2022

A Channel State Information Based Virtual MAC Spoofing Detector

Peng Jiang
Old Dominion University, pjiang@odu.edu

Hongyi Wu
Old Dominion University, h1wu@odu.edu

Chunsheng Xin
Old Dominion University, cxin@odu.edu

Original Publication Citation

Jiang, P., Wu, H., & Xin, C. (2022). A channel state information based virtual MAC spoofing detector. *High-Confidence Computing*, 2(3), 1-6, Article 100067. https://doi.org/10.1016/j.hcc.2022.100067

Contents lists available at ScienceDirect

High-Confidence Computing

homepage: www.elsevier.com/locate/hcc

A channel state information based virtual MAC spoofing detector☆

Peng Jiang[a,b], Hongyi Wu[a,b], Chunsheng Xin[a,b,*]

[a] Department of ECE, Old Dominion University, Norfolk, VA, USA
[b] School of Cybersecurity, Old Dominion University, Norfolk, VA, USA

ARTICLE INFO

2022 MSC:
00-01
99-00

Keywords:
Virtual Mac spoofing
Channel state information
Deep learning

ABSTRACT

Physical layer security has attracted lots of attention with the expansion of wireless devices to the edge networks in recent years. Due to limited authentication mechanisms, MAC spoofing attack, also known as the identity attack, threatens wireless systems. In this paper, we study a new type of MAC spoofing attack, the virtual MAC spoofing attack, in a tight environment with strong spatial similarities, which can create multiple counterfeits entities powered by the virtualization technologies to interrupt regular services. We develop a system to effectively detect such virtual MAC spoofing attacks via the deep learning method as a countermeasure. A deep convolutional neural network is constructed to analyze signal level information extracted from Channel State Information (CSI) between the communication peers to provide additional authentication protection at the physical layer. A significant merit of the proposed detection system is that this system can distinguish two different devices even at the same location, which was not well addressed by the existing approaches. Our extensive experimental results demonstrate the effectiveness of the system with an average detection accuracy of 95%, even when devices are co-located.

1. Introduction

In the past decade, the ubiquitous expansion of smart home applications allow a mass of IoT devices to be connected to the Internet. The majority of devices use WiFi as the primary way for communications. Most WiFi networks nowadays use *WiFi Protected Access 2* (WPA2) or IEEE 802.11i to protect users' security. Nevertheless, such prior authentication is vulnerable to physical layer spoofing attacks in which an inside spoofer can claim to be another node by using the Media Access Control (MAC) address of the latter. Attackers can easily obtain other users physical layer information by using the pervasive public tools on the 802.11 commodity hardware [1] or simply sniffing the ARP packets in the network, which makes it feasible for ordinary users to alter the device's hardware-level parameters in wireless networks, as well as launching various identity theft attacks.

Launching identity attacks allows the attacker to obtain an illegal advantage in the Man-in-the-Middle (MITM) attacks or Denial of Service (DoS) attacks while the system admin cannot identify the real owner of the corresponding MAC address. When multiple devices in the local network use the same MAC address, the collision of the MAC address will cause all devices sharing the same MAC address to be denied from the regular services. Fig. 1 illustrates the traditional MAC spoofing attack in a local wireless network, in which an attacker, Eve, forges its MAC address to masquerade as another benign node, Bob. Then Eve can deceive the Access Point (AP), disrupt the regular network connections of Bob, or advertise false services to nearby mobile stations. However, due to the collision of the MAC address with Bob, Eve cannot maintain a good connection with AP, either.

There are several physical layer proprieties that can be utilized for fingerprinting devices, which can be further used to detect MAC spoofing attacks, i.e., traditional power features and finer-grained channel response. Traditional power features include Received Signal Strengths (RSS) and Received Signal Strength Indicators (RSSI), and finer-grained channel response include Channel State Information (CSI). RSSI was considered in many works in physical layer authentication [2–4] as it reveals the attenuation of radio signal during the propagation. However, RSS and RSSI are incapable of providing robust and stable signal features in complex indoor environments due to multipath fading [5]. Benefiting from the adoption of Orthogonal Frequency Division Multiplexing (OFDM) technology since IEEE 802.11n, channel response can be extracted from the off-the-shelf WiFi receivers to indicate the amplitudes and phases of every subcarrier between the communication peers. Therefore, CSI leverages the finer-grained power feature to discriminate multipath characteristics.

In [6], the authors proposed a CSI-based approach called Profile Matching Authenticator (ProMA), to detect traditional MAC spoofing attacks. It utilizes the CSI amplitude information to build a profile for each legitimate device in the network and then detect MAC spoofing by looking at these profiles. This method relies on the fact that the ampli-

☆ A preliminary version of this work was published in IEEE ICC 2018.
* Corresponding author.
E-mail address: cxin@odu.edu (C. Xin).

https://doi.org/10.1016/j.hcc.2022.100067
Received 25 March 2022; Accepted 17 May 2022

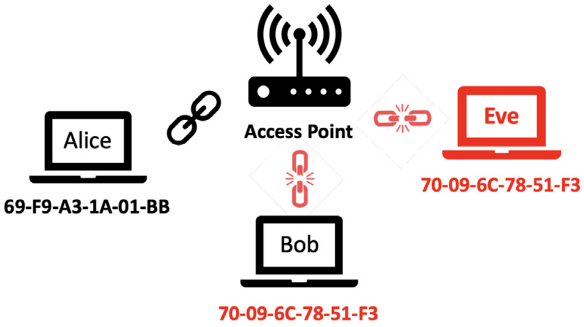


Fig. 1. Illustration of the traditional MAC spoofing attack.

tudes of CSI from devices at different positions are different. Hence, the attackers who forge another node's MAC at a different location can be detected since the CSI amplitude does not match the profile constructed for any legitimate user. Nevertheless, the CSI amplitudes of two devices at the same location are generally similar. Hence this approach has difficulty distinguishing two legitimate devices at the same location, i.e., one of them would be falsely alarmed as a MAC spoofing attacker and vice versa. Moreover, the CSI amplitude from the same location may vary significantly over time.

In addition to the issues discussed above, those existing methods for MAC spoofing detection face even more challenges for a new type of MAC spoofing attack termed *virtual MAC spoofing attack* in this paper. Recently, the development of network virtualization techniques [7–9], especially the MAC layer virtualization [10], enables the virtual MAC spoofing attack. With this attack, an attacker can launch massive MAC flooding attacks [8] to disrupt a large-scale network to cause a severe denial of service while the attacker can still maintain the connection to the access point. For example, as shown in Fig. 2, Eve is the attacker who creates two virtualized interfaces to forge the MAC addresses of Alice and Bob simultaneously. Although Alice and Bob are suffering from the potential packet loss due to the MAC collision, Eve can still use its unique MAC address to maintain the regular connection. Virtual MAC technology also enables malicious users to demultiplex the network traffic by dynamically scheduling packets to be transmitted on different virtualized interfaces to reshape the original traffic, which can conceal malicious network activities and evade the intrusion detection system [10,11]. Therefore, it is rather challenging to detect virtual MAC spoofing using the existing methods.

In this paper, we propose a virtual MAC spoofing detection scheme termed *Virtual MAC Spoofing deteCtor* (VMASC), utilizing the *Channel State Information* (CSI) and the deep learning technique. VMASC can effectively detect both MAC and virtual MAC spoofing attacks. Specifically, based on the amplitude and phase information extracted from CSI, VMASC automatically extracts features to classify devices through a trained *Convolutional Neural Networks* (CNN). VMASC has several merits. First, it can distinguish the devices even at the same location through using both the amplitude and the phase information of CSI. Second, VMASC is very robust and works well even under severe communication environments. It also does not need any a priori information of devices, such as collecting the profiles of devices in previous studies. Finally, VMASC does not need to use an expensive and high-resolution signal processing analyzer. It only uses off-the-shelf devices. The proposed VMASC achieves an average of 95% accuracy in various environments. It can be used for practical applications such as being added into the administrator's toolbox of a wireless network to effectively detect MAC spoofers.

The rest of the paper is organized as follows. Section 2 introduces our approach to distinguish co-locating devices, a key challenge in detect of MAC spoofers. Section 3 describes the system architecture of VMASC. Section 4 presents performance evaluations and Section 5 concludes the paper.

2. Distinguish co-locating devices

The biggest challenge in MAC spoofing detection is how to distinguish devices that are at the same location. VMASC exploits two measurements of CSI to achieve this objective: 1) both the amplitude and the phase information in CSI, and 2) the CSI error or variance of NICs. Next we briefly introduce these two measurements.

The CSI is a fine-grained channel information that can be often provided by NICs. In the rest of the paper, for the ease of description, we assume the wireless network is a WiFi network using the OFDM modulation. However, our approach is applicable to other wireless networks where the CSI can be obtained.

With the OFDM modulation, the data stream is encoded on multiple orthogonal subcarriers over the entire spectrum band. For example, in 802.11n 20Mhz non-High Throughput mode, each WiFi channel contains 56 subcarriers, and in 802.11n 40Mhz High Throughput mode, each WiFi channel contains 114 subcarriers. The quantified channel frequency response for each subcarrier between each transmitter antenna and each received antenna can be obtained by the off-the-shelf NICs such as Atheros 9462 [12]. The frequency domain response in the OFDM system can be described as follows.

$$\mathbf{Y}_s = C_s\mathbf{X}_s + \mathbf{N}_s, \tag{1}$$

where $\mathbf{Y}_s$ and $\mathbf{X}_s$ represent the received and the transmitted signal vectors on subcarrier s, respectively, C_s is the CSI on subcarrier s, and $\mathbf{N}_s$ is the noise vector on subcarrier s. When MIMO communication (N_T number of transmit antenna and N_R number of receive antenna) is engaged, C_s is an $N_T \times N_R$ matrix representing the channel frequency information on a subcarrier. For example, the Atheros NIC supports

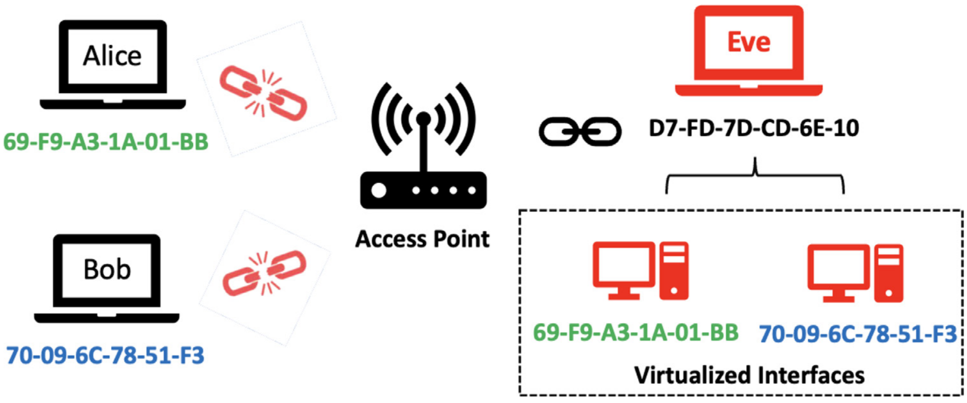
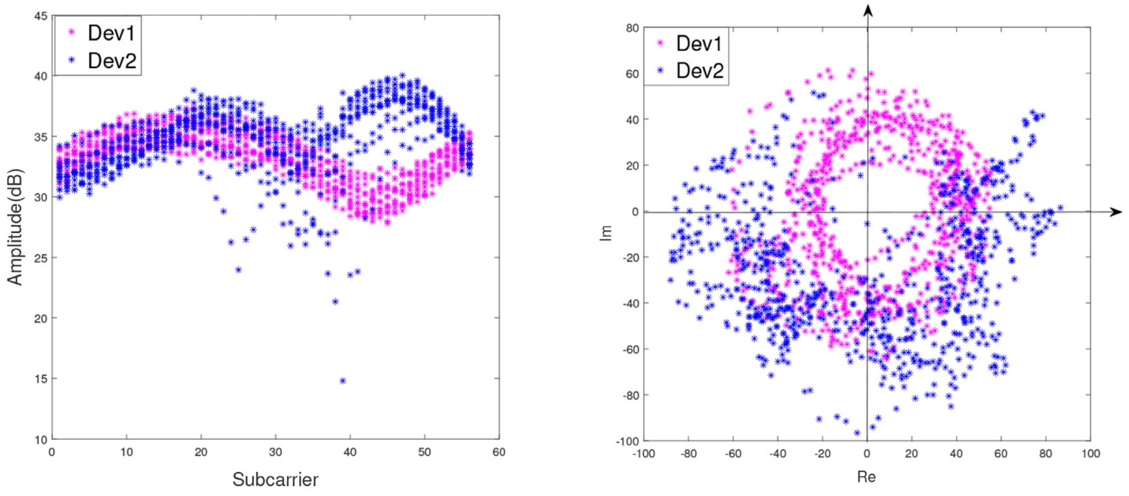


Fig. 2. Illustration of the virtual MAC spoofing attack.



(a) Amplitude (b) Amplitude and phase

Fig. 3. CSI of 20 packet transmissions from two devices at the same location, a) amplitude, b) amplitude and phase.

802.11n with 20Mhz/40Mhz bands at the 2.4Ghz/5Ghz frequency and is equipped with two antennas on each NIC to collect CSI for each transmitted packet. Let N_S represent the number of subcarriers in OFDM. It is dynamically adjusted by the NIC between 56 and 114 based on the condition of the wireless environment. The CSI on each subcarrier is a complex number and it refers to the channel properties such as the channel frequency response of a communication frequency band. Moreover, each CSI value consists of both the amplitude response and the phase response on each subcarrier. Thus, it can be defined as

$$\dot{C}_{(n_r,n_t,n_s)} = \left|C_{(n_r,n_t,n_s)}\right| \exp\left\{ j\angle C_{(n_r,n_t,n_s} \Big) \right\}, \tag{2}$$

where n_r indicates the r-th receive antenna, n_t indicates the t-th transmit antenna, and n_s indicates the s-th subcarrier. $\left|C_{(n_r,n_t,n_s)}\right|$ is the amplitude response and $\angle C_{(n_r,n_t,n_s)}$ is the phase response of a subcarrier, respectively.

To illustrate our motivation to use both the amplitude and the phase of CSI to distinguish co-locating devices, instead of simply using the amplitude of CSI, we plot the CSI amplitude and the full CSI on 56 subcarriers in Fig. 3 for 20 packets transmissions from two co-locating devices. Fig. 3(a) shows that the amplitude of CSI samples from two devices, Dev1 and Dev2, at the same location. We can clearly see that with both the amplitude and the phase, it is easier to distinguish the two devices, e.g., the CSI samples of Dev 2 (blue dots) are predominantly in the lower half of the complex plane in Fig. 3(b), while the CSI samples of Dev1 (red dots) are more evenly distributed in both the lower and the upper half of the complex plane. Relying solely on the amplitude can have difficulty to distinguish the two devices as the CSI amplitudes of the two devices are similar in most subcarriers. Furthermore, the benefit of using both the amplitude and the phase is that the phase information extracted from the CSI is more sensitive to reveal the variations of hardware imperfections, due to the non-linear errors introduced [13].

During the manufacturing process of the NICs, there are always imperfections introduced such that two NICs are not exactly the same in terms of signal transmission [13]. In other words, even the CSIs of the packet transmissions of two NICs of the same type at the same location, same time, and same frequency exhibit some variance. Similarly, in [13], the authors conduct an extensive analysis on the characteristics of the nonlinear errors on the channel frequency response introduced by the hardware imperfections. Such variance is caused by imperfections of the power amplifier, carrier frequency alignment, sampling frequency alignment, undesirable packet detection offset, and phase-locked loop offset [13]. Denote $\delta_{s,l}$ as the error introduced by device l on subcarrier s. Then, the actual CSI on subcarrier s is,

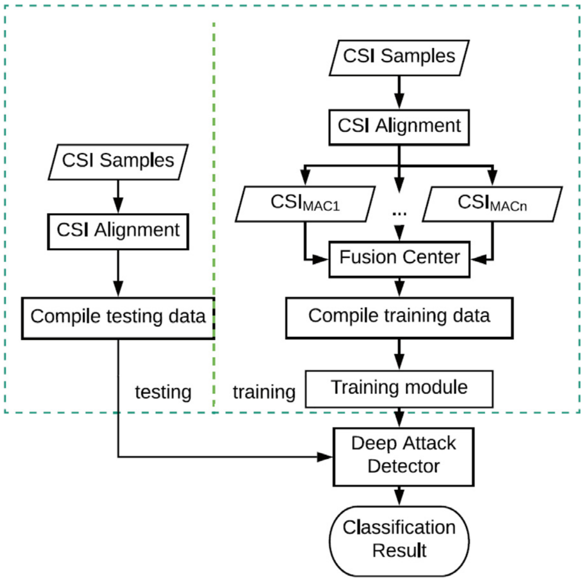


Fig. 4. VMASC architecture.

$$\hat{C}_{s,p} = C_{s,p} + \delta_{s,l}. \tag{3}$$

where $C_{s,p}$ is the theoretical CSI for the p-th packet of device l on s-th subcarrier. We can rewrite (3) as follows:

$$\hat{C}_{s,p} = \left|C_{s,p} + \delta_{s,l}\right| exp\left\{ j\angle(C_{s,p} + \delta_{s,l})\right\} \tag{4}$$

Such device-related CSI errors or variance due to imperfections of NICs is utilized by VMASC to identify the uniqueness of hardware and further differentiate hardware.

In the next section, we present a deep convolutional neural network that exploits the CSI errors caused by hardware imperfections, and the full CSI information extracted from the packet transmissions, to detect the virtual MAC spoofing attack.

3. Deep virtual MAC spoofing detector (VMASC)

The architecture of *VMASC* is shown in Fig. 4. We assume all devices are equipped with an off-the-shelf Atheros NIC, which can read CSI data.

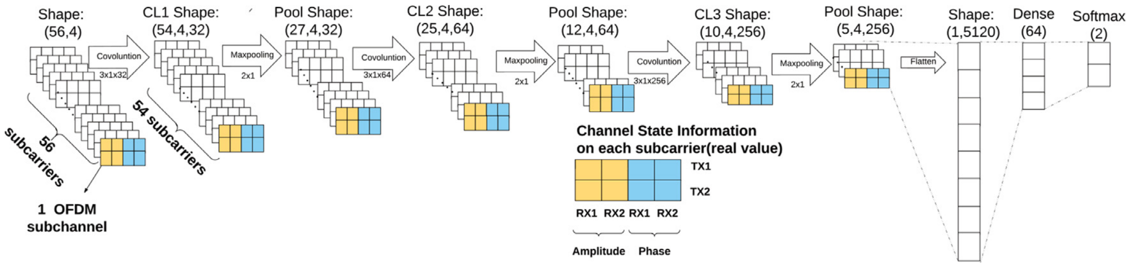


Fig. 5. The CNN structure used in VMASC.

The main idea of this design is to examine the CSI from two packets and use a deep learning classifier as the detector to identify whether the MAC addresses associated with the two packets originated from the same physical device.

3.1. Training data gathering process

To collect training data, CSI samples from all incoming packets are recorded at the AP (admin). However, the CSI extracted for each packet does not contain any upper layer information such as the source/destination MAC or IP address associated with its originated device, thus we need to align all the collected CSI with the traffic log and label each CSI with a right device label before compiling the training data.

Once we have collected the aligned CSI information, we compile the training data as follows. We first subtract the CSI from any two consecutive data packets (e.g., packet j and $j+1$), and record this CSI subtraction result and a corresponding label as one training data sample. The label value is assigned as follows. If the two consecutive packets are actually sent from the same device, the label is 0; otherwise, the label is 1. Let $\hat{\mathcal{X}}_{s,j}$ denote the CSI subtraction result on the sth subcarrier of the jth data in the training dataset, which can be expressed as

$$\hat{\mathcal{X}}_{s,j} = \hat{C}_{s,j} - \hat{C}_{s,j+1} = (C_{s,j} - C_{s,j+1}) + (\delta_{s,l_j} - \delta_{s,l_{j+1}}) \tag{5}$$

where $j \in [1, N-1]$, N is the total number of packets collected for training, l_j denotes the device that sends packet j, and l_{j+1} denotes the device that sends packet $j+1$, respectively.

Based on the spatial property and hardware difference, we have three scenarios as follows.

1. If packets j and $j+1$ are sent from the same device, the spatial property and hardware characteristic should be stationary within the period of the short inter-packet interval. Thus we have $C_{s,j} \approxeq C_{s,j+1}$ and $\delta_{s,l_j} \approxeq \delta_{s,l_{j+1}}$ based on the uniqueness and imperfection features mentioned in the preceding section.
2. If packets j and $j+1$ are sent from two devices at two different locations, the spatial property and hardware characteristics should be different. Thus we should have $C_{s,j} \neq C_{s,j+1}$ and $\delta_{s,l_j} \neq \delta_{s,l_{j+1}}$.
3. If packets j and $j+1$ are sent from two devices at the same location or at very close locations, the spatial property between two packets should be similar but the hardware imperfections are still different. We should have $C_{s,j} \approxeq C_{s,j+1}$ and $\delta_{s,l_j} \neq \delta_{s,l_{j+1}}$.

These scenarios indicate that,

$$\mathcal{X}_{s,j}(\text{Scenario 1}) \ll \mathcal{X}_{s,j}(\text{Scenario 3}) < \mathcal{X}_{s,j}(\text{Scenario 2}) \tag{6}$$

Noted that Eq. (6) shows that even in the extreme scenario where the attacker is co-located with a benign user in the network, VMASC can still detect it.

3.2. Attack detection

VMASC uses the *convolutional neural network* (CNN) to classify if a data sample (the CSI difference of two packets) is from two devices or the same device. CNNs have been proven to be very successful in computer vision [14–16]. By concatenating several convolutional layers, CNNs can extract complex features from data. In contrast to statistical learning which requires hand-crafted features, convolutional layers are automatic feature extractors so the neural network can learn a high-level abstraction from data.

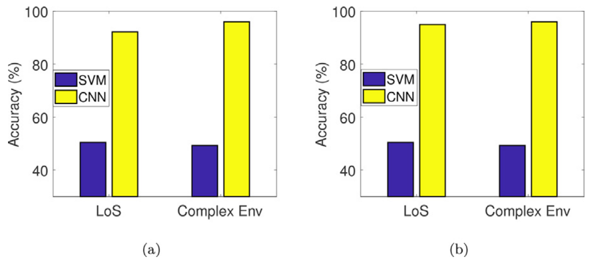
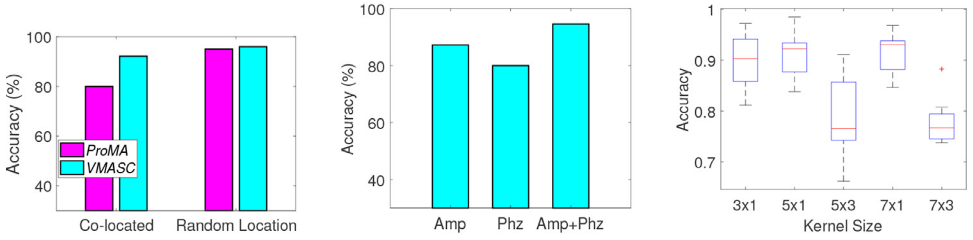


Fig. 6. Comparison of detection accuracy, a) MAC spoofing b) virtual MAC spoofing.

Here, packets from a unique device are identified by the CSI subcarrier information. Due to the dynamics of the wireless channel, these signatures also exhibit high temporal volatility that makes any predetermined feature futile. To extract subtle features from CSI on the packets, CNN is an ideal candidate. The original designs for image classification adopt 2D filters on the image to extract spatial features such as edges, contour, and angle. Here, the CSI input data has a size of $4 \times 2 \times 56$, where 56 is the number of OFDM subcarriers adopted during the transmission, 4 is the four pairs of transmitting and receiving antennas, and 2 indicates the two components of the CSI, i.e., the amplitude and the phase values, as illustrated in Fig. 5. To extend CNN for our purpose, 1D filters are used. For example, 3×1 filter will convolve on three data points each time along with the frequency domain (subcarrier) of each transmit and receive antenna pair to explore the unique hardware imperfection. We also verify that using 2D filters would lead to the performance degradation to be explained in Section 4.3.

Moreover, VMASC needs to use 1D filter to convolve on both the amplitude and phase information. The amplitude and phase extracted from each CSI sample depict the attenuation and the propagation delay of the signal that travels through a multipath environment. Since each of them only reveals partial traits of hardware difference introduced in the channel frequency response, we have to construct the training dataset based on both the amplitude and the phase to avoid the performance degradation. We have conducted extensive experiments to evaluate the performance when only the amplitude or the phase is used, as described in Section 4.3.

VMASC also uses a max pooling layer after some convolutional layers to reduce the dimensions to facilitate the learning process. Note that convolutions over CSI received from multiple packets in the time domain are not feasible as the packets could come from different devices in a mixed pattern. Convolution over such information from different devices does not contribute to the learning process.



(a) (b) (c)

Fig. 7. a) Comparison of detection accuracy between ProMA and VMASC, b) comparison of detection accuracy between different input features, “Amp” means CSI amplitude, “Phz” means CSI phase, c) CNN performance with different filter sizes to detect virtual MAC spoofing.

We propose several CNN architectures. For brevity, a 4-layer CNN is denoted as Conv(32)-MaxPooling-Conv(64)-MaxPooling-Conv(256)-Dense(64)-Softmax, in which there are three Conv layers with 32, 64, and 256 filters respectively and one densely connected layer with 64 activations. We choose 3 Conv layers because it can achieve a higher accuracy than the network with 2 Conv layers and consume less computation resources than 4 Conv network (Table 1 in Section 4.3.2), thus balance between the detection accuracy and resource consumption. The final softmax function projects the multi-dimensional output with arbitrary values into probabilities. If the final output is 0, two packets are sent from the same physical device, and vice versa. From the results, the attacker will be detected if any two packets with different hardware signatures claim the same MAC address (traditional MAC spoofing attack) or multiple packets with different MAC addresses are identified from the same device (virtual MAC spoofing attack).

4. Performance evaluation

In this section, we present the performance evaluation in different experiment environments.

4.1. Experimental setup

Our system has 3 major components, AP, hosted by a Dell Workstation; two Dell Laptops, with one as the benign node and the other as the attacker (already infiltrated and authenticated). All three devices are equipped with Atheros NICs with a modified driver to catalog CSI values. The transmitter and receiver both have two antennas. The AP will report the CSI value for each incoming packet associated with a MAC address. To implement the virtual MAC address spoofing attack, the attacker’s device will create multiple virtual MAC addresses.

We have conducted experiments in various scenarios. The first scenario is a strong Line-of-Sight (LoS) environment and another scenario is a complex environment with obstacles between devices. In LoS, all mobile devices and the attacker reside in an empty room and send data to an access point at their own speeds. The distance between the access point and the devices are within 1 m. The purpose of this setting is to mitigate the noise due to the multi-paths of the signal transmission and CSI variation caused by the spatial diversity from different locations. We arbitrarily selected 20 positions in the room for testing. To make the experiments more practical, we also deployed the system in a complex lab with obstacles, where devices are randomly distributed in a 10 × 5 m^2 area. The obstacles would block most of LoS paths and form a complex radio propagation environment. One position is selected for training and 30 positions are randomly selected for testing. The training data consists of 20,000 CSI samples and the testing data has 10,000 samples. The deep learning framework is implemented on Tensorflow [17]. We develop a 4-layer CNN structure of Conv(32)-MaxPooling-Conv(64)-MaxPooling-Conv(256)-MaxPooling-Dense(64)-Softmax, where the number of filters increases as the neural network gets deeper. The evaluation is based on this structure and we leave the development of an optimal structure to the future work.

4.2. Data collection

We collect the training and testing data from arbitrary locations in each experiment scenario for 10 minutes. When the system is under the traditional MAC spoofing attack, we assume that the network administrator does not have the knowledge of the attackers’ physical layer information since the attackers are not present during the training process (administrators can always conduct training in a controllable setting). Thus, the training dataset is composed of any two consecutive data packets from the benign users (exclude attackers). For virtual MAC spoofing, an attacker can either create virtual network interfaces for “benign” purposes as camouflage or spoof other users’ MAC to conduct malicious activities. Thus, the training data for virtual MAC spoofing is composed of any two consecutive data packets from all the devices (including the attacker).

4.3. Experimental results

4.3.1. Detection performance

We first evaluate the performance under two scenarios in terms of the detection rate, which is defined as the total number of correctly labeled two-packet tuples divided by the total number of two-packet tuples. The detection ratio of our proposed system is over 90% against both traditional and virtual MAC address spoofing attacks. We first compare the performance of VMASC using CNN versus the one using the traditional learning technique, Support Vector Machine (SVM). We also compare the performance of VMASC with the ProMA approach in [6].

Comparison with SVM: Fig. 7 shows the overall performance of the detection accuracy when using SVM and CNN, respectively, for VMASC, in two different attacking scenarios and different environments (i.e., an empty room, and a complex lab, respectively). In Fig. 7, we can easily observe that the average testing accuracy using the SVM classifier is low (about 50%) and the CNN reaches very high classification accuracy (around 95%). It demonstrates the power of CNN as an effective feature extractor compared to SVM even on CSI. Further, under the same attacking scenario, the accuracy of CNN in the LoS environment is slightly higher than the accuracy obtained under the complex environment. Such performance degradation is caused by the rapid change of spatial diversity on each subcarrier with the deflection and multiple paths of the transmitted signal. Moreover, when we compare the CNN result between different attacking scenarios, it is obvious that the overall detection accuracy for a system with more information is relatively higher compared to the one for a system with limited information.

Comparison with ProMA: We compare the performance of VMASC with the one of ProMA in [6] in Fig.. Under the co-located environment,

Table 1
Detection accuracy vs. time cost with different number of CNN layers.

Core Layer	Average Accuracy	Time for 10 epochs
1×Cov	0.83	26.63s
2×Cov	0.87	53.14s
3×Cov	**0.94**	113.14s
4×Cov	0.96	240.964s

ProMA results in a lower accuracy due to the spatial similarity between the attacker and benign users. When testing from randomly selected locations, both methods have good performance but VMASC beats ProMA in all the scenarios.

Comparison with Different Features: We also compare the average accuracy with different feature inputs in Fig. based on the same randomly selected data sets which cover all possible locations of the attacker. In the figure, “Amp” denotes amplitude and “Phz” denotes phase. We evaluate the performance when the CSI amplitude, phase, or both are used as CNN input. From the results we can tell that the performance of the CSI amplitude based detector can reach a higher accuracy than the purely phase based detector. This performance degradation is caused by the limited range of features when only the phase is involved. The range of each phase is within $[-\pi, \pi]$ which is not as wide as the range of amplitude on each subcarrier. However, once both features are combined together, the phase information can better help to identify the variance belonging to each device’s hardware.

4.3.2. Fine-tuning the CNN structure

We further examine the system performance by varying the CNN structure with a different filter size and number of CNN layers. Fig. shows the results of detection accuracy over all the testing samples. We observe that the best accuracy is achieved with filter sizes of 5×1 and 7×1. The choice of filter size is important. A small filter size (3×1) cannot capture features across more CSI subcarriers whereas a large filter size would unwittingly introduce more fluctuations/noise into a single convolution. Here, the results show that both 1D filters of sizes 5, or 7 achieve comparable performance. Fig. also illustrates the performance when 2D filters are used (filters 5×3 and 7×3). We can see that those results are severely degraded compared to the accuracy using 1D filters. Unlike image classification, where pixels have spatial correlations in a 2D plane, the secondary dimension in CSI data would bring irrelevant information into the feature extraction process and cause significant reeducation of accuracy for over 10%. Moreover, Table 1 compares the detection accuracy vs. time cost with a different number of convolution layers. From the table we can tell that with the increase of convolution layers in the neural network, the performance will be increased. However, the trade-off is obvious, resource consumption. With the same computation capability, the system with 4 convolution layers takes 240s to process 1 sample data whereas the system with 3 convolution layer just needs half of the time while achieving a comparable accuracy. Thus, to balance resource consumption and overall performance, we adopt the neural network with 3 convolution layers and 5×1 filter in the experiments.

5. Conclusion

In this paper, we have proposed a Channel State Information (CSI) based virtual MAC spoofing detector using the deep learning technology. Compared with existing approaches, VMASC can effectively distinguish two devices at the same location, using both the amplitude and phase information of CSI, and the CSI errors or variations caused by imperfections of the hardware. Extensive experiments have been conducted in various environments and the results demonstrate the effectiveness and robustness of VMASC compared with a previous approach.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] G. Joshua Wright, C. Joshua, Detecting wireless LAN MAC address spoofing, Cisco Certified Network Associate (2003).
[2] J. Yang, Y. Chen, W. Trappe, J. Cheng, Detection and localization of multiple spoofing attackers in wireless networks, IEEE Trans. Parallel Distrib. Syst. 24 (1) (2012) 44–58.
[3] L. Xiao, Y. Li, G. Han, G. Liu, W. Zhuang, PHY-layer spoofing detection with reinforcement learning in wireless networks, IEEE Trans. Veh. Technol. 65 (12) (2016) 10037–10047.
[4] A. Kalamandeen, A. Scannell, E. de Lara, A. Sheth, A. LaMarca, Ensemble: cooperative proximity-based authentication, in: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, 2010, pp. 331–344.
[5] Z. Yang, Z. Zhou, Y. Liu, From RSSI to CSI: indoor localization via channel response, ACM Computing Surveys (CSUR) 46 (2) (2013) 1–32.
[6] H. Liu, Y. Wang, J. Liu, J. Yang, Y. Chen, Practical user authentication leveraging channel state information CSI, in: Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security, 2014, pp. 389–400.
[7] T.J. Kuik, M.A. Bakke, Virtual MAC Address System and Method, 2008, US Patent 7,415,535.
[8] R.L. Bull, J.N. Matthews, Critical analysis of layer 2 network security in virtualised environments, in International Journal of Communication Networks and Distributed Systems 17 (3) (2016) 315–333.
[9] C. Liang, F.R. Yu, Wireless network virtualization: a survey, some research issues and challenges, IEEE Communications Surveys & Tutorials 17 (1) (2014) 358–380.
[10] F. Zhang, W. He, Y. Chen, Z. Li, X. Wang, S. Chen, X. Liu, Thwarting Wi-Fi side-channel analysis through traffic demultiplexing, in IEEE Transactions on Wireless Communications 13 (1) (2014) 86–98.
[11] F. Zhang, W. He, X. Liu, Defending against traffic analysis in wireless networks through traffic reshaping, in: Proceedings of the IEEE International Conference on Distributed Computing Systems, 2011, pp. 593–602.
[12] Y. Xie, Z. Li, M. Li, Precise power delay profiling with commodity Wi-Fi, IEEE Trans. Mob. Comput. 18 (6) (2018) 1342–1355.
[13] Y. Zhuo, H. Zhu, H. Xue, S. Chang, Perceiving accurate CSI phases with commodity WiFi devices, in: Proceedings of the IEEE Conference on Computer Communications, 2017, pp. 1–9.
[14] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv Neural Inf Process Syst 25 (2012).
[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions , in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[17] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: A system for large-scale machine learning, in: Proceedings of the ACM SIGPLAN International Conference on Functional Programming, 2016. 1–1