A Study on Device Identification and Localization using Wi-Fi

Jamil Haidar-Ahmad, Group 10
Department of Electrical Engineering and Computer Science, University of California, Irvine, USA
Email: jhaidara@uci.edu

***Abstract*—Almost all electrical devices nowadays contain the ability to communicate with the internet through Wi-Fi. While Wi-Fi, namely the IEEE 802.11 wireless standard, has evolved throughout the years with new emerging technologies and capabilities for faster throughput, some security and privacy issues are still present. This paper conducts two studies on privacy issues related to Wi-Fi. The first study attempts to localize a user by installing an application that can listen to scanned Wi-Fi networks. Matching the scanned Wi-Fi lists with the Wigle database gave a very accurate trajectory on the whereabouts of the user. The second study sniffs probe request packets being sent and utilizes different attributes extracted from the probe request to identify a user's device even when its media access control (MAC) address gets randomized. The user's device was correctly identified, and the approach was able to track the device even when it changed its mac address multiple times.**

I. Introduction

In today's connected world, every device relies on the Internet to serve its purpose. This form of communication could be through wired connection such as ethernet, or wireless local-area networks (WLAN). The most common type of WLAN is the wireless standard IEEE 802.11 (Wi-Fi), which is available on almost all devices that have internet connectivity. With the rapid development and popularity of Wi-Fi devices including smartphones and Internet of Things (IoT) devices, security issues begin to rise. By having access to the surrounding Wi-Fi network service set identifiers (SSIDs), a user can be located using a matched map which clusters access point (AP) SSIDs into a known geographic location. Furthermore, by nature, wireless connectivity has an open nature where any listener on certain frequencies could attain important information. This would not be an issue, however, if when scanning for Wi-Fi networks, a mobile phone simply listens to packets sent as beacons from access points since the phone would be ideally not sending any signals and therefore invisible. Unfortunately, discovering access points by scanning all possible channels & listening to beacons (passive scanning) is not considered to be very efficient and is highly discouraged. To enhance this discovery process, devices often use what is called active scanning through sending probe request frames.

One significant part of the IEEE 802.11 standard is the probe request/response management framework. This framework facilitates discoverability of access points and establishing and maintaining communications. A device sends probe requests asking access points for information regarding network properties. Access points then reply to probe requests with a probe response management frame which includes information such as supported data rates. Next, authentication messages are exchanged to establish each device's identity. Once authentication is completed, the device requesting internet connectivity registers with the access point to gain full access to the network.

There is an inherent issue in this system. In the probe request phase, the device's Media Access Control (MAC) address is sent to establish and identity with the access point. However, since this request does not contain any user data, it is not encrypted and is sent as clear text. Furthermore, any person with the appropriate commercial network card supporting monitor mode could capture the signal too. This means that the user could be uniquely identified using probe requests. Furthermore, even when a device is connected to a network, the device still sends probe requests, so the user is constantly transmitting their MAC address and could be easily identified throughout many locations leading to breach of location privacy.

Companies have attempted to mend this issue by introducing MAC address randomization. This means an iPhone would randomize its MAC address when sending probe requests and only keeps it constant when it connects to a network. Furthermore, devices connected to networks send less probe requests than devices searching for networks. Nevertheless, the IEEE 802.11 standard probe request frame structure does include other information which could be sent in a standard probe request. This information could include a Service Set Identifier (SSID) list, vendor information, incremental sequence numbers, supported data rates, bandwidth, Very High Through (VHT) capabilities, and antenna information. A combination of this information can be used for device fingerprinting which could uniquely identify devices even when the MAC address is randomized.

This paper attempts to study the feasibility of localization by Wi-Fi association as well as the identifiability of devices from probe requests using random MAC addresses. The paper makes the following contributions:

- Study the accessibility to Wi-Fi scanning on Android phones and attempt to localize a user by utilizing scanned Wi-Fi network.
- Study the identifiability and uniqueness of probe requests sent by a device even when the MAC address is randomized.

The rest of the paper is organized as follows: section II will present background and related work, respectively; section III describes the proposed method for each of the

contributions; section IV discusses experimental results; and section V presents conclusions and future work.

II. Literature Review

A. AP Discovery & Connection

In the modern era of Internet of Things and mobile networks, AP discovery forms the basis of every device's first step when connecting to a Wi-Fi network, and as such it has been studied in privacy and security research. When attempting access point discovery, the Wi-Fi standard provides devices with 2 operational modes: passive service discovery and active search discovery.

The first mode, passive service discovery, is the more secure mode of the two. In this mode, the device listens passively every certain period of time and scans multiple channels to find an access point to connect to. If the AP's SSID is known to the device, the device will attempt to initiate connection to the network. This method is highly discouraged due to its inefficiency and setbacks. First, the device must listen while waiting for beacon signals from APs, which means it could not discover APs that are idle during the device's scanning period. Second, this process is energy exhaustive, which means continuously scanning could drain the battery quickly. Since scanning's power cost is high, the device would attempt to reduce scanning time which could further worsen the issue of the possibility of missing an AP beacon [1].

The second and most commonly used mode is active service discovery. The device actively searches for access points through sending probe requests. Probe requests are low-level signals standardized by the IEEE 802.11 specification for service discovery and are implemented in any Wi-Fi capable device irrespective of the manufacturer or the model [2]. The active scan occurs when a device sends probe requests that contain some information about the device and its communication capabilities. The probe requests could contain a preferred network list (PNL) where for every saved network in the device's memory, a probe request is sent asking for that network's SSID. If an AP's SSID matches an SSID attached in a probe request, it would reply to the device and initiate connection. This practice is discouraged, however, as it would provide sensitive data to anyone monitoring Wi-Fi probe requests. An attacker could read the SSID data and use it to identify the whereabouts of a device by mapping the PNL SSIDs with a database (such as Wigle) which connects SSIDs to geographic locations [3].

To mend this issue, devices send probe requests without asking for specific SSIDs. Instead, a "Broadcast" probe request is sent where the preferred SSID's MAC address is referred to as a "Wildcard" and is given the value "ff:ff:ff:ff:ff:ff". APs receiving a broadcast probe request would reply with their own SSID allowing the device to finally identify the desired AP and initiate connection. An important thing to note is that this approach is most used in Android devices while the PNL approach is still used by Apple devices [4].

The abundance of probe requests sent by any device with Wi-Fi capability poses security and privacy risks. Researchers have studied the effectivity and useability of MAC address randomization and have found that MAC address randomization has not been adopted by most operating systems (OS), and even devices that do randomize their MAC address, such as devices with the latest Android OS, still periodically transmit their global MAC address anyway [5]. MAC address randomization in mobile devices has been shown to fail in protecting privacy on several occasions [6, 7, 8]. Researchers have regularly used the time difference between consecutive probe request frames, inter-frame arrival times (IFAT), as the component utilized in identifying devices which have their MAC addresses randomized [9]. Some attempts have been made where the IFAT would be split up into different features like a signature time window while taking into consideration random mac address lifetime [10]. Probe requests coming from two different MAC addresses, but having close IFATs, would be categorized as coming from the same device since the difference in IFAT would be insignificant, and both IFATs would fall into the same bin [11]. Furthermore, researchers have been able to identify if the device's screen is on or off simply by studying the frequency of probe requests being sent from the device. Mobile devices with their screen on would send more probe requests than those with their screen off [12]. Other approaches looked at different channels in the frequency spectrum and utilized attributes relating to the transitioning happening between channels during active scanning [13]. Most approaches with more robust and accurate results attempt a combination of IFAT and other frame attributes, both statistically [14, 15, 16] and with utilizing machine learning [17]. Finally, many researchers have utilized these approaches for estimating pedestrian density and counting humans in an area, stressing that probe requests provide a non-invasive way of tracking and counting devices without affecting the privacy of users [18, 19, 20].

From a physical layer point of view, researchers have also been capable of identifying devices and even the network card manufacturer by studying the radio frequency (RF) signal properties themselves for each device [21]. This works by analyzing Wi-Fi RF signals and extracting physical differences such as imperfect oscillator and imbalanced I/Q, which act as features utilized for fingerprinting [22]. These methods not only identify devices, but can also classify user activity as well as their current pose [23, 24]. However, these methods require high-cost equipment for sampling RF signals and developing software defined radio (SDR) to process the signals in their purest form.

While the studied methods have proved to be effective, there is evidence to show that they could be counteracted as chip manufacturers have started adding random noise into probe requests to improve privacy. Even though frame inter-arrival times get slightly randomized, approaches such as binning IFATs have been proposed to allow device identification; however, these approaches have lower accuracies as the binning intervals are too short for any real identification of unique patterns.

B. Wi-Fi Scanning Permissions

Modern smartphone operating systems require applications to request permissions when accessing sensitive data or system

resources. Wi-Fi scanning is considered to be one of these resources. Any application with access to a list of nearby Wi-Fi network SSIDs could cross-evaluate with a database such as Wigle and have access to the user's location without their explicit consent [25].

However, apps can circumvent the permission model and gain access to protected data without user consent by using both covert and side channels [26, 27, 28, 29, 30, 31]. One example of that would be an app without permissions communicating with another app that does have permission to access the Wi-Fi list. The Android permission system has been studied in relevant work and its security principle was found to be vulnerable to applications circumventing the permission model by utilized convert and side channels. In terms of obtaining location information without any location-related permissions on Android, The authors in [32] utilize the currently connected APs BSSID, which is available to any Android application. However, the result is inaccurate because only one APs BSSID could be used. The authors in [33, 34] prove that ACCESS WIFI STATE permission can be used to obtain coarse-grained location information. Furthermore, the authors in [35] proposed a method that is able to obtain the list without access to location-related permissions and by using a drift-adjusting algorithms utilizing historical information location, average signal strengths, and the changing Wi-Fi list, the authors showed highly accurate tracking of users and detection of their daily activities.

III. Proposed Framework

A. Wi-Fi Scanning

A case study on the latest Android version (Android 12) is proposed. First, the feasibility of having access to the Wi-Fi list without explicit permission for accurate location is investigated on different Android APIs. Second, Wi-Fi lists are logged periodically for a certain period of time while a user with the smartphone goes about their day. Finally, the Wi-Fi list SSIDs are matched with the Wigle database.

B. Probe Request Identification

Wireshark is utilized to collect Wi-Fi packets through a cheap wireless adapter capable of entering monitor mode. Next, the packets are filtered for containing only probe requests. The filter used to get clean packets is shown in Figure The selected packets are then exported in JSON format for further processing. A Python script first anonymizes the entire JSON file. The script collects all MAC addresses, broadcast SSIDs, and specific frame data. The script then assigns for each unique MAC address, a completely randomized MAC address and removes all specific transmitter address (ta) and sender address (sa) data. Finally, the JSON file is overwritten and data that could possible be linked to a device or a person could be recovered.

A statistical study is then conducted on what possible tags could be used in identifying and clustering devices. The distribution of management tags is studied, and the relevant tags are selected for filtering and identifying devices. As a last step, weights are assigned to each tag according to its number of occurrences and the entropy of its values.

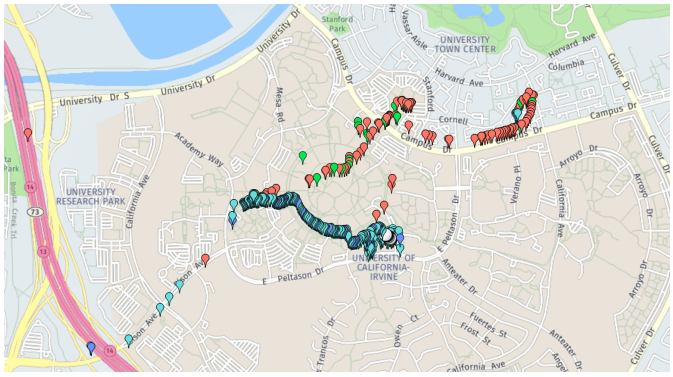

Fig. 1. Tracked user through Wi-Fi APs and Wigle

Finally, a network graph is reconstructed to attempt device identification. Each node is identified by a MAC address, and weighted edges connect each node. The weights of the edges correspond to the similarity between each node's tags. A threshold is calculated to lie 80% of the way between the edge with the minimum weight and the edge with the maximum weight. All edges below this threshold are then removed since they are deemed irrelevant. Finally, the network graph is presented and nodes with 90% correlation or more are labeled as the same device.

IV. Experimental Results

A. Wi-Fi Scanning

Using Android Studio, a simple app was created to test the permissions that allow for Wi-Fi scanning capabilities. The results show that any app created for an Android device running a version prior to Android 8 (API level 26) allows any app to get Wi-Fi scanning without permission. In Android 8.0 and Android 8.1, any app having the $CHANGE_WIFI_STATE$ permission, a permission unrelated to location privacy, is allowed access to the Wi-Fi list. Only for Android 10 (API level 29) and higher, does Android require explicit permissions to $ACCESS_WIFI_STATE$ and $ACCESS_FINE_LOCATION$. However, it has been shown that these permissions could easily be circumvented using side channels. A Wi-Fi list logger is installed on a phone running Android 12, and Wi-Fi SSIDs are saved in a database. Finally, the database is compared with Wigle. The output was able to completely track the whereabouts of the use as seen in Figure 1.

B. Probe Request Identification

A Wi-Fi adapter capable of entering promiscuous monitor mode must be utilized for Wi-Fi sniffing. The Alfa (AWUS036ACM) network card was chosen for conducting the experiment as it has both 2.4 GHz and 5 GHz capabilities as well as support for all IEEE 802.11 ac/a/b/g/n standards. The adapter was connected to a laptop running Windows, and Wireshark was utilized for packet sniffing. A OnePlus 9T device turns on its Wi-Fi. The device's MAC address is

chosen to be randomly changing, and after a while, the MAC address is changed to be static for comparison. The packets were then filtered to keep only probe request frames, removing all possibly private or sensitive communication data. The probe request packets are exported in JSON and anonymized. A code snippet is shown in Figure 2. Figure 3 illustrates the setup and workflow of the experiment's framework.

```
def generate_mac():
    return ':'.join(''.join(random.choice(letters) for i in range(2)) for j in range(6))

with open('Collected_Data/new_static.json','r') as f:
    data = f.read()

letters = [char for char in '0123456789abcdef']
parsed_data = json.loads(data, object_pairs_hook=dict_raise_on_duplicates)

filtered_packets = [elem for elem in parsed_data if 'wlan' in elem['_source']['layers']]
filtered_packets = [elem for elem in filtered_packets if 'wlan.mgt' in elem['_source']['layers']]
filtered_packets = [elem for elem in filtered_packets if 'wlan.fc.type_subtype' in elem['_source']['layers']['wlan']]
filtered_packets = [elem for elem in filtered_packets if elem['_source']['layers']['wlan']['wlan.fc.type_subtype']=='0x0004']
filtered_packets = [elem for elem in filtered_packets if len(elem['_source']['layers']['wlan.mgt']['wlan.tagged.all'])>0]
filtered_packets = [elem for elem in filtered_packets if 'wlan.ta_resolved' in elem['_source']['layers']['wlan']]

randomized_mapping = dict()
mapping = dict()
for frame in filtered_packets:
    sa = frame['_source']['layers']['wlan']['wlan.sa']
    sa_resolved = frame['_source']['layers']['wlan']['wlan.sa_resolved']
    ta = frame['_source']['layers']['wlan']['wlan.ta']
    ta_resolved = frame['_source']['layers']['wlan']['wlan.ta_resolved']
    mapping[sa] = [sa]
    mapping[sa].append(sa_resolved)
    mapping[sa].append(ta)
    mapping[sa].append(ta_resolved)
    if sa in randomized_mapping:
        continue
    random_sa = generate_mac()
    while random_sa in randomized_mapping:
        random_sa = generate_mac()
    randomized_mapping[sa] = random_sa
for elem in randomized_mapping:
    for subelem in mapping[elem]:
        data = data.replace(subelem,randomized_mapping[elem])
with open('anon.json','w+') as f:
    f.write(data)
```

Fig. 2. Anonymization code snippet

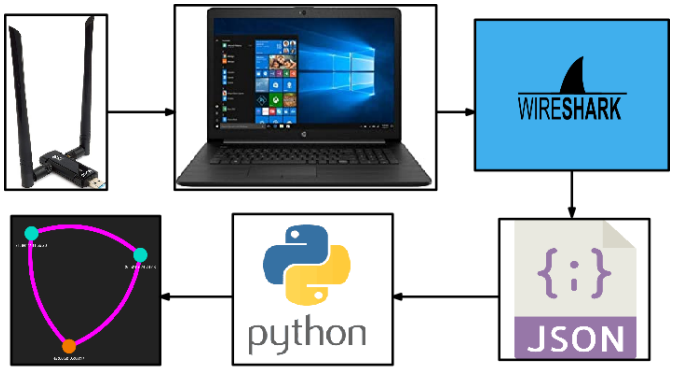

Fig. 3. Experimental setup and workflow.

The probe requests are then analyzed, and relevant attributes are identified and categorized as shown in Table I.

A study is conducted to analyze the availability and frequency of these tags in the probe requests found. Figure illustrates the results found. OUI information is present in most probe requests as well as requested SSIDs. Furthermore, most devices include their supported rates as well as extended capabilities and high throughput (HT) and very high throughput (VHT) capabilities. However, most tags found are barely found in most probe requests due to different firmware which could allow for easy identification. Furthermore, it would seem that only new devices do present extensive supported capabilities as these rates have only been available in phones recently. Furthermore, as phone network adapters have been more and more standardized, the differences in these capabilities would in the future become negligible removing them as viable options for utilizing in fingerprinting. Figure 4 shows the

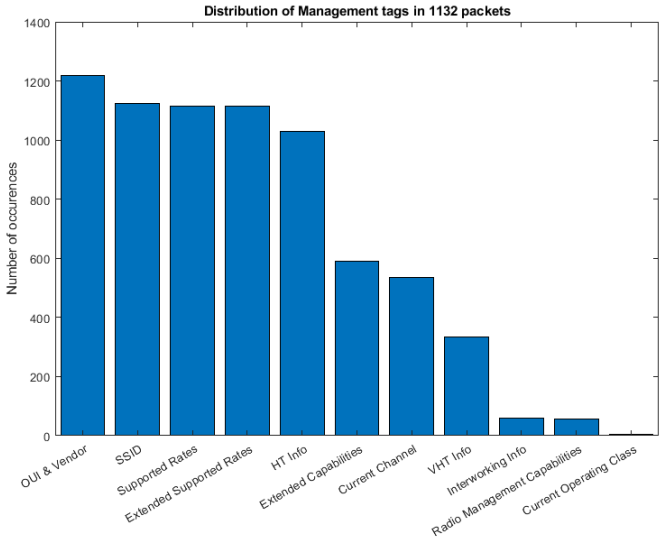

Fig. 4. Distribution analysis on availability of probe request tag attributes.

frequency of the the most available attributes in the probe requests collected.

Weights are assigned to each tag according to their uniqueness and availability, and finally a network is created and visualized using the PyViz library. Figure shows the network at a high edge weight threshold. Here, we notice the three highlighted connected nodes with thick edges corresponding to the user's OnePlus 9T device. This behavior is correctly validated as the device was sending some of its probe requests to a specific SSID, which is known to the author, in its saved networks.

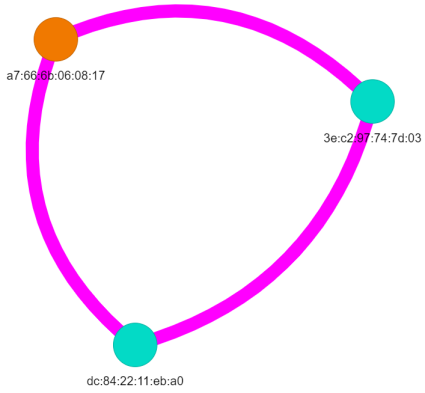

Fig. 5. Identified device network graph.

A reduction of the weight threshold shows some weaker connection between nodes, but these connections could arise due to similarities in the devices' operating system and manufacturer instead of belonging to the same device. Figure 6 illustrates the network graph that can be drawn at a slightly lower threshold.

V. Conclusion & Future Work

This article analyzed the utilization of Wi-Fi as a possible source of privacy concerns. First, an app that has direct access

TABLE I
RELEVANT ATTRIBUTES IN PROBE REQUEST FRAME.

Frame	Antenna	Extended Capabilities
frame.len	radiotap.dbm_antsignal	wlan.ext_tag.he_mac_caps
frame.time_delta		wlan.extcap

Interworking	VHT Capabilities	WLAN Management
interworking.access_network_type	vht.capabilities	wlan.seq
interworking.internet	vht.mcsset.rxmcsmap	wlan.supported_rates
interworking.asra	vht.mcsset.rxhighestlonggirate_raw	wlan.ds.current_channel
interworking.uesa	vht.ncsset.reserved	wlan.extended_supported_rates
interworking.hessid	vht.ncsset.ext_nss_bw_cap	

Sub 1GHz	HT Capabilities	Device Information
s1g.rps.raw_control	ht.capabilities	wlan.tag.oui
s1g.rps.raw_slot_definition	wlan.ht.ampduparam	wlan.vendor.oui.type
s1g.rps.channel_indication	wlan.ht.mcsset.rxbitmask	wlan.wfa.ie.type

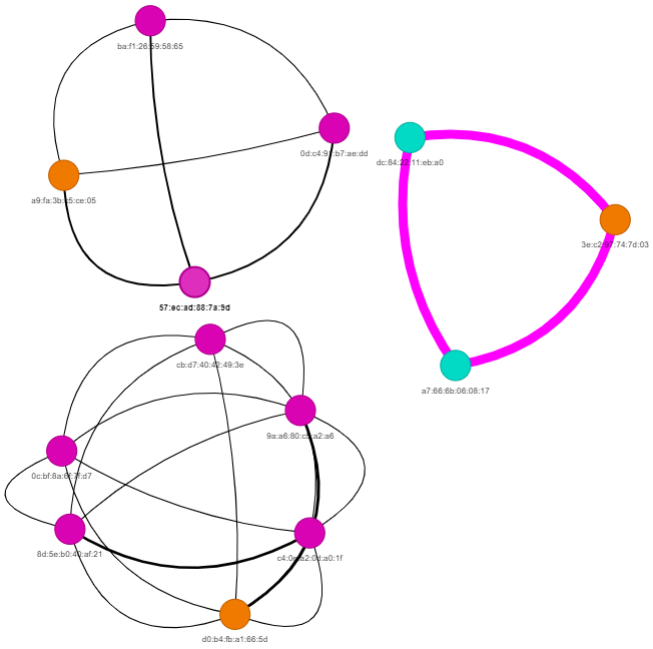

Fig. 6. Device Network Graph at lower threshold

to the phone and to the scanned Wi-Fi list was demonstrated to be capable of localizing and tracking the user without explicit access to geolocation permissions. Next, a non-intrusive approach where without any access to the user's device, the identifiability and uniqueness of the device through only the usage of probe requests was demonstrated. Future work could include performing a complete study on the legality of Wi-Fi sniffing, specifically probe requests. Another direction would be to include a more robust IFAT algorithm for fingerprinting and utilizing machine learning for more precise results.

REFERENCES

[1] Genevieve Bartlett, John Heidemann, and Christos Papadopoulos. "Understanding passive and active service discovery". In: *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. 2007, pp. 57–70.

[2] Balamurugan Soundararaj, James Cheshire, and Paul Longley. "Estimating real-time high-street footfall from Wi-Fi probe requests". In: *International Journal of Geographical Information Science* 34.2 (2020), pp. 325–343.

[3] Eric McKinion and Alan Lin. "Evaluation of security flaws in the current probe request design and proposed solutions". In: *International Conference on Cyber Warfare and Security*. Academic Conferences International Limited. 2017, p. 529.

[4] Ante Dagelić, Toni Perković, and Mario Čagalj. "Location Privacy and Changes in WiFi Probe Request Based Connection Protocols Usage Through Years". In: *2019 4th International Conference on Smart and Sustainable Technologies (SpliTech)*. IEEE. 2019, pp. 1–5.

[5] Luiz Oliveira et al. "Mobile device detection through WiFi probe request analysis". In: *IEEE Access* 7 (2019), pp. 98579–98588.

[6] Jeremy Martin et al. "A Study of MAC Address Randomization in Mobile Devices and When it Fails." In: *Proc. Priv. Enhancing Technol.* 2017.4 (2017), pp. 365–383.

[7] Célestin Matte et al. "Defeating MAC address randomization through timing attacks". In: *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. 2016, pp. 15–20.

[8] Mathy Vanhoef et al. "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.

[9] Sandhya Aneja, Nagender Aneja, and Md Shohidul Islam. "IoT device fingerprint using deep learning". In: *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*. IEEE. 2018, pp. 174–179.

[10] Pegah Torkamandi, Ljubica Kärkkäinen, and Jörg Ott. "An online method for estimating the wireless device count via privacy-preserving wi-fi fingerprinting". In: *International Conference on Passive and Active Network Measurement*. Springer. 2021, pp. 406–423.

[11] Christin Groba. "Demonstrations and people-counting based on Wifi probe requests". In: *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. IEEE. 2019, pp. 596–600.

[12] Shuja Jamil et al. "Classifying smartphone screen ON/OFF state based on wifi probe patterns". In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. 2016, pp. 301–304.

[13] Wyatt Praharenka and Ioanis Nikolaidis. "Identifying device type from cross channel probe request behavior". In: *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 2021, pp. 392–394.

[14] Jiajie Tan and S-H Gary Chan. "Efficient Association of Wi-Fi Probe Requests under MAC Address Randomization". In: *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE. 2021, pp. 1–10.

[15] Mathieu Cunche, Mohamed-Ali Kaafar, and Roksana Boreli. "Linking wireless devices using information contained in Wi-Fi probe requests". In: *Pervasive and Mobile Computing* 11 (2014), pp. 56–69.

[16] Hande Hong, Girisha Durrel De Silva, and Mun Choon Chan. "Crowdprobe: Non-invasive crowd monitoring with wi-fi probe". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.3 (2018), pp. 1–23.

[17] Xiaolin Gu et al. "Probe request based device identification attack and defense". In: *Sensors* 20.16 (2020), p. 4620.

[18] Woramate Pattanusorn et al. "Passenger estimation system using Wi-Fi probe request". In: *2016 7th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*. IEEE. 2016, pp. 67–72.

[19] Ooi Boon Yaik et al. "Measuring the accuracy of crowd counting using Wi-Fi probe-request-frame counting technique". In: *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* 8.2 (2016), pp. 79–81.

[20] Edwin George Vattapparamban. "People counting and occupancy monitoring using wifi probe requests and unmanned aerial vehicles". In: (2016).

[21] Pengfei Liu et al. "Real-time identification of rogue WiFi connections using environment-independent physical features". In: *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE. 2019, pp. 190–198.

[22] Jingyu Hua et al. "Accurate and efficient wireless device fingerprinting using channel state information". In: *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE. 2018, pp. 1700–1708.

[23] Fangzhan Shi, Kevin Chetty, and Simon Julier. "Passive activity classification using just WiFi probe response signals". In: *2019 IEEE Radar Conference (RadarConf)*. IEEE. 2019, pp. 1–6.

[24] Mingmin Zhao et al. "Through-wall human pose estimation using radio signals". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7356–7365.

[25] Joel Reardon et al. "50 ways to leak your data: An exploration of apps' circumvention of the android permissions system". In: *28th USENIX security symposium (USENIX security 19)*. 2019, pp. 603–620.

[26] Raphael Spreitzer et al. "Systematic classification of side-channel attacks: A case study for mobile devices". In: *IEEE Communications Surveys & Tutorials* 20.1 (2017), pp. 465–488.

[27] Le Nguyen et al. "Unlocin: Unauthorized location inference on smartphones without being caught". In: *2013 International Conference on Privacy and Security in Mobile Systems (PRISMS)*. IEEE. 2013, pp. 1–8.

[28] Yan Michalevsky et al. "{PowerSpy}: Location Tracking Using Mobile Device Power Analysis". In: *24th USENIX Security Symposium (USENIX Security 15)*. 2015, pp. 785–800.

[29] Claudio Marforio et al. "Analysis of the communication between colluding applications on modern smartphones". In: *Proceedings of the 28th Annual Computer Security Applications Conference*. 2012, pp. 51–60.

[30] Kenneth Block, Sashank Narain, and Guevara Noubir. "An autonomic and permissionless android covert channel". In: *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 2017, pp. 184–194.

[31] Ahmed Al-Haiqi, Mahamod Ismail, and Rosdiadee Nordin. "A new sensors-based covert channel on android". In: *The Scientific World Journal* 2014 (2014).

[32] Xiaoyong Zhou et al. "Identity, location, disease and more: Inferring your secrets from android public resources". In: *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 2013, pp. 1017–1028.

[33] Piotr Sapiezynski et al. "Tracking human mobility using wifi signals". In: *PloS one* 10.7 (2015), e0130824.

[34] Jagdish Prasad Achara et al. "Short paper: Wifileaks: Underestimated privacy implications of the ACCESS_WIFI_STATE Android permission". In: *Proceedings of the 2014 ACM conference on Security and privacy in wireless & mobile networks*. 2014, pp. 231–236.

[35] Fenghua Li et al. "TrackU: Exploiting User's Mobility Behavior via WiFi List". In: *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE. 2017, pp. 1–6.