating power*. For two MAC addresses emitted from devices A and B and their respective frame-based signatures, $\mathscr{S}_f^A$ and $\mathscr{S}_f^B$ the similarity, $\mathscr{Z}$ is:

$$Z(\mathscr{S}_f^A, \mathscr{S}_f^B) = \sum_{i=1}^{8} isEqual(\mathscr{S}_f^A[i], \mathscr{S}_f^B[i]) \tag{4}$$

The function $isEqual$ checks if the corresponding elements of either signature are equal and are not absent (ϕ). If this is satisfied, it returns 1, else 0. Intuitively, $Z(\mathscr{S}_f^A, \mathscr{S}_f^B)$ indicates the extent of similar elements transmitted by both devices.

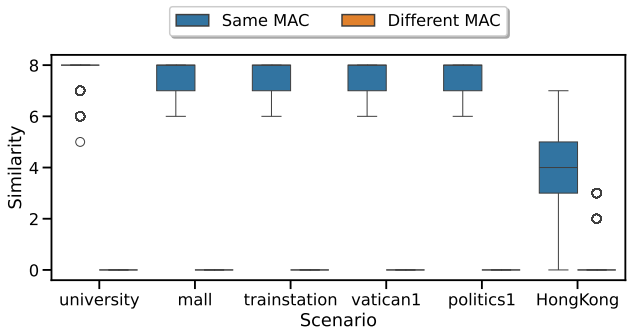


Figure 8: Similarity between frame-based signatures.

We investigate the similarity across a large number probe-requests pairs transmitting the same and different MAC addresses while considering frames transmitting their real MACs in the Sapienza and HongKong datasets. We look at the distribution of $Z(\mathscr{S}_f^A, \mathscr{S}_f^B)$ for both the cases in Figure 8. We observe that the similarity is very high for probes from the same device (MAC), while it is practically zero for different MACs. HongKong dataset has relatively diverse values for the same MACs as Sapienza scenarios due to the absence of certain IE fields in some of the frames. The absence leads to the highest attainable similarity value as lower than 8 for some frames.

There is a considerable gap in similarities between a potential true and false association by the signature, demonstrating the high *Discriminating power* of $\mathscr{S}_f$. The higher the gap, the easier it is for the signature to distinguish between the true and the false associations. Moreover, signatures from the same MAC, or in this case, the device, show a high degree of similarity. This also showcases the high *consistency* of $\mathscr{S}_f$, hence validating its effectiveness.

8. Step 4: MAC Association

We utilize Algorithm 2 to associate randomized probe-request addresses. It takes as input the set of appearing and disappearing trails along with the obtained collection of signatures ($\mathscr{S}$). Algorithm 2 yields a dictionary of MAC

pairs ($\mathscr{A}$), denoting the associated randomized addresses. The association relies on the correctness of predictions for accurate associations when considering pairs of disappearing and appearing MAC trails.

We start with no associated MAC addresses. We sort the appearing and disappearing trails in time so that we can match each disappearing trail with a corresponding appearing trail if it is the same advertising device ($\mathscr{T}_{in}^{'}, \mathscr{T}_{out}^{'}$). We also keep track of associated appearing trails that are already paired in order to avoid comparing them again while resolving another conflict ($associated$).

8.1. Logistic regression predictor

We opted to utilize a logistic regression model for predicting the degree to which a potential MAC trail pair represents a correct association. We choose this model over methods like random forest or CatBoost for various reasons. It offers interpretability through coefficients that directly show the impact of predictors on the outcome, making it ideal for understanding relationships. Logistic regression assumes a linear relationship between predictors and the log-odds of the outcome, which can be advantageous when this holds true, ensuring stable performance. Also, it is computationally efficient for large datasets, contrasting with more complex models. Additionally, logistic regression's focus on binary classification tasks and straightforward feature importance metrics further supports its suitability for probe-request datasets.

To train this model, we combined frame and time signatures as features. The first feature, denoted as $f1$, is computed by determining the similarity between the representative frames of the conflicting MAC address trail pair using Equation 4. We perform this process for each possible pair of disappearing and appearing MAC trails observed in the training dataset ($\mathscr{S}_f^{f_{in}}, \mathscr{S}_f^{f_{out}}$). The second feature, denoted as $f2$, is derived by calculating the absolute difference between the mean Inter-Frame Spacing (IFS) periods observed in the trail pair. The logistic regression predictor is likely to be efficient as we only have a couple of features with the two classes (true and false associations) distinctly different due to the high discriminative power of both features (cf. Section 7.3).

8.2. Resolving randomized MACs

We examine each disappearing trail (tr_{out}) one by one from the set of trails that have been previously sorted in chronological order ($\mathscr{T}_{out}^{'}$). To ensure the significance of the signatures, we filter out trails that are too short by considering only those with at least 4 frames (MIN_TRAIL_LENGTH). Subsequently,

Algorithm 2 MAC address association

procedure ADDRESSASSOCIATION($\mathscr{S}, \mathscr{T}_{in}, \mathscr{T}_{out}$) ▷ input variables
  $\mathscr{A} \leftarrow \phi$
  $\mathscr{T}_{in}{}', \mathscr{T}_{out}{}' \leftarrow TimeSort(\mathscr{T}_{in}), TimeSort(\mathscr{T}_{out})$
  $associated \leftarrow [False] x length(\mathscr{T}_{in}{}')$

  $f1 = Similarity(\mathscr{S}_f^{f_{in}}, \mathscr{S}_f^{f_{out}})$
  $f2 = |\mu^{IFS}{}_{tr_{in}} - \mu^{IFS}{}_{tr_{out}}|$
  $\mathscr{L} \leftarrow LogisticRegression((f1, f2))$

  for $tr_{out} \leftarrow \mathscr{T}_{out}{}'$ **do**
    if $tr_{out}.length > MIN\_TRAIL\_LENGTH$ **then**
      $\mathscr{C} \leftarrow Conflicts(\mathscr{T}_{in}{}', tr_{out}, T_c^{\tau_i})$
      $\mathscr{V} \leftarrow \phi$
      for $ctrail \leftarrow \mathscr{C}$ **do**
        $fvect \leftarrow LogisticFeatures(tr_{out}, ctrail)$
        $\mathscr{V} \leftarrow PredictionProb(\mathscr{L}, fvect)$
      end for
      $\mathscr{V}' \leftarrow Sort(\mathscr{V})$
      for $ctrail \leftarrow \mathscr{C}$ **do**
        $dseq \leftarrow SeqNumGap(tr_{out}, ctrail)$
        if $dseq < SEQ\_TH$ & associated[ctrail] !=True **then**
          $\mathscr{A} \leftarrow (tr_{out}, ctrail)$
          $associated[ctrail] = True$
          $ExitTheLoop()$
        end if
      end for
    end if
  end for
  return $\mathscr{A}$
end procedure

we identify the set of appearing trails that conflict ($\mathscr{C}$) during the duration ($T_c^{\tau_i}$) following the disappearance of the considered MAC trail, tr_{out}.

For each conflict trail, we obtain the corresponding feature vector ($fvect$) to derive the prediction probabilities from the trained logistic regression model,

$\mathscr{L}$. This yields a probability vector of the size of the conflicts ($\mathscr{V}$), indicating the likelihood of the MAC trail pairs being transmitted from the same device. We then sort this vector in descending order of probabilities to select the best feasible match.

Although it is possible that some associations may involve a new device in the sniffing zone rather than a randomized MAC from a previously seen device, we propose a methodology to address this issue. For each conflicting trail ($ctrail$), we calculate the gap in sequence numbers ($dseq$) between this trail and the disappearing trail under consideration. To ensure accuracy, we establish a threshold for this sequence number gap (SEQ_TH) within our association algorithm.

To determine an appropriate threshold, we analyze the sequence number gaps observed across all MAC trails in various datasets, as illustrated in Figure 9. We observe that approximately 85% of trails exhibit a sequence number gap of less than 64. Therefore, we set the value of SEQ_TH to 64.

For sequence number gaps larger than 64 up to 4095, Figure 9 shows a gradual and uniform increase, possibly indicating a device re-entering the sniffing zone after missing multiple consecutive bursts. Additionally, new devices often appear in the dataset, with their initial frame having a sequence number randomly distributed within the range [0, 4095]. Hence, in `Bleach`, we disregard sequence number gaps from 64 onwards.

Finally, we can proceed with the final step of the MAC association algorithm, which involves linking a newly detected randomized MAC trail to a previously seen one. If we encounter a conflicting appearing trail (considered based on their prediction probabilities) that meets the sequence number threshold and hasn't been associated before, we label this MAC as associated and exit the loop to continue with the next disappearing MAC. If none of the conflicting MACs that appear meet the SEQ_TH criterion, we assume that it is a disappearing MAC, representing the last trail observed by that device in the sniffing zone.

9. Evaluation

In this section, we present the evaluation methodology utilized for `Bleach` framework before presenting the evaluation of its effectiveness in associating randomized WiFi MAC addresses.

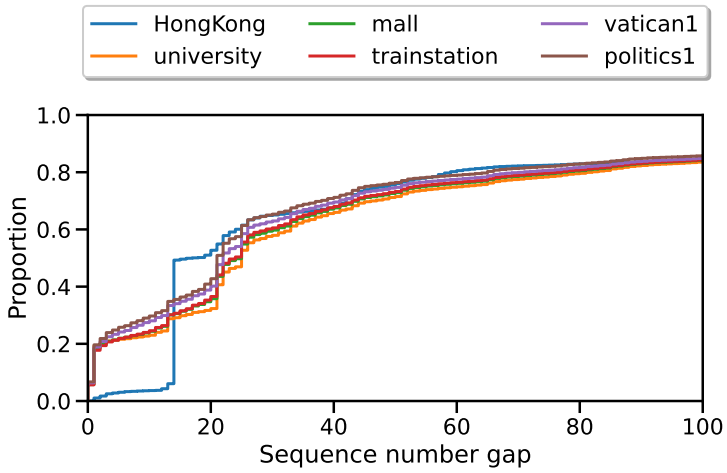


Figure 9: Sequence number gap between MAC trails.

9.1. Evaluation methodology

We first investigate the efficiency of chosen signatures that were used as features to train the logistic regression classifier. Then, we proceed to assess the MAC association capabilities of `Bleach`.

To evaluate the `Bleach`'s association performance, we use a variety of datasets. The first dataset is where we have a *ground-truth* of MACs from the same device. These datasets are part of the Sapienza collection and comprise scenarios like *university*, *mall*, *trainstation*, *vatican1*, and *politics1*. After validating the framework `Bleach` with ground-truth datasets, we utilize the HongKong dataset, which consists of capturing randomized MACs of devices in a shopping Mall using a large number of sniffers. This dense dataset contains both devices that transmit their true (non-randomized) MAC addresses and randomized MACs. We associate the randomized MAC of HongKong dataset, thus generalizing the performance of `Bleach` to the cases with no *ground-truth* of random MAC addresses from the same sender.

9.2. Performance of signatures

Since the base of our association framework is the logistic regression classifier trained with features comprising of the time and the frame-based signature,

the first step of the evaluation process is to evaluate the performances of such signatures.

Training/test datasets: We train the logistic regression model over the two features, using the Sapienza datasets due to the access of *gound-truth*. For obtaining the *ground-truth*, we manually randomize the Sapienza datasets by grouping the MAC addresses per device into a sequence of bursts using the burst duration (t_b).

We assign new unique identifiers to a device after every 4 bursts. We opt for the same number of bursts per MAC address as in literature [17] to keep a *ground-truth* of appearing and disappearing trails in the dataset. We isolate positive (true association) and negative (false association) MAC pairs to eventually train the logistic regression model ($\mathscr{L}$).

We train the model on *university* and *mall* scenarios and observe the accuracy of the classifier on test sets comprising of the remaining three datasets: *trainstation*, *vatican1*, and *politics1*. IE fields and the mean IFS in the frame and time-based signatures are device-specific and, hence, are not heavily dependent on the choice of training scenarios. We chose 50k random MAC trails from each dataset for the test. The accuracy depicts the model's effectiveness in correctly separating the true and false associations among the respective disappearing and appearing MAC trails.

Evaluation metrics: We use three metrics to look at the performance on the test set: i) Precision, ii) Recall, and iii) F1-score. Precision is the ratio between the True Positives and all the positives, while Recall shows the proportion of actual positives that were identified correctly. F1-score is the Harmonic mean of the Precision and Recall.

Case	Precision	Recall	F1-score	Dataset
False association (negative)	0.79	0.65	0.71	trainstation
	0.99	0.67	0.80	vatican1
	0.91	0.61	0.73	politics1
True association (positive)	0.70	0.82	0.76	trainstation
	0.75	0.99	0.86	vatican1
	0.70	0.94	0.81	politics1

Table 5: Performance of signatures

Evaluation results: We observe in Tab. 5 that we achieve an F1-score up to 86% with a minimum of 71%. This certifies relatively high Precision and Recall

achieved by our signature-based logistic regression classifier, both in false and true associations. It shows that the model produces fewer false positives and negatives, demonstrating its effectiveness. The accuracy of association across a dataset could vary depending on the number of MAC addresses in a conflict that Alg. 2 has to resolve. We next discuss this in detail.

9.3. Datasets with ground-truth

In Tab. 6, we illustrate the accuracy of association obtained in different contextual scenarios of Sapienza datasets. We define accuracy as the percent of correct association of the disappearing trail with respect to the total number of disappearing trails that `Bleach` considered for the address resolution. Considering different scenarios helps the framework to be robust against i) a variety of mobile devices with specific temporal behavior of probe-request bursts, ii) high densities of mobile devices around the sniffer, and iii) diverse address randomization strategies by the manufacturer.

Scenario	Accuracy	Scenario	Accuracy
university	99.14	trainstation	94.82%
mall	60.89	vatican1	74.80%
politics1	69.14		

Table 6: Accuracy in Sapienza datasets.

We observe that the accuracy of association is variable across datasets, demonstrating the heterogeneity that we expect each of them to possess. In *university* scenario, we resolve close to 99% of randomized address trails, while the train station to exhibits a high accuracy of close to 95 %. Even the highly dense outdoor setting of Vatican city square (*vatican1*) achieves a modest accuracy of around 75%. Finally, the major indoor scenario of *mall* and political meeting hall obtain relatively low accuracy of around 61 and 69%, respectively. We next explore and reason the performance variability for `Bleach` and, in general, any association framework in detail.

We now compare the above results with the state-of-the-art accuracies on the Sapienza datasets (shown on Table 3 in Case studies). In the *Mall* scenario, the highest accuracy from `Bleach` (60.89%) is 18.89% higher than the highest accuracy from the case studies (42%). For the *Trainstation* scenario, the highest accuracy from our solution (94.82%) is 35.82% higher than the highest accuracy from the case studies (59%). In the *Vatican1* scenario, the highest accuracy

from *Bleach* (74.80%) is 34.80% higher than the highest accuracy from the case studies (40%).

Interpreting association accuracy: In the following, we aim to understand the heterogeneity of datasets as an input to MAC association frameworks, which cause the fluctuation in performance in time and across scenarios. As we propose and illustrate in Sec. 6.1, MAC association can be abstracted into the resolution of address conflicts $\mathscr{C}$. $\mathscr{C}$ showcase all possible appearing MAC trails that could be associated with disappearing ones during various conflict periods $T_c^{\tau_i}$ of the input dataset. The size of conflicts, $|\mathscr{C}(T_c^{\tau_i})|$ in the dataset captures the *complexity* that an association framework has to face.

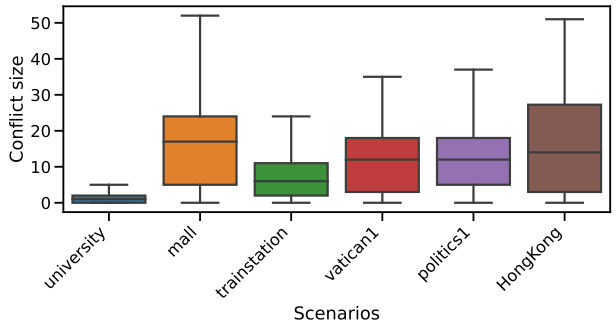


Figure 10: Conflict sizes of datasets.

$|\mathscr{C}(T_c^{\tau_i})|$ acts as a generic metric that captures various phenomena that could potentially affect the performance of address association like: i) Inter-arrival times of probe-requests, ii) Mobility patterns of users across the capturing sniffers, iii) Heterogeneity of hardware (mobile devices), and, iv) State of devices transmitting probe-requests (like idle screen, WiFi switched off, power-saving mode on, number of known access points) [38].

While lower inter-arrival times of frames at sniffer are likely to inflate the conflict size, short-term stay of the mobile device or repeated entry-exit in the sniffing zone will make the $|\mathscr{C}(T_c^{\tau_i})|$ high and volatile. This induces errors in the association as resolution means successfully isolating correct MAC trails among a large number of possible pairs. Similarly, various datasets could have differences in the kinds of mobile devices and their state during the probe-request collection. These factors affect the frequency and pattern of transmitted probes, leading to variable conflict sizes faced by the resolution framework. Instead of looking at individual phenomena, conflict sizes act as a common

metric to compare and *benchmark* the performance of our framework.

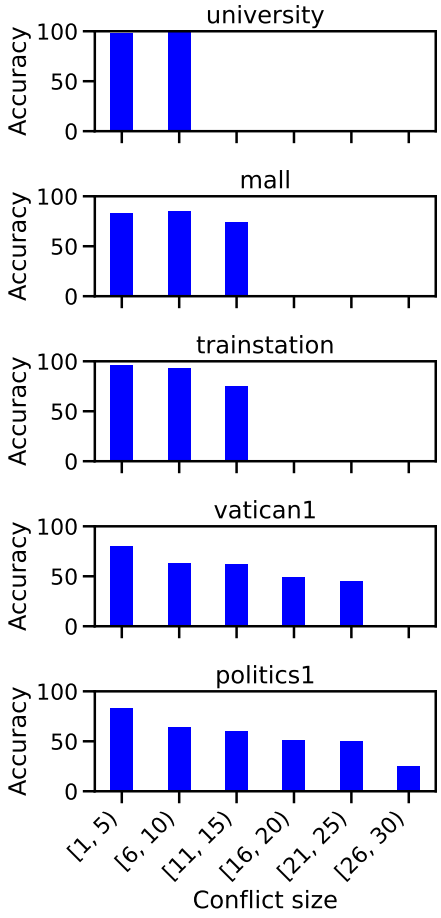


Figure 11: Association accuracy in different conflict sizes bins.

Consequently, we look at the performance of `Bleach` with respect to $|\mathscr{C}(T_c^{\tau_i})|$ seen in various input datasets. In Figure 10, we observe the distribution of conflict sizes resolved in various scenarios. *University* and *trainstation* have relatively lower value of $|\mathscr{C}(T_c^{\tau_i})|$, which should transform in to better accuracy of association. Indeed, Tab. 6 validates the claim as we achieve overall accuracy of 99.14% and 94.82%, respectively. *Vatican1* and *politics1* have mid-range conflict sizes resulting in slightly lower but good accuracy. In contrast, highly dense shopping mall scenarios like *mall*, *HK dataset 1*, and *HK dataset 2* face

considerably high conflict sizes for address resolution, resulting in lower accuracy among the input datasets.

Next, we investigate the variability in association accuracy inside a single scenario across time. The hypothesis is that even with higher overall conflict sizes, there might be periods with low $|\mathscr{C}(T_c^{\tau_i})|$, which could be exploited by the adversary to resolve randomized addresses of target user devices. We indeed observe in Figure 11 that all scenarios generally have periods with low conflict sizes that yield better accuracy. While scenarios like *university* and *trainstation* perform reasonably well in all low $|\mathscr{C}(T_c^{\tau_i})|$, *vatican1* and *politics1* see a wide range of high conflict sizes causing a depletion in achieved correct MAC associations.

This characterization acts as a benchmark for `Bleach` in any new input datasets to the framework with similar or higher expected values of $|\mathscr{C}(T_c^{\tau_i})|$. It ensures the reliability of our framework, unlike other existing frameworks in the literature, which perform variably in different contextual scenarios in proprietary datasets.

9.4. Datasets without ground-truth

For datasets with no *ground-truth* (here HongKong dataset), we propose an alternate metric that denotes the correct association of the MAC addresses. The proposed metric is the sojourn time of a particular device around the sniffer zone. In the case of MAC randomization, the sojourn time is the sum of the sojourn times of all associated randomized MACs plus the time gaps between the associated MAC trails. More specifically, the device's sojourn time is the difference between the timestamps of the first frame of the first associated MAC trail and the last frame of the last associated MAC trail.

In the case of randomized MACs, which are not associated, the sojourn times correspond to the lifetimes of each random MAC address that the device advertises. While in the case of true or non-randomized MAC addresses, the sojourn time is the time for which the device was seen in the sniffing zone.

Hypothesis: We propose the hypothesis that for a large number of users observed by the sniffers, the distribution of the sojourn times of correctly associated MAC addresses and the true MACs advertised by users should demonstrate similar behavior in a scenario during a given period of time. To recall, the true MACs are the physical MAC addresses of devices that remain static across all sent probe-requests. The consistent nature of human mobility during that

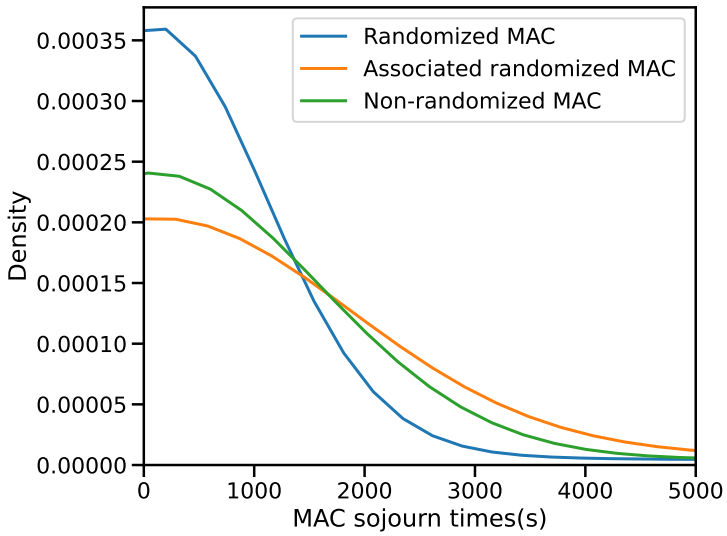


Figure 12: MAC sojourn times before and after association.

short period, and the uniform randomization nature of the device's MAC address for the large population, ensures that the sojourn times of devices are independent of MAC randomization.

Observations: In Figure 12, we present the probability densities of sojourn times observed when considering probe-requests from HongKong dataset that advertise non-randomized, randomized, and associated MAC addresses. Here, we consider around 9000 randomized and non-randomized MACs. We observe that randomized MAC addresses have quite lower sojourn times than the other two, as expected. Devices change MAC addresses frequently, lowering the time for which one of its random MAC was seen in the sniffing zone. Next, to validate the effectiveness of MAC association in `Bleach`, we look at the closeness between the sojourn time of devices of non-randomized MAC addresses of a device and the associated randomized ones. We notice that the sojourn times of devices after association and those of non-randomized ones are indeed very similar in their distributions. Perfect overlap is not possible because of the limits of the association algorithms in highly dense (in terms of probe-requests) and mobile scenarios like shopping malls (cf. Figure 10).

10. Conclusion

MAC address randomization is used by modern WiFi devices, where randomly generated virtual MAC addresses are used in probe-requests, instead of true MAC addresses. Though privacy-protecting, MAC randomization hinders the continuation of works such as people counting, human mobility inference, and crowd flow estimation. We find out that current address association frameworks underperform and are unreliable with respect to new input datasets. We henceforth present `Bleach`, a framework capable of associating randomized probe-requests advertised in the observation zone. We implement `Bleach` and used extensive datasets in different contextual scenarios which shows that `Bleach` is robust and greatly outperforms the state-of-the-art works in terms of accuracy.

References

[1] A. K. Mishra, A. Carneiro Viana, N. Achir, C. Palamidessi, Public wireless packets anonymously hurt you, in: 2021 IEEE 46th Conference on Local Computer Networks (LCN), 2021, pp. 649–652.

[2] M. Čavojskỳ, M. Uhlar, M. Ivanis, M. Molnar, M. Drozda, User trajectory extraction based on wifi scanning, in: 2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), IEEE, 2018, pp. 115–120.

[3] M. Kotaru, S. Katti, Position tracking for virtual reality using commodity wifi, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 68–78.

[4] A. Trivedi, C. Zakaria, R. Balan, A. Becker, G. Corey, P. Shenoy, Wifitrace: Network-based contact tracing for infectious diseases using passive wifi sensing, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 5 (1) (2021) 1–26.

[5] J. Weppner, B. Bischke, P. Lukowicz, Monitoring crowd condition in public spaces by tracking mobile consumer devices with wifi interface, in: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, 2016, pp. 1363–1371.

[6] A. Di Luzio, A. Mei, J. Stefa, Mind your probes: De-anonymization of large crowds through smartphone wifi probe requests, in: IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE, 2016, pp. 1–9.

[7] L. Aalto, N. Göthlin, J. Korhonen, T. Ojala, Bluetooth and wap push based location-aware mobile advertising system, in: Proceedings of the 2nd international conference on Mobile systems, applications, and services, 2004, pp. 49–58.

[8] R. Mitchell, I. R. Chen, A survey of intrusion detection techniques for cyber-physical systems, ACM Computing Surveys (CSUR) 46 (4) (2014) 55:1–55:29. doi:10.1145/2542049.

[9] P. B. J. L. H. Liu, H. Darabi, Survey of wireless indoor positioning techniques and systems, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 37 (6) (2007) 1067–1080. doi:10.1109/TSMCC.2007.905750.

[10] G. F. R. B. E. Martin, O. Vinyals, Precise indoor localization using smart phones, Proceedings of the International Conference on Multimedia (2010) 787–790doi:10.1145/1873951.1874060.

[11] S. M. M. P. J. Gubbi, R. Buyya, Internet of things (iot): A vision, architectural elements, and future directions, Future Generation Computer Systems 29 (7) (2013) 1645–1660. doi:10.1016/j.future.2013.01.010.

[12] S. J. R. Want, B. N. Schilit, Enabling the internet of things, Computer 48 (1) (2015) 28–35. doi:10.1109/MC.2015.12.

[13] R. B. A. D. L. Beal, W. He, Smart wireless lans with passive client positioning: A method and experimental validation, IEEE Transactions on Mobile Computing 17 (3) (2018) 580–593. doi:10.1109/TMC.2017.2737060.

[14] M. S. M. Conti, R. Poovendran, Fakeprobe: A privacy preserving scheme for wi-fi fingerprint-based localization, Proceedings of the 8th ACM Conference on Security Privacy in Wireless and Mobile Networks (2015) 1–11doi:10.1145/2766498.2766501.

[15] M. P. E. Bardram, R. Kjær, Context-aware user authentication: Supporting proximity-based login in pervasive computing, Proceedings of the 6th

International Conference on Pervasive Computing (2008) 233–250doi:10.1007/978-3-540-79576-6_14.

[16] J. Tan, S.-H. Gary Chan, Efficient association of wi-fi probe requests under mac address randomization, in: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications, 2021, pp. 1–10.

[17] C. Matte, M. Cunche, F. Rousseau, M. Vanhoef, Defeating mac address randomization through timing attacks, WiSec '16, Association for Computing Machinery, New York, NY, USA, 2016.

[18] P. Robyns, B. Bonné, P. Quax, W. Lamotte, Noncooperative 802.11 mac layer fingerprinting and tracking of mobile devices, Security and Communication Networks 2017 (2017).

[19] F. Guo, T.-c. Chiueh, Sequence number-based mac address spoof detection, in: International Workshop on Recent Advances in Intrusion Detection, Springer, 2005, pp. 309–329.

[20] M. Vanhoef, C. Matte, M. Cunche, L. S. Cardoso, 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, 2016, pp. 413–424.

[21] 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) 1–7524.

[22] J. Martin, E. Rye, R. Beverly, Decomposition of mac address structure for granular device inference, in: Proceedings of the 32nd Annual Conference on Computer Security Applications, 2016, pp. 78–88.

[23] J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, D. Brown, A study of mac address randomization in mobile devices and when it fails, Proceedings on Privacy Enhancing Technologies 2017 (4) (2017) 365–383.

[24] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, D. Wetherall, 802.11 user fingerprinting, in: Proceedings of the 13th annual ACM international conference on Mobile computing and networking, 2007, pp. 99–110.

[25] M. Cunche, M.-A. Kaafar, R. Boreli, Linking wireless devices using information contained in wi-fi probe requests, Pervasive and Mobile Computing 11 (2014) 56–69.

[26] F. Guo, T.-c. Chiueh, Sequence number-based mac address spoof detection, in: Recent Advances in Intrusion Detection: 8th International Symposium, RAID 2005, Seattle, WA, USA, September 7-9, 2005. Revised Papers 8, Springer, 2006, pp. 309–329.

[27] G. Chandrasekaran, J.-A. Francisco, V. Ganapathy, M. Gruteser, W. Trappe, Detecting identity spoofs in ieee 802.11 e wireless networks, in: GLOBECOM 2009-2009 IEEE Global Telecommunications Conference, IEEE, 2009, pp. 1–6.

[28] T. He, J. Tan, S.-H. G. Chan, Self-supervised association of wi-fi probe requests under mac address randomization, IEEE Transactions on Mobile Computing (2022) 1–14doi:10.1109/TMC.2022.3205924.

[29] M. Uras, E. Ferrara, R. Cossu, A. Liotta, L. Atzori, Mac address de-randomization for wifi device counting: Combining temporal-and content-based fingerprints, Computer Networks 218 (2022) 109393.

[30] T. Kohno, A. Broido, K. C. Claffy, Remote physical device fingerprinting, IEEE Transactions on Dependable and Secure Computing 2 (2) (2005) 93–108.

[31] C. Arackaparambil, S. Bratus, A. Shubina, D. Kotz, On the reliability of wireless fingerprinting using clock skews, in: Proceedings of the third ACM conference on Wireless network security, 2010, pp. 169–174.

[32] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. V. Randwyk, D. Sicker, Passive data link layer 802.11 wireless device driver fingerprinting., in: USENIX Security Symposium, Vol. 3, 2006, pp. 16–89.

[33] B. Bezawada, M. Bachani, J. Peterson, H. Shirazi, I. Ray, I. Ray, Behavioral fingerprinting of iot devices, in: Proceedings of the 2018 workshop on attacks and solutions in hardware security, 2018, pp. 41–50.

[34] M. Maduraga, R. Abeysekara, Comparison of supervised learning-based indoor localization techniques for smart building applications, in: 2021 International Research Conference on Smart Computing and Systems Engineering (SCSE), Vol. 4, IEEE, 2021, pp. 145–148.

[35] M. V. Barbera, A. Epasto, A. Mei, S. Kosta, V. C. Perta, J. Stefa, CRAWDAD dataset sapienza/probe-requests (v. 2013-09-10), downloaded from https://crawdad.org/sapienza/probe-requests/20130910 (Sep. 2013). doi:10.15783/C76C7Z.

[36] A. K. Mishra, A. C. Viana, N. Achir, Introducing benchmarks for evaluating user-privacy vulnerability in wifi, in: 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring), 2023, pp. 1–7. doi:10.1109/VTC2023-Spring57618.2023.10199706.

[37] L. Jouans, A. C. Viana, N. Achir, A. Fladenmuller, Associating the randomized bluetooth mac addresses of a device, in: IEEE Annual Consumer Communications & Networking Conference, CCNC 2021, Las Vegas, NV, USA, January 9-12, 2021, 2021, pp. 1–6.

[38] J. Freudiger, How talkative is your mobile device? an experimental study of wi-fi probe requests, in: Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks, 2015, pp. 1–6.