Cross-Layer Device Fingerprinting and Its Applications to Network Security

Li Xue
Baskin School of Engineering
University of California, Santa Cruz
Santa Cruz, USA
xueli@ucsc.edu

Katia Obraczka
Baskin School of Engineering
University of California, Santa Cruz
Santa Cruz, USA
katia@soe.ucsc.edu

Zouheir Rezki
Baskin School of Engineering
University of California, Santa Cruz
Santa Cruz, USA
zrezki@ucsc.edu

***Abstract*—In this paper, we describe a novel cross-layer fingerprinting approach (CL-FP) that aims at uniquely identifying wireless devices by extracting inherent cross-layer device features based on the requirements of the driving network application(s). We demonstrate how our system can be applied to MAC spoofing detection. Our contributions include: (1) We propose the CL-FP framework and pipeline for detecting MAC spoofing attacks, which, to our knowledge, is the first of its kind; (2) We experimentally confirm theoretical results showing that easier-to-extract FP features can reliably represent harder-to-extract intrinsic physical characteristics of devices; (3) We test and evaluate the performance of the proposed CL-FP approach through simulations in the context of the MAC spoofing detection use case. Our preliminary results show that the proposed CL-FP pipeline provides a lightweight, scalable and reliable end-to-end cross-layer device fingerprinting framework.**

***Index Terms*—Cross-Layer, device fingerprinting, MAC spoofing.**

I. Introduction

As the number of end user devices connected to the Internet and their applications continue to dramatically increase, managing and administering edge and access networks have become increasingly more challenging. For instance, ensuring network security via detecting and denying or restricting access unauthorized devices in the era of the Internet of Things (IoT) is of paramount importance. Flexible, efficient and effective security mechanisms are one of the most important targets in the design of today's wireless networks. Resource limitations that characterize edge devices require security solutions that have small resource requirement footprint.

Device fingerprinting (FP) techniques that aim to enhance wireless network security have gained attention from the research community in recent years. FP techniques typically use device intrinsic features as a way to uniquely identify devices. Radio Frequency (RF) FP is the most common form of fingerprinting suitable for wireless networks. RF-FP is a physical layer security mechanism that extracts inherent and unique physical layer features related to hardware manufacture imperfections or imbalances in the waveform of a transmitter [1] [2] [3] [4]. There has also been work on using RF features to assist conventional network authentication [5] [6] [7] [8]. In these RF-FP applications, physical layer features are used independently or together with information from higher network layers, e.g., MAC layer information [8].

Although device fingerprinting has been the focus of some research efforts, there are still some important gaps that need to be addressed. Most existing RF-FP approaches extract and analyze only physical layer characteristics. Most current methodologies do not evaluate the scalability and flexibility of the selected features in relation to the driving application, which may require an integrated approach using FP features from different network protocol layers rather than from the physical layer only. There has been some work on applying integrated features from the physical and MAC layers [8], but there is not yet a well-defined framework for selecting appropriate features from various network layers based on the application's requirements. In addition, most existing techniques rely solely on experimental findings to select and extract FP features. Empirical approaches make it difficult to select appropriate features for specific applications without a thorough theoretical understanding. Lastly, current techniques largely overlook the intrinsic complexity of FP and instead rely entirely on the correctness of the proposed FP analysis algorithms. Consequently, the complexity and computational resource needs of existing FP techniques may be too high and thus impractical for deployment in low-end edge devices.

Our work focuses on cross-layer device fingerprinting (CL-FP) with the goal of closing the gaps discussed above, opening the way for the deployment of an end-to-end, scalable, lightweight, and reliable FP framework. The proposed CL-FP aims at identifying and extracting inherent cross-layer device features based on the requirements of the driving network application(s). In this paper, we focus on the MAC spoofing attack detection use case.

Contributions: Our contributions include: (1) We propose the CL-FP framework and pipeline for detecting MAC spoofing attacks, which, to our knowledge, is the first of its kind; (2) We experimentally confirm theoretical results showing that easier-to-extract FP features can reliably represent harder-to-extract intrinsic physical characteristics of devices; (3) We test and evaluate the performance of the proposed CL-FP approach through simulations in the context of the MAC spoofing detection use case. Our preliminary results show that the proposed CL-FP pipeline provides a lightweight, scalable and reliable end-to-end cross-layer device fingerprinting framework.

Roadmap: The remainder of this paper is organized as follows: Section II provides a brief overview of related work. Section III describes the proposed CL-FP approach for MAC spoofing detection in detail. Section IV presents our experimental methodology and Section V discusses the results. Section VI concludes the paper and presents future research opportunities.

II. Related Work

Device FP has been used as a way to secure wireless networks [9]. Existing approaches either use a set of manually selected features [8] or, more recently, features are extracted using deep learning [10]. However, the question of how to properly select and extract FP features is still an open research problem. For example, when used to identify devices, FP requires different features for different applications, such as security [8] [9] or localization [11].

Cross-layer FP (CL-FP) techniques using features from multiple layers have been proposed [9] [8] [10]. Overall, these approaches can be split into two categories based on how cross layer features are selected, namely: manual selection and deep learning-based cross layer feature extraction.

Approaches that fall in the first category manually identify relevant features at different layers of the protocol stack, including MAC-layer clock skew, MAC-layer frame inter-arrival times, RF features, network-layer packet inter-arrival times, and application-layer traffic patterns [9]. Most of these features are applied independently for different applications. For instance, MAC-layer clock skew is used as a device feature to detect MAC address spoofing. How to combine and select these cross-layer features is still an open question. Additionally, lightweight approaches to obtain physical layer features without expensive signal processing computation are crucial.

As discussed in [11], location-dependent cross-layer features can be generated from application-layer visual fingerprints, physical-layer motion and signal fingerprints, or a combination thereof. However, in [11], there is no discussion of a general principle or architecture to combine cross-domain features. In [8], Error Vector Magnitude (EVM) and MAC address are used to identify MAC spoofing attacks. However, the theoretical analysis in [8] is limited and further investigation is needed to provide solid reasoning for why the combination of EVM and MAC address meets the requirements of the driving application, in this case MAC address spoofing detection.

Examples of CL-FP approaches based on deep learning include [12] and [13]. They propose different deep learning approaches using raw physical-layer I/Q data to uniquely identify radio transmitters. Cross-layer approaches based on deep learning require vast amounts of data that need to be processed by expensive hardware consuming significant computation and energy resources.

III. CL-FP Approach

A. Overview

Our exploration of the CL-FP state-of-the-art calls for more efficient, adaptive and low-cost CL-FP solutions. As such, the overall goal of the proposed CL-FP framework is to fill this gap. Fig. 1 illustrates our CL-FP framework which aims at capturing FP features from different network layers to satisfy application requirements. Specifically, we show the different stages of our CL-FP pipeline considering MAC spoofing detection as the driving application. Each stage of our CL-FP pipeline is described below:

- *CL-FP Information Collection* collects information from end-user devices. It processes or decodes user device information in order to select, formulate, and extract specific device features. These features can be extracted from various network layers and analyzed later according to the driving application requirements, in this case MAC spoofing detection. This stage is the front-end of the CL-FP pipeline and can be hosted by different devices, including Access Points (APs) and Base Stations (BSs).
- *CL-FP Information Analysis* analyzes extracted CL-FP features and creates the CL-FP database. It then identifies, classifies, or clusters devices, e.g., using machine learning (ML) methods. In the case of MAC spoofing detection, the CL-FP Analysis stage detects spoofed MAC addresses.
- *CL-FP Application* makes decisions on how to handle end-user devices based on the driving application requirements, in this case, MAC spoofing detection, and using data from CL-FP Information Analysis. For example, upon detection of a device with spoofed MAC address, this device's access to the network may be denied or restricted.

B. Error Vector Magnitude (EVM) as Fingerprint

As previously discussed (see Section II), the EVM [14] has been used to fingerprint devices [8]. The EVM is defined as the root-mean-square (RMS) of the difference between the received symbols and the expected symbols [14]. The main benefits of using the EVM, a.k.a. Relative Constellation Error (RCE), as a device fingerprint feature include: (1) by quantifying the distortion between the ideal and the actual received signal, the EVM can be used to represent a transmitter's unique features; (2) Additionally, the EVM computation is relatively lightweight and is usually readily available in most signal analyzers.

In this section, we use the theoretical formulation presented in [15] to show that the EVM can be expressed as a unique function of a device's I/Q gain imbalance, which is independent of the modulation scheme. In other words, we demonstrate that, assuming constant Signal-to-Noise Ratio (SNR) and perfect transmitter phase imbalance[1], the EVM can be used as a proxy to represent a transmitter's I/Q gain imbalance, and thus can be used to fingerprint devices.

The theoretical background underpinning the relationship between the EVM and the I/Q gain imbalance is described in [15]. This model assumes a perfect demodulator and an I/Q phase imbalance that is relatively negligible. As previously noted, assuming constant SNR, if the EVM is normalized to

[1]In this work, we focus on the I/Q gain imbalance as a feature and assume that the I/Q phase imbalance is marginal. Incorporating the I/Q phase imbalance will be explored in our future work.

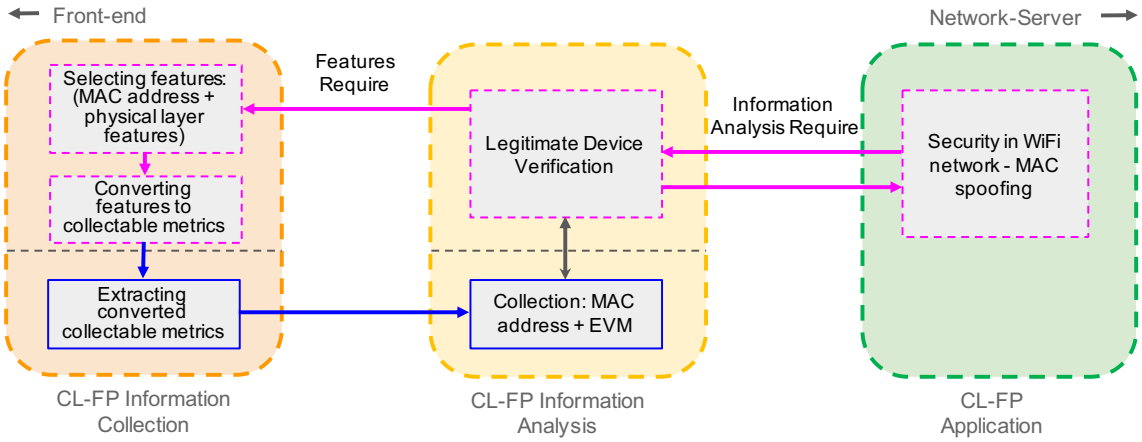

Fig. 1: CL-FP for MAC Spoofing Detection

the average power of the reference signal (1), the EVM relative to the I/Q gain imbalance is independent from the modulation order. On the other hand, if the EVM is normalized to the peak power of the reference signal (2), the EVM relative to the I/Q gain imbalance is dependent on peak-to-average energy ratio (PAV) which is dependent on modulation orders. As a result, for fingerprinting purposes, it is better to calculate the EVM using the average power normalization (1) as the resulting EVM will uniquely reflect the device's I/Q gain imbalance.

$$EVM_{rms,avg} = \sqrt{\frac{1}{SNR} + 2 - \left(\sqrt{2 + \frac{4g_t}{1+g_t^2}}\right)} \quad (1)$$

$$EVM_{rms,peak} = \sqrt{\frac{1}{SNR} + 2 - \left(\sqrt{2 + \frac{4g_t}{1+g_t^2}}\right)} \times \sqrt{\frac{1}{PAV}} \quad (2)$$

where,

1) $EVM_{rms,avg}$, $EVM_{rms,peak}$ are, respectively, the RMS of the error vectors computed and normalized to the average or peak symbol power of the EVM reference;
2) $SNR = \frac{E_s}{N_0}$, E_s is the average signal symbol energy, and N_0 represents the power spectral density (PSD) of white Gaussian noise;
3) g_t is the I/Q gain imbalance of the transmitter; and
4) $PAV = \frac{E_{peak}}{E_s}$, E_{peak} represents the peak symbol energy.

C. EVM as I/Q Imbalance Proxy

In order to validate the use of the EVM to represent a device's I/Q gain imbalance as established in Equations (1) and (2), we conducted extensive experiments using MATLAB's WLAN toolbox to simulate WLAN signals [16].

In our experiments, we generated random binary data and modulated it with randomly selected Quadrature Amplitude Modulation (QAM) orders. We used data streams of 60,000, 600,000, and 6 million bits. QAM modulation orders were set

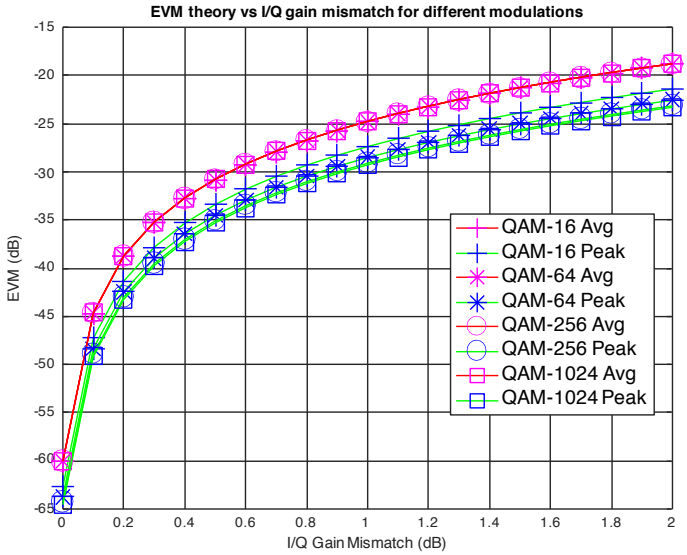

Fig. 2: Validating the EVM as I/Q Imbalance Proxy. Results have been obtained by averaging over 6 millions symbols and for an average $SNR = 60$ dB.

to M = 16, 64, 256, 1024. To simulate different hardware imperfections for different transmitter devices, various I/Q gain imbalance values were applied to the signals. We decoded the signals and calculated their EVM.

In our study, we evaluated the EVM theoretical calculation approach. Note that EVM theoretical calculation algorithm calculates the RMS EVM value defined in [16].

The EVM of the simulated devices is calculated theoretically. Our experimental results are presented in Fig. 2 where the EVM was computed using average normalization, as illustrated by the pink curves. Note that all pink curves are superimposed confirming that the relation between the EVM and the I/Q gain imbalance is unique and independent of the modulation type as established in Equation (1). If instead normalization by peak is used, the EVM versus I/Q gain imbalance (the blue curves) show that the relation between the

EVM and the I/Q gain imbalance is affected by the modulation type, determined by PAV value as established in Equation (2).

In summary, our experimental results confirm the theoretical models and demonstrate that, if we assume constant SNR, the EVM normalized by average signal power can be used to uniquely represent a device's I/Q gain imbalance, regardless of the signal modulation method.

IV. CL-FP Use Case - MAC Spoofing Detection

In this section, we describe how our CL-FP framework can be used for MAC spoofing detection. We start by presenting our experimental methodology, then describe our experimental setup and how we ran our experiments, and what performance metrics we use to evaluate our approach.

A. Experimental Methodology

Figure 3 shows our experimental setup for the MAC spoofing detection use case. In our experiments, we make the following assumptions: (1) Each device is tagged with a source MAC address, uses a random QAM modulation order and a given I/Q gain imbalance to represent the device's unique hardware imperfection. We assume that other hardware imperfections are negligible; (2) All devices are in the same SNR channel; and (3) The demodulators are ideal, i.e., they do not alter the transmitter's CL-FP features.

We carried out the experiments as follows. First, we built the reference database (DB) to store the source MAC address and EVM as CL-FP features for all legitimate devices. Each legitimate device was randomly assigned a source MAC address and a certain I/Q gain imbalance representing its unique hardware characteristics. Then, each device generated and transmitted its own Wi-Fi signals using MATLAB's WLAN toolbox [16]. These WiFi signals were captured and decoded at the receiver side to obtain the EVM which along with the device's MAC address were stored in DB. For each legitimate device, we obtained multiple samples of the EVM which are all stored in DB to represent that device's EVM. The reason to store multiple EVM samples for each legitimate device is to obtain more accurate distribution of the EVM values for improved MAC spoofing detection accuracy.

After building the DB with information from legitimate devices, a new set of devices including legitimate and illegitimate devices was generated. Legitimate devices are represented by blue triangles in Fig. 3, while orange triangles denote illegitimate devices. Each device generated and transmitted their own WiFi signals which were captured and decoded to obtain the corresponding EVM and MAC address as the device's CL-FP features. These CL-FP features were then analysed and compared with features stored in DB to detect potential MAC spoofing attacks. For example, Fig. 3 shows an illegitimate device, which was assigned a known source MAC address A, but different I/Q gain imbalance. Using the relation between I/Q gain imbalance and EVM, the EVM calculated for device with MAC address A was ZZ, which is different when compared to the EVM value X for the legitimate device A as stored in DB. This indicates that this device is not device A.

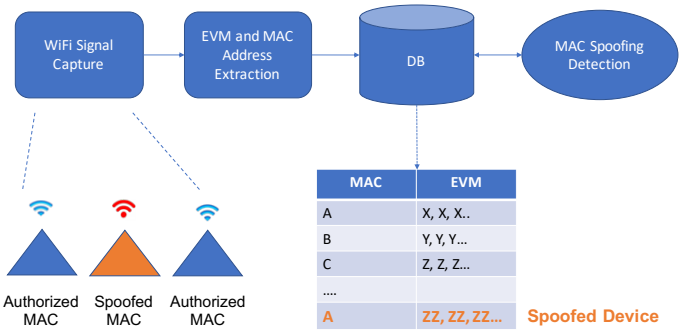

Fig. 3: MAC Spoofing Detection Case Study

Physical Device Emulation The setup we used to emulate physical devices (adapted from [17]) is shown in Fig. 4 and consists of three main components, namely:

- *Transmitter:* This component emulates individual transmitter devices which can be legitimate or illegitimate. Using MATLAB's WLAN Toolbox, each device was configured to generate WiFi signals with various modulation configurations, source MAC address and I/Q gain imbalance.
- *Channel:* This component represents the wireless channel between the transmitter device and the receiver. Gaussian channel was used;
- *Receiver:* At the receiver side, generated waveforms by the transmitter were captured and processed [16] and CL-FP features, i.e., the EVM and source MAC address, were extracted.

Building DB: The DB was built as follows:

- *Physical device emulation*: A transmitter device was created and assigned a random source MAC address and I/Q gain imbalance in the range [0,1], using a step difference of 0.05. In total, there were 20 different I/Q gain imbalance values. To emulate device modulation variations, three type of devices were used: Type1: same QAM modulation, same I/Q gain imbalance, and same source MAC address; Type 2: random QAM modulation, same I/Q gain imbalance, and same source MAC address; Type 3: random QAM modulation, same I/Q gain imbalance, and different source MAC address.
- *Physical device generation:* For each type of device, we generated 20 different I/Q gain imbalance values in the range [0,1] with 0.05 as step. This resulted in 1200 (20*20*3) different devices. Then we have randomly picked a unique combination of the I/Q gain imbalance and source MAC address as a unique device. 31 devices were selected.
- *CL-FP feature extraction:* Signals from the 31 devices chosen were decoded. Multiple samples of the EVM and source MAC address were obtained from decoded signals transmitted by each transmitter device. These features were stored in DB as reference. The number of the feature samples for each device is named as the number of reference CL-FP values, as shown in the x-axis of Fig. 5.

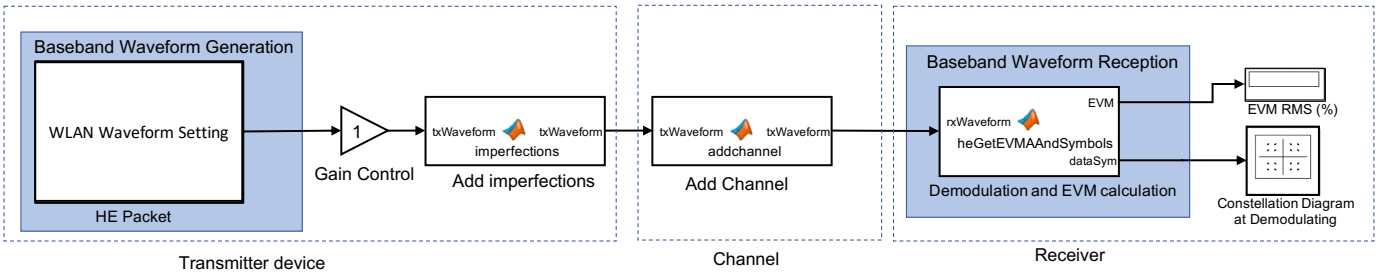

Fig. 4: Physical Device Emulation

Generating test devices:

- *Test device generation:* 28 transmitter devices were generated to test our CL-FP-based MAC spoofing detection system. 20 of the 28 devices are "illegitimate" devices with spoofed source MAC addresses; all other devices are legitimate. The 20 devices with spoofed MAC addresses assigned to these 28 transmitter devices were a subset of the MAC address registered in DB. The MAC spoofed devices were assigned different I/Q gain imbalance values. These values are 0.1, 0.05, 0.025, or 0.0125 different from the legitimate devices with the same source MAC address. This I/Q gain imbalance difference is referred to as *sensitivity* and determines how sensitive our detection mechanism is to the differences in I/Q gain mismatch. Legitimate devices were assigned the same I/Q gain imbalance values.
- *Extracting CL-FP features:* The waveforms of test devices were captured and their CL-FP features, including the EVM and the source MAC address, were extracted.

Detecting MAC spoofing: Algorithm 1, which employs a variation of the nearest neighbor approach, describes how we perform MAC spoofing detection, essentially, by comparing CL-FP features of test devices ($EVM_{testDevice_i}$) against the CL-FP features of the "whitelisted" devices (EVM_{Ref_j}) stored in DB. If a test device has the same source MAC address as a "whitelisted" device in DB, the distance between $EVM_{testDevice_i}$ of the test device and each EVM sample EVM_{Ref_j} of the whitelisted device is calculated. The probability p that these two devices are the same is based on whether the distance is less than or more than the distance threshold ϵ. We use a probability threshold η such that if p is less than η, the two devices are deemed different, and consequently a MAC spoofing event is detected.

B. Performance Metrics

We evaluate the proposed CL-FP based MAC spoofing detection system based on the True Positive Ratio (TPR), defined as: $TPR = \frac{TP}{TP+FN}$,

where:

- True Positive (TP): MAC spoofing is detected successfully;
- False Negative (FN): devices are falsely deemed legitimate;
- False Positive (FP): devices are falsely deemed illegitimate;
- True Negative (TN): devices are rightly deemed as legitimate.

Algorithm 1 MAC spoofing detection

Input: $EVM_{testDevice_i}$, EVM_{Ref_j}, ϵ, η
Output: $testDevice_i$ is MAC spoofing or not

for *each* $newDevice_i$ **do**
  if $MAC_{testDevice_i}$ in DB **then**
    $p_{sum} \leftarrow 0$
    for $j \leftarrow 1$ to N: EVM_{Ref_j} *of the device with* $MAC_{testDevice_i}$ *in DB* **do**
      $d \leftarrow |EVM_{testDevice_i} - EVM_{Ref_j}|$.
      if $d > \epsilon$ **then** ▷ ϵ *is the distance threshold*
        $p_j = 0$
      else
        $p_j = 1$
      end if
      $p_{sum} = p_{sum} + p_j$
    end for
    $p = p_{sum}/N$
    if $p < \eta$ **then** ▷ η *is the probability threshold*
      There is MAC spoofing
    else
      There is no MAC spoofing
    end if
  end if
end for

Table I shows the "Confusion Matrix", which summarizes the different outcomes of our MAC spoofing detection algorithm.

TABLE I: Confusion Matrix

	Actual Value (1, MAC spoofing)	Actual Value (0, no MAC spoofing)
Predicted Value (1, MAC spoofing)	TP	FP
Predicted Value (0, no MAC spoofing)	FN	TN

V. Experimental Results

The experimental results shown in Fig. 5 reveal the relationship between the performance metric TPR and the two main parameters used in our CL-FP based MAC spoofing detection approach, namely the *sensitivity* and *the number of reference CL-FP samples*. The x-axis, named number of reference CL-FPs, indicates how many samples of reference CL-FP features for each device have been stored in the DB.

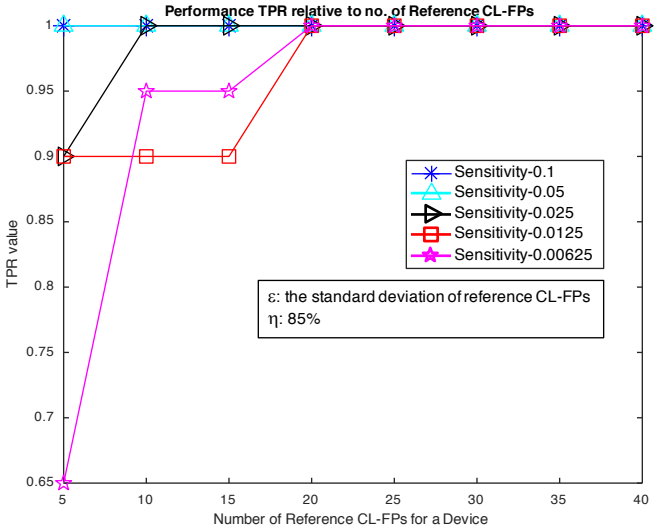

Fig. 5: Network Performance-TPR

The y-axis represents the TPR. The different curves in the graph show the performance of our approach as a function of the *sensitivity* parameter.

Fig. 5 shows the TPR when the *distance threshold* ε is set to the standard deviation of the distribution of the reference CL-FP samples stored in DB, and the *detection probability threshold* η is 85%[1]. As expected, when the *sensitivity* is higher than 0.05 (the arrow curve in blue), the TPR is 100% using relatively few CL-FP feature samples. When the *sensitivity* is lower than 0.05 and when there are not enough CL-FP samples in DB (less than 20 in our experiments), the TPR takes longer to converge as the EVMs of the test- and reference devices are very similar and thus harder to distinguish. As shown in the graph, the more samples of the the more accurate it is to distinguish legitimate from illegitimate devices using their EVMs.

Overall, our preliminary results are promising: they indicate that the proposed CL-FP based MAC spoofing detection approach exhibits adequate convergence with the number of CL-FP samples thus providing adequate accuracy (greater than 90% in the scenarios we tested) with relatively low overhead, i.e., relatively small number of CL-FP samples. In future work, we will expand our experiments to include a greater variety of emulated devices as well as explore different variations of our current MAC spoofing detection algorithm. We also plan to evaluate our CL-FP MAC spoofing detection system using an experimental testbed with real devices.

VI. Conclusion and Future Work

In this paper, we introduced a novel approach to perform MAC spoofing detection using cross-layer fingerprinting. The proposed approach uses the device's source MAC address and its EVM, extracted from its transmitted WiFi signals. Another contribution of our work is that we experimentally confirm that easier-to-extract FP features, such as the EVM, can reliably represent harder-to-extract intrinsic physical characteristics of devices. Results from our preliminary experiments confirm that the proposed MAC spoofing detection system can achieve adequate accuracy with relatively low overhead. As future work, we will deploy our system in a testbed with real devices. We will also explore applying our CL-FP approach for different applications (e.g., device localization, QoS provisioning) and network environments which will require selecting, formalizing and extracting different device features as their fingerprint.

[1]We have run our MAC spoofing detection algorithm for other combinations of these two parameters and notice similar trends. We omit these graphs here due to space limitations.

References

[1] S. U. Rehman, K. W. Sowerby, S. Alam, and I. Ardekani, "Radio frequency fingerprinting and its challenges," in *2014 IEEE Conference on Communications and Network Security*. IEEE, 2014, pp. 496–497.

[2] A. Jagannath, Z. Kane, and J. Jagannath, "Rf fingerprinting needs attention: Multi-task approach for real-world wifi and bluetooth," *arXiv preprint arXiv:2209.03142*, 2022.

[3] B. Danev, D. Zanetti, and S. Capkun, "On physical-layer identification of wireless devices," *ACM Computing Surveys (CSUR)*, vol. 45, no. 1, pp. 1–29, 2012.

[4] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*, 2008, pp. 116–127.

[5] X. Guo, Z. Zhang, and J. Chang, "Survey of mobile device authentication methods based on rf fingerprint," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019, pp. 1–6.

[6] S. Hanna, S. Karunaratne, and D. Cabric, "Deep learning approaches for open set wireless transmitter authorization," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2020, pp. 1–5.

[7] J. Bassey, X. Li, and L. Qian, "Device authentication codes based on rf fingerprinting using deep learning," *arXiv preprint arXiv:2004.08742*, 2020.

[8] G. An and S. H. Kim, "Mac spoofing attack detection based on evm in 802.11 wlan," *UBICOMM 2013 - 7th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, no. c, pp. 163–167, 2013.

[9] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device fingerprinting in wireless networks: Challenges and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 94–104, 2015.

[10] H. Jafari, O. Omotere, D. Adesina, H.-H. Wu, and L. Qian, "Iot devices fingerprinting using deep learning," in *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*. IEEE, 2018, pp. 1–9.

[11] Q. D. Vo and P. De, "A survey of fingerprint-based outdoor localization," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 491–506, 2015.

[12] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, "Oracle: Optimized radio classification through convolutional neural networks," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 370–378.

[13] S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury, "Deep learning convolutional neural networks for radio identification," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 146–152, 2018.

[14] M. D. McKinley, K. A. Remley, M. Myslinski, J. S. Kenney, D. Schreurs, and B. Nauwelaers, "Evm calculation for broadband modulated signals," in *64th ARFTG Conf. Dig.* Orlando, 2004, pp. 45–52.

[15] A. Georgiadis, "Gain, phase imbalance, and phase noise effects on error vector magnitude," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 2, pp. 443–449, 2004.

[16] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A tutorial on ieee 802.11 ax high efficiency wlans," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 197–216, 2018.

[17] "Modeling and testing an 802.11ax RF transmitter - MATLAB & Simulink," https://www.mathworks.com/help/wlan/ug/modeling-and-testing-an-802-11ax-rf-transmitter.html.