2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W) | 978-1-6654-3950-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/DSN-W52860.2021.00024

Authenticating Mobile Wireless Device Through Per-packet Channel State Information

Bing Chen*†‡§, Yubo Song*‡§, Zhenchao Zhu*‡§, Shang Gao¶, Junbo Wang†§, Aiqun Hu†§
*School of Cyber Science and Engineering, Southeast University
†School of Information Science and Engineering, Southeast University
‡Key Laboratory of Computer Network Technology of Jiangsu Province
§Purple Mountain Laboratories
¶Department of Computing, The Hong Kong Polytechnic University
chenbing9961@163.com, songyubo@seu.edu.cn, zhuzc@seu.edu.cn, shanggao@comp.polyu.edu.hk
jbwang@seu.edu.cn, aqhu@seu.edu.cn

***Abstract*—Non-cryptographic mobile wireless device authentication based on channel signature has aroused extensive attention. This technology uses the mobile device's physical characteristics to mark and verify its identity, and can be used to detect impersonation attacks and information forgery attacks. Channel State Information (CSI) has been used to generate fine-grained channel signatures. However, there are two sticking points in using CSI-based signature for authentication in association phase. The first is that the channel state will change as the device moves, which means that the local authenticator should be updated in real time to adapts to the latest channel state. The second is that the time complexity of authentication should be small enough to do packet-level authentication in association phase and detect attackers in time. In this paper, we propose a CSI-based authentication scheme, which can authenticate mobile devices at the packet level. Further, we provide an packet-level authentication framework based on neural networks. It uses a simple real-time authenticator update method to keep the authenticator valid. What's more, an ensemble of small-scale autoencoders are used to build the authenticator. It has been shown to significantly reduce the authentication's time complexity while maintaining the accuracy, providing the possibility for packet-level authentication. The evaluation shows that the packet-level framework can authenticate legitimate mobile devices with 95.19% accuracy and filter out attackers with even greater accuracy, which has higher time efficiency than traditional large-scale neural networks.**

***Index Terms*—Packet-level mobile device authentication, channel state information, physical layer signature, autoencoder, ensemble learning**

I. Introduction

The number of mobile users around the world has soared with the rapid development of wireless communication technology, and the coverage area of wireless correspondence infrastructure has increased quickly to satisfy people's needs[1]. Besides, the popularization of mobile devices such as smart phones and laptops helps the wireless networks be widely used in many important businesses such as financial transactions and business management. However, also more and more attackers use unauthorized mobile devices to invade wireless networks and violate the valid users, causing serious losses [2]. For most attackers, the first step is often to intercept radio messages and gain the identity information of valid users. Using the identity information, the attacker can pretend to be a valid user and launch further attacks (such as information forgery attacks and DDoS attacks [3]). The cryptographic mobile device authentication technologies in 802.11i have always been the main technologies to prevent these attacks, but the new attack scheme shows that it is not reliable [4, 5]. Also, they will cause large hardware costs and time overhead.

Recently, many non-cryptographic authentication technologies based on channel signature have been proposed and received widespread attention [6–11]. These technologies identify devices with the physical layer features, making it more difficult to launch attacks with unauthorized devices. Commonly used channel signatures base on Received Signal Strength (RSS), Channel Impulse Response (CIR) and CSI. RSS and CIR provide coarse-grained channel information on a single frequency point, while CSI presents fine-grained channel information. CSI is composed of the amplitude and phase of multiple subcarriers used in orthogonal frequency division multiplexing (OFDM). Different from cryptographic authentication, these technologies can authenticate and filter packets in the underlying hardware, and have higher security because physical layer signature is difficult to forge.

The basic authentication method for these technologies is use authenticator to perform pattern recognition between local signatures and the new signature. It must be noted that the mobile device is different from the fixed device because the channel state of the mobile device will change quickly. But this change is numerically continuous. [6] confirmed the possibility of CSI-based mobile device authentication. They measured the CSI of two mobile devices, and calculated the correlation coefficients between adjacent CSI measurements. The correlation coefficients of the same device are close to 1, while the correlation coefficients of different devices are distributed in [-1,0.2]. However, there are still two problems that cannot be ignored when using channel signatures to authenticate mobile devices:

- The channel state of the mobile device changes with

978-1-6654-3950-3/21/$31.00 ©2021 IEEE
DOI 10.1109/DSN-W52860.2021.00024

the device's movement, which means that we cannot expect that the authenticator generated with the original channel signatures can always be valid in subsequent authentication.
- The receiving rate of packets may reach the millisecond level. For complex authenticators, their generation and execution time efficiency cannot match with the receiving rate of packets. Thus, they may not authenticate the device at the packet level.

These problems bring huge challenges to the use of channel signatures in mobile wireless device authentication. In this paper, we try to use a simple authenticator update method to ensure that the local authenticator always matches the wireless channel state. The basic idea is updating the authenticator with the new legal signatures during authenticating. Further, the packet-level framework uses multiple small-scale autoencoders rather than a large-scale neural network. The former has significantly lower time complexity than the latter, making it possible to perform packet-level authentication.

The main contributions of our work are as follows:

- We propose a packet-level mobile wireless device authentication scheme based on channel signature, and design an packet-level authentication framework to carry out the authentication. It generates signature based on CSI and uses neural networks for authentication.
- We present a simple authenticator real-time update method to improve the performance of the authentication framework.
- We design an authenticator consisting of an ensemble of small-scale autoencoders in the packet-level framework. The theoretical analysis is used to prove that our design can lessen the time complexity of the authenticator.
- We conduct the experiments in the laboratory and evaluate the packet-level framework about accuracy and time efficiency.

The rest of the paper is organized as follows. Section 2 introduces related work, Section 3 presents the whole authentication framework and the implementation of each part. In Section 4, we show the experimental results and the performance evaluation of the packet-level framework. Finally, we provide the conclusion in Section 5.

II. Related Work

In recent years, researchers have proposed various non-cryptographic device authentication schemes based on physical layer signature. [7] found that the packets' timing information relates to the hardware and clock skew of devices closely, and extracted statistical features from the timing information as the device hardware signature.

[8] used multi-CIR to authenticate wireless devices, and proposed an enhanced multi-CIR authentication scheme to further improve the performance. In [9], the authors made a research on the RSS-based authentication techniques, and summarized several RSS-based authentication frameworks suitable for large-scale critical infrastructure security system. CSI has also used to generate physical layer signature [6, 10, 11]. It contains more fine-grained channel information than RSS and CIR, which can achieve better performance. Besides, CSI can be easily obtained from 802.11a/g/n devices while the collection of CIR depends on special equipment [12]. [6] provided different authentication frameworks for both fixed devices and mobile devices. In this paper, mobile devices are authenticated by comparing the correlation coefficient of adjacent CSI measurements with a given threshold. [10] used Convolutional Neural Network (CNN) to implement authenticator, and [11] employed a trained Generative Adversarial Neural Network (GAN) to authenticate wireless devices.

However, most of the authentication technologies ignored the impact of device mobility and used fixed authenticator [10, 11]. [6] used the correlation between adjacent CSI measurements for authentication rather than a fixed authenticator, but this rough judgment dropped the accuracy rate below 90%. Our packet-level framework uses a simple authenticator update method to ensure that the authenticator matches with the real-time channel state. Also, the authenticator is composed of an ensemble of small-scale autoencoders, which has less complexity than large-scale neural network used in [10, 11].

III. Packet-level Authentication Framework

In this section, we present the packet-level authentication framework, including packet preprocessor, signature generator and authenticator. We also detail the implementation of the signature generator and authenticator. Further, we analyze the time complexity of the authenticator theoretically.

A. Framework Overview

The packet-level mobile device authentication framework is an authentication module that works on the underlying hardware. It uses the CSI measurements collected from received packets to generate channel signatures and perform authentication and packets filtering. This framework has two working phases, including the training phase and the authentication phase.

The framework consists of the following three parts:

- **Packet Preprocessor:** It deals with receiving new packets and extracting CSI measurements.
- **Signature Generator:** It is responsible for the preprocessing of CSI measurements, which mainly includes noise weaken and partition. After preprocessing, it sends the sub-signatures to the authenticator.
- **Authenticator:** It handles the authentication and the update of internal neural network.

Most popular mobile devices using OFDM (such as 4G devices and 802.11a/g/n devices) supports the measurement and

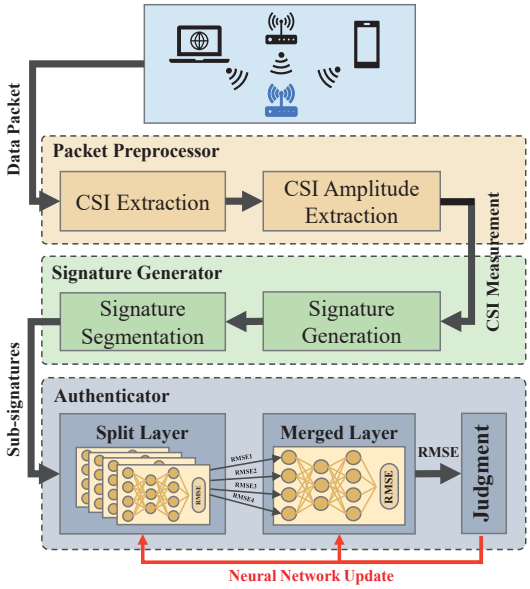


Fig. 1. The packet-level authentication framework

extraction of CSI. Thus, the available commercial hardware can work as the packet preprocessor, which is not our focus.

The rest of this section focuses on the design of the signature generator and authenticator. Please note that, by default we have collected enough valid signatures belonging to a mobile device before authenticating this device. This can be done. [6] provided a user profile generator, which actively collects CSI measurements from users and uses K-means algorithm to determine whether the measurements are collected from the attackers. Therefore, we can build a valid signature set for authenticator training before authentication. The signature set can be obtained by performing outlier elimination and low-pass filtering of the CSI measurements in the time domain, instead of using the signature generator we gave in the next sub-section. [6] details the method.

Here is a workflow to show the framework clearly:

- **Training Phase:** After building a valid signature set for the target mobile device:
  1) The signature generator uses the statistics of local signature set to initialize itself. Then, it evenly divides each signature into sub-signatures and sends them to the authenticator.
  2) The authenticator receives the sub-signatures and uses them to train the neural network.
- **Authentication Phase:** When the packet preprocessor receives a new packet:
  1) The packet preprocessor sends the CSI measurement to the signature generator.
  2) The signature generator first preprocesses the CSI measurement and generates a signature, then divides it into equal length sub-signatures.
  3) After receiving the sub-signatures, the authenticator send them to the trained neural net and outputs the execution result. The authenticator then judges whether the authentication is successful according to the result. If successful, it updates the neural net.

B. Signature Generator

CSI measurement reflects the channel response of the wireless channel. The packet-level framework only uses the amplitude of the CSI measurements to generate signatures.

The environment noise will reduce the accuracy of the framework, and the preprocessing of the CSI measurements is a significant part in the framework. According to surveys, the conventional preprocessing methods include time/frequency domain filtering and de-mean. The above methods are impractical in terms of memory, which are not suitable for packet-level authentication.

We propose a novel preprocessing scheme, which doesn't need to store current CSI measurements and perform simple calculations. Inspired by the damped increment statistics [13], we define a time decay factor to adjust the effect of the historical CSI measurements on the new measurement. The larger the time interval, the smaller the effect, so the signature always represents the latest channel state. We also proposed a standard deviation decay factor, which is positively associated with the standard deviation. Therefore, the CSI measurements will have a large effect on the new measurement in the subcarriers with large deviation, which helps to smooth the amplitude of these subcarriers.

Let $C = \{C_1, C_2, ..., C_N\}$ be the CSI measurement of a packet, where N is the number of subcarriers. The generator stores N tuples $M_i = (K, LS_i, QS_i, \sigma_i, t_{lst})(i = 1, 2, ..., N)$, where K is the number of packets received, LS_i is the linear sum of C_i, QS_i is the quadratic sum of C_i, t_{lst} is the latest packet's received time and $\sigma_i = \sqrt{|\frac{QS_i}{K} - (\frac{LS_i}{K})^2|}$ is the standard deviation of C_i. The decay factors are defined as:

$$d_{\lambda_1}(\Delta t) = 2^{-\lambda_1 \Delta t} \tag{1}$$

$$d_{\lambda_2}(\sigma_i) = 2^{-\frac{\lambda_2}{\sigma_i}} \tag{2}$$

where λ_1 is the time decay coefficient, Δt is the arrival time interval between the previous packet and the latest packet, λ_2 is the standard deviation decay coefficient.

In the training phase, we initialize $M_i(i = 1, 2, ..., N)$ with the statistics of the local signature set. The details is shown as follows. The initial value of K is 1, LS_i is the mean of C_i, QS_i is the mean of C_i^2, σ_i is the standard deviation of C_i and t_{lst} is the receiving time of the last packet.

In the authentication phase, when there is a new CSI measurement, the signature generator completes the following tasks:

1) Calculate d_{λ_1} and d_{λ_2}, let $\varepsilon = d_{\lambda_1} d_{\lambda_2}$
2) Calculate new tuples: $M_i' = (K', LS_i', QS_i', \sigma_i', t_{lst}')\,(i = 1, 2, \ldots, N)$, where $K' = \varepsilon K + 1$, $LS_i' = \varepsilon LS_i + C_i$ and $QS_i' = \varepsilon QS_i + C_i^2$
3) Calculate the signature: $S = \{\frac{LS_1'}{K'}, \frac{LS_2'}{K'}, ..., \frac{LS_N'}{K'}\}$
4) Return M_i' and the sub-signatures $\{S_1, S_2, \ldots, S_I\}$ obtained by evenly dividing the signatures into I pieces

It is important to note that the old tuple stored in the generator cannot be directly overwritten by the new one, because the new CSI measurement may belong to the attackers. However, the generator should only reflect the statistics of the true device, so we will only overwrite it after the new one has been proved to be valid.

C. Authenticator

The authenticator is the core of the packet-level framework. The basic structure of the internal neural network is an ensemble of small-scale autoencoders. An autoencoder is a neural network that extracts low-dimensional feature from input and then reconstructs it. It is composed of three layers of neurons that are fully connected. The neurons' number in the middle layer represents the dimension of the compressed feature. When the data set used for training has similar distribution, the autoencoder will improve its reconstruction ability as the number of training increases, and finally the reconstruction ability will be maintained at a stable level. In the authenticator, we use the root mean square error (RMSE) of the input and output to quantify this reconstruction ability. If the authenticator receives a new signature from the true device, it has a good ability to reconstruct the signature, and the RMSE is close to 0; on the contrary, if it receives a signature from an attacker, it cannot be reconstructed, and the RMSE is going to be large.

In practical design, we build the authenticator with a two-layer autoencoder network, as shown in Fig. 1. The first layer named split layer, which is composed of I autoencoders. The i-th auto-encoder in split layer is mapped to S_i. This layer is responsible for checking the reconstruction ability of each sub-signature, and its reconstruction error RMSEs will be sent to the next layer. The second layer named merged layer, which is composed of only one autoencoder. The RMSE of the input and output in this layer will be used as the final decision for authentication.

Before training the authenticator, we initialize each autoencoder with random numbers from the distribution $\mathcal{U}\left(\frac{-1}{dim(x)}, \frac{1}{dim(x)}\right)$, where x is the input. We detail the authenticator's implementation in both the training phase and the authentication phase.

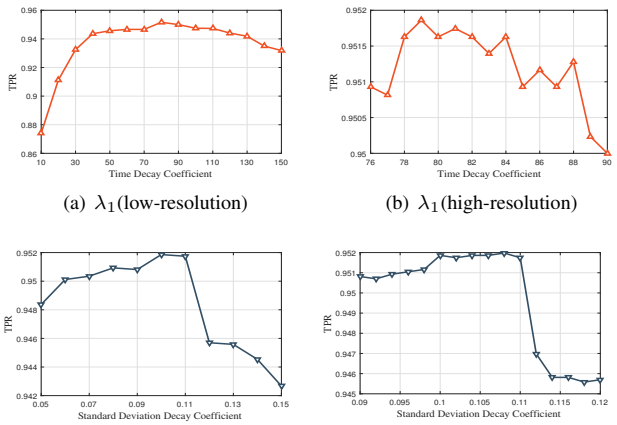
- **Training Phase:** $L1$ and $L2$ are used to represent the split layer and the merged layer. θ_i represents the i-th autoencoder in $L1$ and θ_0 represents the autoencoder in $L2$. The authenticator's workflow is as follows:
  1) Initialize vector I-dimensional vector v as the input of θ_0
  2) Normalize $S_i (i = 1, .., N)$
  3) For the θ_i in $L1$:
     (a) Forward propagation: input S_i into θ_i and obtain the reconstruction result S_i'
     (b) Backward propagation: use stochastic gradient descent (SGD) algorithm to update θ_i
     (c) Calculate $RMSE\,(S_i, S_i')$ and save it in the i-th value of v
  4) Normalize v
  5) Forward propagation: input v into θ_0 and obtain the reconstruction result v'
  6) Backward propagation: use SGD to update θ_0
- **Authentication Phase:** The authenticator's workflow in this phase is as follows:
  1) Initialize vector I-dimensional vector v as the input of θ_0
  2) Normalize $S_i (i = 1, .., N)$
  3) For θ_i in $L1$:
     (a) Forward propagation: input S_i into θ_i and obtain the reconstruction result S_i'
     (b) Calculate $RMSE\,(S_i, S_i')$ and save it in the i-th value of v
  4) Normalize v
  5) Forward propagation: input v into θ_0, obtain the reconstruction result v' and calculate $RMSE\,(v, v')$
  6) Judging:
     (a) $RMSE(v, v') < threshold$: update $L1$, $L2$ (using SGD) and the tuple of the signature generator
     (b) $RMSE(v, v') \geq threshold$: packet authentication fails

D. Complexity

In this part, we show the time complexity of the packet-level framework in both the training phase and the authentication phase. Then we compare the time complexity of integrated authenticator (an ensemble of small-scale autoencoders) and non-integrated authenticator (single large-scale autoencoder).

Before calculating the complexity, let's review the variables that will be used. N is the number of subcarriers, I is the number of sub-signatures (also the number of autoencoders in the split layer). Thus, N/I and I respectively represent the input dimension of the autoencoder in $L1$ and $L2$. In addition, we use α to represent the dimensionality reduction ratio of the middle layer in an autoencoder, thus the neurons' number in the middle layer of the autoencoder in $L1$ and $L2$ are $\alpha N/I$ and αI.

We first calculate the time complexity of the signature generator. The single operation's complexity in work steps (1) and (2) is $O\,(1)$ and the iterations is N, so the complexity is $O\,(N)$. Work step (3) performs N division operations, so the complexity is $O\,(N)$. The complexity of work step (4)



(a) λ_1(low-resolution) (b) λ_1(high-resolution)

(c) λ_2(low-resolution) (d) λ_2(high-resolution)

Fig. 2. The TPR under different decay coefficients

is $O(1)$. Therefore, the overall complexity of the signature generator is $O(N+N+1) = O(N)$.

For a single autoencoder, we assume that its input dimension is u and the dimensionality reduction ratio of the middle layer is α. In the forward propagation, the activation of the middle layer the output layer both requires $u \cdot \alpha u = \alpha u^2$ calculations. Therefore, the complexity of a forward propagation is $O(\alpha u^2) = O(u^2)$. The backward propagation has the same complexity. Before calculating the complexity of the authenticator, we make the worst plan for that the autoencoders in the split layer operate serially. Therefore, in the training phase, the complexity of the authenticator is $O(N^2/I + I^2)$. Let $N = \beta I$, where β is the length of the sub-signature. If we limit the size of the autoencoder in the split layer to 6 and below, then β can be regarded as a constant. Thus the complexity is $O(\beta^2 I + I^2) = O(I^2)$. In the authentication phase, we consider the worst case that all packets are authenticated successfully and the authenticator needs to be updated every time. Then the complexity of the authenticator is the same as the training phase.

Therefore, the time complexity of the framework is $O(N + I^2) = O(I^2)$. If we use a large-scale autoencoder instead of the ensemble, the complexity becomes $O(N + N^2) = O(N^2)$. It can be seen that the integrated design reduces the complexity from $O(N^2)$ to $O(I^2)$.

IV. Experiment and Evaluation

In this section, we evaluate the performance of the packet-level authentication framework in terms of its accuracy and time efficiency. We first describe the experiments and datasets, followed by selecting the parameters. Then, we show the authentication accuracy and time efficiency of this framework under different size of autoencoder. Then, we present the validity of the authenticator update method. Finally, we make a comparison between different machine learning algorithms.

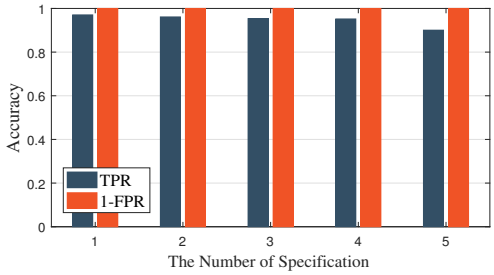


Fig. 3. The TPR and FPR (shown as 1-FPR for the convenience of observation) under different specifications of authenticator when $\lambda_1 = 79$ and $\lambda_2 = 0.105$

TABLE I
Authentication Performance in 5 Specifications

Metric	The Number of Specification				
	1	2	3	4	5
TPR	97.05%	96.11%	95.34%	95.19%	90.04%
FPR	0	0	0	0	0

A. Experiment Setup and Metrics

The test environment is a complex space with 16 tables, 2 cabinets and 16 desktop computers. It is nosier than isolation environment and the movement of people and objects can cause disturbance on the datasets. We use an ETTUS USRP B210 (single receiving antenna, 20MHz) as the packet preprocessor. It captures the packets sent by the mobile device under authenticating, then extracts source MAC and CSI measurements. Several USRPs (single transmitting antenna) work as the mobile devices to be authenticated are located in the external and internal corridors of the laboratory. They move back and forth at the normal pace of human and send packets at a speed of 100 pkts/sec. The packet preprocessor extracts the CSI measurements containing 48 subcarriers' amplitude from each packet and forms a mapping pair with its MAC. We obtain more than 121,000 CSI measurements from three different mobile devices. They are collected in different time periods during a day, and the duration of measurement in each time period exceeded 3 minutes.

We select three metrics for the evaluation: true positive rate (TPR), false positive rate (FPR) and accuracy. TPR is calculated as $TP/(TP+FN)$, where TP is the number of true positive samples and FN is the number of false negative samples; FPR is calculated as $FP/(FP+TN)$, where FP is the number of false positive samples and TN is the number of true negative samples; accuracy is calculated as $(TP+TN)/(TP+FP+TN+FN)$.

B. Select the Decay Coefficient

Fig. 2a shows the TPR when λ_1 changes from 10 to 150 (λ_2 = 0.1), with a resolution of 10. We can see that the TPR

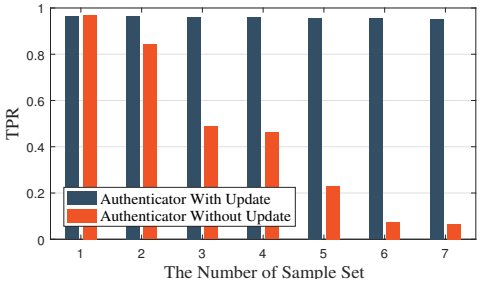


Fig. 4. The TPR of the authentication framework with and without authenticator update

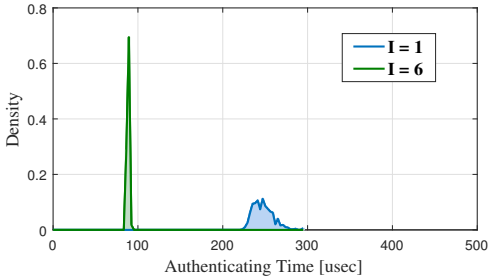


Fig. 5. Density plot of the authenticating time for one packet with $I = 1$ and $I = 6$

reaches the top when λ_1 = 80, so we increased the resolution to 1 and re-measured in [76, 90] (Fig. 2b). It can be seen from Fig. 2b that the TPR reaches peak when λ_1 = 79, so we set it to 79 in the following test. We use the same analysis method to select λ_2. The TPR curve in Fig. 2d reaches top in [0.10, 0.11], and we choose the median value 0.105.

C. Authentication Performance

In this part, we evaluate the authentication performance (including the accuracy and time efficiency) of the packet-level framework under different autoencoder specifications. We set the threshold of the authenticator to 0.10, which is an empirical value obtained in experiments.

We select 5 different specifications for evaluation. The I of these 5 specifications are 1, 2, 4, 6, and 12. Fig. 3 compares the TPR and FPR with different specifications (for clearer observation, we use 1-FPR to draw the figure). Table 1 give the results. It is worth noting that FPR is close to 0 in experiments, so we focus our attention on TPR. We can see that the authentication capacity of the system increases as autoencoder's scale increases. The recognition ability is the best when the authenticator uses a single large-scale autoencoder, but the TPR is only 1.86% higher than small-scale autoencoder with $I = 6$. so we let $I = 6$ for implementation.

We prove the effectiveness of the authenticator update method used in the authentication framework. We conduct two tests using the same data set, one with the authenticator update method and the other without. For each test, we divide the CSI data set into 7 sample sets according to the time of collection, which has 3000 CSI samples collected successively. Then, we get the TPR of each sample set and plot them in Fig. 4. It shows that the authenticator update method can greatly improve the performance of the authentication framework. The authenticator update method ensures that the authenticator matches with the current channel state.

To evaluate the time efficiency, we code the packet-level framework in C and perform it on a single-core Ubuntu virtual machine. We test the authenticating time of each packet with $I = 1$ and $I = 6$. The result is shown in Fig. 5. The per-packet authenticating time with $I = 1$ is about 240 usec, while other is about 90 usec. The time efficiency is increased by 2.67 times when small-scale autoencoder is used. Considering that the USRP only uses 48 subcarriers, the scale of neural network is relatively small when $I = 1$. Thus, the time efficiency improvement is not obvious in our experiments. The improvement will be more significant when 108 subcarriers are used.

D. Performance comparison with Different Algorithms

To conduct a more comprehensive evaluation of the packet-level framework, we implement authentication algorithms based on SVM, KNN and CNN, and evaluate them with the data set obtained in our experiments. Fig. 6 and Fig. 7 shows the accuracy and the time efficiency of different authentication algorithms, where EAE is the algorithm used in the framework.

It can be seen that only CNN has higher accuracy than EAE, which reaches 97.13%. However, its authentication time is 1.8 times longer than EAE, and this gap will be even greater when using higher-dimensional signature. SVM has a higher authentication efficiency then CNN, but its authentication acuracy is only 87.5%.

The result of comparison with different algorithms shows that the packet-level framework improves the time efficiency with high accuracy, which has a good application prospect.

There is a significant concern that the authentication leveraging CSI is strictly location signature instead of device signature. When the mobile devices reconnect to the network after a long period of offline, the authenticator may not tell who they are. Therefore, we are looking at the device signature based on CSI and packet time interval, which can authenticate mobile device and update the authenticator in our framework at the access phase.

V. Conclusion

In this paper, we propose a packet-level authentication scheme for mobile wireless device based on channel signature, and provide an authentication framework for implementation. This framework can be used to authenticate and filter each packet at the bottom of the device hardware, thereby enhancing

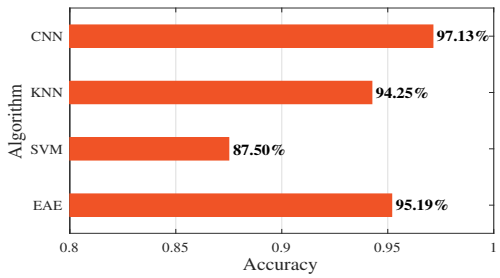


Fig. 6. The authentication accuracy under different authenticating algorithms

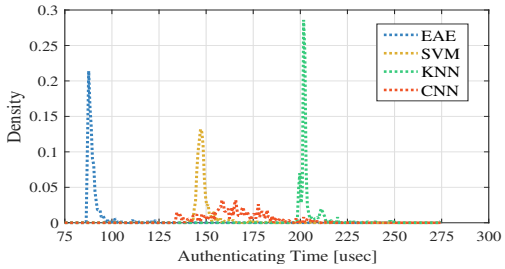


Fig. 7. The density plot of the authenticating time for one packet under different algorithms

the security of the wireless network. It uses CSI extracted from per packet to generate signatures. We present a simple authenticator update method, which can improve the authenticating performance. For packet-level authentication, we design an authenticator with low time complexity. To evaluate the performance of the packet-level framework, we programmed it and completed the mobile device authentication experiments in the laboratory. The results show that the framework can authenticate mobile device with high accuracy, and has a significant improvement in time efficiency compared with large-scale neural network.

ACKNOWLEDGMENT

This work is supported by Frontiers Science Center for Mobile Information Communication and Security, Southeast University, Nanjing, China. This work is also supported by Zhishan Youth Scholar Program Of SEU, Nanjing, China. Yubo Song is the corresponding author.

REFERENCES

[1] O. G. Aliu, A. Imran, M. A. Imran, and B. Evans, "A survey of self organisation in future cellular networks," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 336–361, First 2013.

[2] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1727–1765, Sep. 2016.

[3] J. Yang, Y. Chen, W. Trappe, and J. Cheng, "Detection and localization of multiple spoofing attackers in wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 44–58, Jan 2013.

[4] M. Vanhoef and F. Piessens, "Release the kraken: New kracks in the 802.11 standard," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 299–314. [Online]. Available: https://doi.org/10.1145/3243734.3243807

[5] R. Song, Y. Song, S. Gao, B. Xiao, and A. Hu, "I know what you type: Leaking user privacy via novel frequency-based side-channel attacks," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6.

[6] H. Liu, Y. Wang, J. Liu, J. Yang, Y. Chen, and H. V. Poor, "Authenticating users through fine-grained channel information," *IEEE Transactions on Mobile Computing*, vol. 17, no. 2, pp. 251–264, Feb 2018.

[7] L. Babun, H. Aksu, L. Ryan, K. Akkaya, E. S. Bentley, and A. S. Uluagac, "Z-iot: Passive device-class fingerprinting of zigbee and z-wave iot devices," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–7.

[8] N. Xie, J. Chen, and L. Huang, "Physical-layer authentication using multiple channel-based features," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2356–2366, 2021.

[9] A. Raza Cheema, M. Alsmadi, and S. Ikki, "Survey of identity-based attacks detection techniques in wireless networks using received signal strength," in *2018 IEEE Canadian Conference on Electrical Computer Engineering (CCECE)*, May 2018, pp. 1–6.

[10] R. Liao, H. Wen, F. Pan, H. Song, A. Xu, and Y. Jiang, "A novel physical layer authentication method with convolutional neural network," in *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, March 2019, pp. 231–235.

[11] K. St. Germain and F. Kragh, "Physical-layer authentication using channel state information and machine learning," in *2020 14th International Conference on Signal Processing and Communication Systems (ICSPCS)*, Dec 2020, pp. 1–8.

[12] A. Iqbal, V. Jeoti, M. Drieberg, and W. P. Wen, "A time-domain channel impulse response estimation method for an ofdm sounding system," in *2019 IEEE International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*, Aug 2019, pp. 1–5.

[13] L. Yang, Y. Song, S. Gao, B. Xiao, and A. Hu, "Griffin: An ensemble of autoencoders for anomaly traffic detection in sdn," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.