Probing with a Generic MAC Address: An Alternative to MAC Address Randomisation

1st Johanna Ansohn McDougall
Security in Distributed Systems
University of Hamburg
Hamburg, Germany
johanna.ansohn.mcdougall@uni-hamburg.de

2nd Alessandro Brighente
Department of Mathematics
University of Padova Padua, Italy
alessandro.brighente@unipd.it

3rd Anne Kunstmann
Security in Distributed Systems
University of Hamburg
Hamburg, Germany
anne.kunstmann@uni-hamburg.de

4th Niklas Zapatka
Security in Distributed Systems
University of Hamburg
Hamburg, Germany
niklas.zapatka@uni-hamburg.de

5th Hanna Schambach
Security in Distributed Systems
University of Hamburg
Hamburg, Germany

6th Hannes Federrath
Security in Distributed Systems
University of Hamburg
Hamburg, Germany
hannes.federrath@uni-hamburg.de

***Abstract*—Active scanning via probe requests is a means for mobile devices to detect known networks. To protect the device from being tracked via an unchanging identifier contained in the probe request, MAC address randomisation is used. While it has been in use since 2014, the standardisation of MAC address randomisation is still in its draft stage. This also leads to manufacturers devising their own randomisation schemes, some of which have been proven insufficient to prevent tracking. In this paper, we strive to reignite the discussion on standardising the use of Locally Administered Addresses (LAAs) like randomised MAC addresses. To overcome the limitations of MAC address randomisation, we propose the use of one generic address over all devices. We implement and test this scheme, and additionally ascertain that a generic MAC address not only enhances user anonymity during probing but also offers operational efficiency comparable to MAC address randomisation. In conclusion, this contribution highlights the need for a standardised approach to preserve device anonymity, and simultaneously introduces a novel alternative of employing a single generic address across devices.**

***Index Terms*—Probe Requests, Wi-Fi Tracking, Privacy Preserving Technologies, MAC Address Randomisation, Alternative, Generic MAC Address**

I. Introduction

While a privacy-preserving means of network discovery, namely passive scanning, exists, the predominantly used mechanism is active scanning: In search of available networks, mobile devices transmit probe requests and monitor for probe responses containing known Service Set Identifiers (SSIDs). To protect users from being tracked via the MAC address contained in the probe requests, modern devices typically use MAC address randomisation to hide the identity of the sender: A regularly changing Locally Administered Address (LAA) is used instead of the real hardware address.

Manufacturers use various approaches to implement MAC address randomisation [17], [22], as well as researchers propose their original schemes [6], [10]. While standardisation of MAC address randomisation was proposed in February 2022, it is, as of today, still in its draft stage [16]. Lacking a clearly defined and standardised approach, manufacturers, therefore, implement randomisation as they see fit. One approach is to keep the first three bytes of the MAC address, also known as the Organizationally Unique Identifier (OUI), unchanged, and randomise only the last three bytes [17]. However, this still allows attackers to infer information on devices, and possibly track and trace them [22].

To increase privacy and explore alternatives to existing approaches, we propose the use of a generic MAC address during active discovery. The use of a generic address extends the idea to reduce the probe request content in general, such as suggested by [3]. In combination, both approaches make probe requests less distinguishable, and eliminate the possibility to infer information on devices via the OUI or the probe request content in general. In this publication, we provide a proof-of-concept of the use of a generic MAC address and evaluate it both with respect to scalability as well as in comparison to other schemes. To this end, we contribute the following:

- To defend against attacks targeting MAC address randomisation, we propose and implement the use of a generic address. Our results show that MAC address randomisation can be replaced by a single fixed MAC address for all devices.
- We test the use of a generic address across a number of devices to ensure it scales well. The results show that connection establishment is not impeded by several devices probing with the same MAC address.
- We further show that the time required for connection establishment using our scheme is comparable to that of MAC address randomisation, but providing higher privacy guarantees than schemes using a fixed OUI.

This paper is structured as follows: We provide a background on network discovery, connection establishment and the Time-to-Traffic metric in Section II, and Related Work in Section III. Section IV introduces the attacker model.

Section V presents the implementation and test setup to verify the feasibility of using a generic address, and presents the results of the tests. We subsequently discuss our work and conclude it in Section VII.

II. Background

In this section, we first provide a background on MAC addresses and network discovery and subsequently give an overview over connection establishment in Wi-Fi networks. Afterwards, we introduce the Time-to-Traffic metric, which we require to measure the duration of connection establishment in our experimental setup in Section V.

A. MAC Addresses

A Media Access Control (MAC) address is an identifier used in Wi-Fi networks. It has a length of 48 bits [15]. The I/G bit is the least significant bit of the most significant byte and concerns multicast or unicast addressing [16]. The U/L bit, the second-least significant bit of the most significant byte, shows whether a MAC addresses is locally or universally administered [14]. A Universally Administered Address (UAA), sometimes called the hardware address or burned-in address, is the permanent device identifier. Its first 24 bits constitute the Organisationally Unique Identifier (OUI), and are assigned to the manufacturer of the device. The last 24 bits, the Network Interface Controller (NIC) are assigned to the device by its manufacturer. A common example of a Locally Administered Address (LAA) is an address generated via MAC address randomisation [16]. There are various randomisation schemes depending on the manufacturer of the device, the two most common ones being (i) 46-bit Randomisation, where all bits except for the U/L bit and the I/G bit are randomised, and (ii) randomisation with a persistent OUI, where only the last 24 bits are randomised [17]. In Apple devices, MAC address randomisation was first implemented starting with iOS 8 in 2014 [1]. Android, on the other hand, introduced MAC address randomisation in 2015 in Android 6.0 [2].

B. Network Discovery

To identify suitable networks within reach, a client can actively query for Access Points (APs) using probe requests. This process is called active discovery or active scanning. The probe request can be *directed*, containing the Service Set Identifier (SSID) of an AP, or *undirected*, containing an empty SSID field. The latter is the common case, as the transmission of SSIDs can reveal potentially private information on users and serves as a fingerprint of the device [4]. The first, on the other hand, is used to locate hidden networks, or the case in outdated mobile Operating Systems (OS) or misconfiguration by the users [4]. An AP receiving a probe request can respond with a probe response, containing their SSID. Upon identifying a known network in transmission range, a device can initialise connection establishment.

The counterpart to active discovery is passive discovery. Here, APs advertise themselves via beacons every 102.4 ms. Passive discovery is privacy friendly, since it doesn't require for mobile devices to transmit any information to locate nearby networks. Despite this, active discovery is the more prevalently used method of network discovery due to its reduced overhead [9].

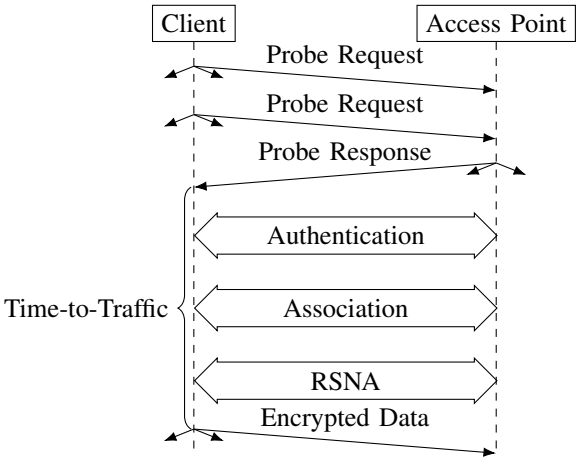

Fig. 1: IEEE 802.11 connection establishment with Time-to-Traffic. The device discovery via probe requests is followed by authentication, association and Robust Security Network Association (RSNA), subsequent to which encrypted data can be exchanged.

C. Connection Establishment in Wi-Fi Networks

The IEEE 802.11 standard defines Wireless Local Area Networks (WLANs) [15]. While WLAN is the technical term describing the standard, the Wi-Fi Alliance instead encourages the utilisation of the term Wi-Fi, which is a trademark protecting certified products with Wi-Fi interoperability [23]. Wi-Fi is the de-facto standard term used in anglophone publications, which we adhere to as well in this publication.

A Wi-Fi network is identified by its SSID, and Wi-Fi capable devices can locate known networks via network discovery as described in Section II-B. Upon identifying a known network, a client can initiate connection establishment. In Wi-Fi networks, this consists of three steps according to the current WPA standard: authentication, association and Robust Security Network Association (RSNA). The IEEE 802.11 authentication and association are a relic of the WEP protocol and are still maintained to grant access to the network and enable the transfer of frames on higher layers. Up-to-date security parameters and mechanisms are subsequently negotiated and exchanged in RSNA, after which data frames can be encrypted before transmission. An example of connection establishment is depicted in Fig. 1.

D. Time-to-Traffic

We introduce the Time-to-Traffic (TtT) metric for measuring the duration of the connection process. The active establishment of a connection begins with the first transmission of probe requests by the client, and concludes with data transfer between the client and the AP. For the sake of reproducibility, instead of choosing the initial probe request as the starting

point, we await the first probe response from the AP before commencing the measurement to ensure server availability; otherwise, the waiting time in case the AP is busy or out of range would be included in the connection establishment duration, which is undesirable. Thus, we define the TtT as the **time span between the arrival of the first probe response from the AP and the transmission of the first data frame** (cf. Fig. 1). The TtT metric is applied in Section V.

III. Related Work

By monitoring probe requests sent by nearby devices, an eavesdropper can triangulate their origin. Additionally, the more identifying information is contained within the probe requests, the easier it is to fingerprint a device. In the following, we present existing attacks on probe requests in general, and MAC address randomisation in particular. Subsequently, we highlight different strategies to circumvent the use of the hardware address.

A. Attacks on MAC Address Randomisation and Probe Requests

A paper drawing much attention to the privacy implications of the use of probe requests was published by Freudiger et al. [7] in 2015, who quantify the amount of probe requests sent by various different devices. They identify an additional threat that allows the re-identification despite MAC address randomisation: the use of sequential and unrandomised sequence numbers. Subsequently, Vanhoef et al. [22] presented their 2016 study, revealing that the Information Element also provides enough information to fingerprint a device and track it over a significant period of time. They additionally demonstrate two attacks that can be used to reveal the hardware address of a device. In 2017, Martin et al. [17] conducted a broad study on the use of MAC address randomisation across various devices and identified other ways to circumvent MAC address randomisation. They discovered that a significant amount of Android devices persist the OUI, randomising only the last 24 Bytes of the address.

In their 2021 study, Fenske et al. [5] study whether the results of Martin et al. [17] still reflect the current pervasiveness of MAC address randomisation, or whether notable changes occurred during the three years between the two studies. They find that while in 2016, around 82 % of the devices sent probe requests using their hardware address, the amount was reduced significantly between 2019 and 2020, to 56 % of the tested devices and operating systems (OSes). When analysing probe requests sent by devices with deactivated Wi-Fi, they identified one device that continued probing with its hardware address. They additionally manifest that a variety of devices running Android periodically cycle through their hardware address during active discovery. Their results show that MAC address randomisation has, as of 2021, not yet been deployed consistently and pervasively, and its implementation is still, if less, lacking in certain places, with some manufacturers still relying on fixed OUIs or periodically leaking the hardware address.

Another publication that analyses the pervasiveness of MAC address randomisation was published by Gomez et al. [8]. The authors evaluate data from public Wi-Fi networks, collected between 2016 and 2021 in different Latin American countries. They state that even though randomisation was first implemented starting in 2014, its wide-spread use only started in 2020.

The remaining recent research on de-randomisation strategies for probe requests puts a much larger emphasis on fingerprinting the IE instead of inspecting the MAC address. This is to be expected, since there is only so much one can derive from (partially) randomised strings, while the IE provides a very useful fingerprint. With this respect, Gu et al. [11] utilise deep learning attacks on the IE, Tan et al. [20] and He et al. [12] perform minimum-cost flow optimisation and Uras et al. [21] and Pintor et al. [19] use clustering approaches. To protect users from transmitting such identifiers, several publications therefore suggest to reduce the complexity of the IE field [5], [17], [22].

B. MAC Address Obfuscation Strategies

In 2003, Gruteser et al. [10] suggested the use of disposable identifiers, namely a regularly changing LAA, to increase location privacy. Their approach concatenates an existing OUI, randomly chosen from an IEEE OUI assignment list to a 24-bit long part of a random string. This string is generated by using a random seed to initialise a chain of MD5 hashes. Part of the resulting 128-bit hash is used as the NIC (cf. Section II-A) of the MAC address, with each rotation of the MAC address using the subsequent element in the hash chain. Since Gruteser et al. suggest the application of their randomisation scheme to wireless communication in general and not only probe requests, they also have to take the probability of MAC collisions into account.

After having demonstrated the feasibility of fingerprinting wireless device drivers via their probe requests, Franklin et al. [6] suggest to circumvent such fingerprinting techniques by employing MAC address masquerading: Here, a device changes its own MAC address to that of another device in transmission range. This way, when attempting to infer information on devices by fingerprinting transmission characteristics, two different devices would exhibit the same MAC address and be indistinguishable from one another.

To protect user privacy in the face of insufficient randomisation schemes, Martin et al. [17] devise best practices for MAC address randomisation. These entail the randomisation of all the bits of the MAC address (excluding the U/L and the I/G bits), the use of a new randomised address for every transmitted frame, and never to transmit probe requests via the hardware address.

The research on de-randomisation strategies of recent years has put a distinct emphasis on the IE field, rightly suggesting to minimise the fingerprint generated via the IE field (cf. Section III-A). To additionally reduce the attack vector via the MAC address, we strive to reignite the discussion on standardising randomisation schemes to remove even more

identifiers from probe requests. To this end, we propose an alternative to the use of randomised MAC addresses during active discovery: The use of one generic address over all devices.

IV. Attacker Model

As common in the related literature, we consider a passive attacker who wants to infer the movement of people in an area by monitoring their device's probe requests. We assume that the attacker has a sufficiently large number of distributed receivers with which they can monitor an area, e.g. a large shop or mall or a university campus. They can distinguish globally and locally set MAC addresses via the U/L bit, and use stable OUIs of locally assigned MAC addresses to infer information on certain devices. We assume that this enables the attacker to track devices via recurring or stable elements of the MAC address over time.

V. Generic Address Scheme

Although MAC address randomisation guarantees high anonymity levels, it is still prone to correlation attacks undermining the anonymity of the user [17]. To counteract this attack, we propose the use of a generic address: This allows multiple devices to probe with the same MAC address to make single devices disappear in a large anonymity set: They become indistinguishable from one another as they share identical identifiers. The intricacy of the use of a generic address lies in the time during which a device maintains its generic address: only while sending probe requests. Once the device attempts to establish a connection, it has to switch to its UAA or per-network-LAA (cf. Section II-A) to avoid MAC address collisions in the network. The following sections first describe the implementation and test setup and subsequently present the test results.

A. Implementation

A tool suite for network configuration that provides the correct functionality is the NetworkManager[1]. It facilitates both tests of devices using randomised MAC addresses while probing, as well as manually setting MAC addresses. This is done via the setting `wifi.scan-generate-mac-address-mask` in the file `/etc/NetworkManager/conf.d/generate-mac-address.conf`. We modify the file to contain the following:

```
[device-wlan0]
    wifi.scan-generate-mac-address-mask=
    FF:FF:FF:FF:FF:FF 22:22:22:22:22:22
```

The setting takes two arguments, the first being a mask of bits that are to be set, and the second being the values that the bits are to be set to. The mask FF:FF:FF:FF:FF:FF ensures that all bits of the address are to be overwritten, and 22:22:22:22:22:22 is the value by which they are to be overwritten. This address was chosen since it is a universally available and unreserved address; certain MAC addresses are reserved for special purposes and would therefore not have been usable in the implementation of the generic address [13].

[1]https://networkmanager.dev/

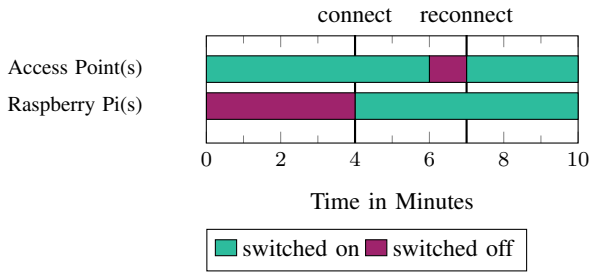

Fig. 2: One test run encompasses two data points: The Raspberry Pis are initially turned on after 4 minutes, where the first connection attempt with the AP is recorded. After 6 minutes, the AP is turned off for one minute and a second connection attempt is recorded after 7 minutes.

In order to test the scheme against NetworkManager running MAC address randomisation, MAC address randomisation has to be explicitly turned on using the `wifi.scan-rand-mac-address-mask` configuration.

The advantage of taking this approach using NetworkManager is that both the `wifi.scan-rand-mac-address-mask` setting as well as the `wifi.scan-gene- rate-mac-address-mask` setting modify the MAC address only *during scanning*, and return to the individual MAC address for association [18].

B. Scalability Analysis

We test our solution using five Raspberry Pis running the above-mentioned configuration of NetworkManager and two APs. The APs are implemented using a Wi-Fi dongle via which we provide two APs using hostapd on a split interface and a network bridge. Three of the Raspberry Pis possess the credentials of one AP, and the other two possess those of the second AP.

We test the generic address with test runs as depicted in Fig. 2. Each test run is started by setting up and turning on the APs. After four minutes, we turn on the Raspberry Pis and the probing and connection establishment with the APs can be monitored. After six minutes, the APs are turned off for a minute; the second probing and connection establishment can be monitored when they are turned on again in minute seven. The TtT is measured for each connection establishment, leading to two measurements per test run for each client device.

The results can be seen in Fig. 3: The average TtT for two devices is 9.31 seconds (14 test runs, 53 data points), 9.61 for three devices (12 test runs, 72 data points), 9.63 for four devices (5 test runs, 38 data points) and 9.77 seconds for five devices (5 test runs, 50 data points). In very few cases, the resulting captures did not contain certain required measurement points for the TtT, e.g., since they were either not sent or not recorded. Their values therefore had to be omitted. However, in all test runs considered in this comparison,

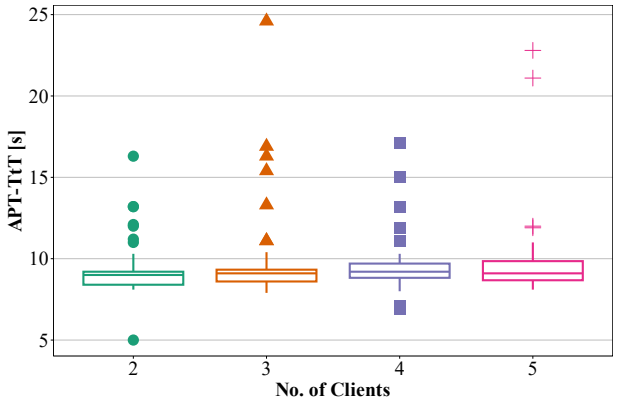

Fig. 3: A comparison of two to five devices and the time required to connect to a network

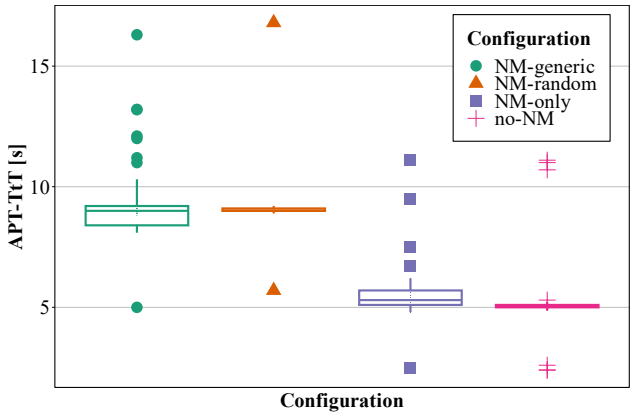

Fig. 4: A comparison of the Time-to-Traffic required in four different settings: without NetworkManager (no-NM), with NetworkManager (NM-only), with Generic Address Scheme (NM-generic) and with MAC address randomisation (NM-random)

the specified number of devices successfully established a connection.

The test runs show that while a slight increase of the TtT is detectable, the connection establishment requires a comparable time for two, three, four and five devices.

While this experiment was mainly conducted to find out whether connecting is possible with multiple devices probing with the same MAC address, we consider it future work to simulate connection attempts with a much larger amount of devices to estimate whether the minimal increase in TtT is consistent with an increasing amount of devices, or just a coincidence reflecting the variable amount of data points.

C. Comparative Evaluation

The use of a generic address is compared to other probing methods using four different settings: i) A Raspberry Pi connecting without the use of NetworkManager, ii) with the use of NetworkManager, iii) using a generic address implemented in NetworkManager, and iv) with MAC address randomisation set in NetworkManager. The results are presented in the following and can also be observed in Fig. 4.

NetworkManager with a Generic Address: The average TtT when using a generic address takes 9.31 seconds. The time required remains fairly stable with an increased amount of devices. This can also be observed in Fig. 3.

NetworkManager with MAC Address Randomisation: To compare the NetworkManager to a scheme that offers similar device privacy, we tested the same setup as above with devices running NetworkManager with MAC address randomisation turned on. In this setting, the TtT is 9.27 seconds on average.

Without Privacy-Enhancing Schemes: The subsequent tests were performed to determine whether the time required in the previous settings is representative of connection establishment without the use of privacy enhancing schemes, or whether both schemes decelerate the connection establishment. Therefore, the TtT both during connection establishment using NetworkManager without any additional configuration, as well as the connection establishment without the use of NetworkManager were tested.

In case of the Raspberry Pi connecting without the use of NetworkManager, the average time required was 5.36 seconds. When connecting with NetworkManager, the average time required was a bit higher than without, 5.59 seconds.

VI. Discussion

The tests show that the use of a generic address performs comparably well as MAC address randomisation in NetworkManager. They require 9.31 and 9.27 seconds, respectively, for their TtT from the first probe response to the first data stream. The TtT required without the use of NetworkManager is 5.36 seconds, which is why we evaluated whether the overhead of 4 seconds stems from the use of NetworkManager or the additional configuration in NetworkManager: When using NetworkManager without a privacy-enhancing scheme, the TtT was 5.59 seconds. NetworkManager therefore only causes an overhead of 0.23 seconds and the extra configurations in conjunction with the privacy-enhancing schemes are the causes of the overhead. Both MAC address randomisation and the use of a generic address cause a comparable overhead, but a generic address entails a larger privacy gain since it allows single devices to disappear in a large anonymity set, and simultaneously eliminates the possibility of inferring information via a stable OUI or a recurring use of the real hardware address. It should therefore be considered as a replacement for MAC address randomisation.

Our experiments aim at a proof-of-concept. The objective is to verify whether multiple devices can use the same MAC address for probing without a major overhead when compared to existing approaches. This is confirmed by our data. To additionally compare the scheme and it's efficiency to others, we vary the number of clients, encompassing two to five devices, and the device configuration to increase internal validity. Our results demonstrate the feasibility of the use of a generic address: The scalability analysis determines that a

varying amount of clients can connect within a comparable time frame, and the comparative evaluation shows that a generic address causes a similar overhead in NetworkManager as the use of MAC address randomisation, while yielding better privacy protection.

To perform experiments without the overhead induced by additional configurations of NetworkManager, both the generic address, as well as MAC address randomisation should be implemented in the device driver. Additionally, in order to analyse whether the scheme works on a large scale, we consider it future work to perform a large-scale simulation or real-world experiment.

The experiments conducted in this study additionally show, that colliding MAC addresses during network discovery are tolerable and can increase privacy in certain settings. Since the generic address is only used while probing, the subsequent connection establishment and connection are not influenced by our suggested modifications, since the device changes to its LAA or UAA before associating with a network.

VII. Conclusion

The lack of standardisation of MAC address randomisation drives manufacturers to implement their own schemes, sometimes even ones maintaining device information by persisting the OUI [17], [22] or recurrently disclosing the hardware address [5]. This can allow attackers to infer information on devices from the MAC address, despite the use of MAC address randomisation. But as MAC address randomisation is, as of today, not standardised yet [16], we strive to reignite the discussion on using a unified scheme to preserve user anonymity. To initiate the discussion on an alternative to the currently used schemes, we propose to make probe requests as indistinguishable as possible by sending them all from one generic MAC address. We evaluate the feasibility of this approach by implementing it using NetworkManager, and test it with five devices simultaneously. By calculating the Time-to-Traffic in various setting, we show that it is comparable in efficiency to MAC address randomisation. When combined with sequence number randomisation per frame, and the removal of IE content as suggested in [3], it leaves single devices indistinguishable from one another and maximises the anonymity set they disappear in.

References

[1] Aaron Mamiit: Apple implements random MAC address on iOS 8. Goodbye, marketers. https://www.techtimes.com/articles/8233/20140612/apple-implements-random-mac-address-on-ios-8-goodbye-marketers.htm (2014)

[2] Android Developers: Android 6.0 Changes. https://developer.android.com/about/versions/marshmallow/android-6.0-changes (2023)

[3] Ansohn McDougall, J., Brighente, A., Kunstmann, A., Zapatka, N., Federrath, H.: Reduce to the MACs - privacy friendly generic probe requests. In: Steven Furnell, Sokratis K. Katsikas, K.M., Pitropakis, N. (eds.) ICT Systems Security and Privacy Protection, 39th IFIP TC 11 International Conference, SEC 2024. Springer (2024)

[4] Ansohn McDougall, J., Burkert, C., Demmler, D., Schwarz, M., Hubbe, V., Federrath, H.: Probing for passwords – privacy implications of ssids in probe requests. In: Ateniese, G., Venturi, D. (eds.) Applied Cryptography and Network Security. pp. 376–395. Springer International Publishing, Cham (2022)

[5] Fenske, E., Brown, D., Martin, J., Mayberry, T., Ryan, P., Rye, E.C.: Three years later: A study of MAC address randomization in mobile devices and when it succeeds. PETS'2021 pp. 164 – 181 (2021)

[6] Franklin, J., McCoy, D., Tabriz, P., Neagoe, V., Van Randwyk, J., Sicker, D.: Passive data link layer 802.11 wireless device driver fingerprinting. In: USENIX Security'06. USENIX Association (2006)

[7] Freudiger, J.: How Talkative is Your Mobile Device? An Experimental Study of Wi-Fi Probe Requests. In: WiSec'15. ACM (2015). https://doi.org/10.1145/2766498.2766517

[8] Gomez, C.A., Guerrero, L.J., Pedraza, L.F.: Evolution of the use of random MAC addresses in public Wi-Fi networks. Journal of Engineering Science & Technology Review **15**(3) (2022)

[9] Goovaerts, F., Acar, G., Galvez, R., Piessens, F., Vanhoef, M.: Improving Privacy Through Fast Passive Wi-Fi Scanning. In: Secure IT Systems. pp. 37–52. Springer (2019)

[10] Gruteser, M., Grunwald, D.: Enhancing location privacy in wireless LAN through disposable interface identifiers: A quantitative analysis. In: Proceedings of the 1st ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots. p. 46–55. WMASH '03, Association for Computing Machinery, New York, NY, USA (2003). https://doi.org/10.1145/941326.941334

[11] Gu, X., Wu, W., Gu, X., Ling, Z., Yang, M., Song, A.: Probe Request Based Device Identification Attack and Defense. Sensors **20**(16) (2020). https://doi.org/10.3390/s20164620

[12] He, T., Tan, J., Chan, S.H.G.: Self-supervised association of Wi-Fi probe requests under MAC address randomization. IEEE Transactions on Mobile Computing (2022)

[13] (IANA), I.A.N.A.: Ethernet Numbers. https://www.iana.org/assignments/ethernet-numbers/ethernetnumbers.xhtml

[14] IEEE: IEEE Std 802-2014 - Local and Metropolitan Area Networks: Overview and Architecture. https://ieeexplore.ieee.org/document/6847097 (2014)

[15] IEEE: IEEE Std 802.11 - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9363693 (2020)

[16] JC. Zuniga, CJ. and Bernardos, Ed. and A. Andersdotter: Randomized and changing MAC address. https://datatracker.ietf.org/doc/html/draft-ietf-madinas-mac-address-randomization-10 (2023)

[17] Martin, J., Mayberry, T., Donahue, C., Foppe, L., Brown, L., Riggins, C., Rye, E.C., Brown, D.: A study of MAC address randomization in mobile devices and when it fails. PETS'2017 **4**, 268–286 (2017)

[18] NetworkManager: NetworkManager.conf — NetworkManager configuration file. https://networkmanager.dev/docs/api/latest/NetworkManager.conf.html (2024)

[19] Pintor, L., Atzori, L.: Analysis of wi-fi probe requests towards information element fingerprinting. In: GLOBECOM 2022 - 2022 IEEE Global Communications Conference. pp. 3857–3862 (2022). https://doi.org/10.1109/GLOBECOM48099.2022.10001618

[20] Tan, J., Chan, S.H.G.: Efficient association of Wi-Fi probe requests under MAC address randomization. In: INFOCOM'21. pp. 1–10. IEEE (2021)

[21] Uras, M., Cossu, R., Ferrara, E., Bagdasar, O., Liotta, A., Atzori, L.: Wifi probes sniffing: an artificial intelligence based approach for MAC addresses de-randomization. In: 2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD) (10 2020). https://doi.org/10.1109/CAMAD50429.2020.9209257

[22] Vanhoef, M., Matte, C., Cunche, M., Cardoso, L.S., Piessens, F.: Why MAC address randomization is not enough: An analysis of Wi-Fi network discovery mechanisms. In: Asia CCS '16. p. 413–424. ACM (2016). https://doi.org/10.1145/2897845.2897883

[23] wikipedia: Wi-Fi. https://en.wikipedia.org/wiki/Wi-Fi (2023)