Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 2993019, 12 pages
https://doi.org/10.1155/2021/2993019

Research Article

Enhancing Packet-Level Wi-Fi Device Authentication Protocol Leveraging Channel State Information

Yubo Song[1,2], **Bing Chen**[1,2,3], **Tianqi Wu**[1,2], **Tianyu Zheng**[1,2], **Hongyuan Chen**[1,2], **and Junbo Wang**[2,3]

[1]*Key Laboratory of Computer Network Technology of Jiangsu Province, School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China*
[2]*Purple Mountain Laboratories, Nanjing 211189, China*
[3]*School of Information Science and Engineering, Southeast University, Nanjing 211189, China*

Correspondence should be addressed to Yubo Song; songyubo@seu.edu.cn

Received 10 May 2021; Accepted 18 October 2021; Published 17 November 2021

Academic Editor: Jinbo Xiong

Copyright © 2021 Yubo Song et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wi-Fi device authentication is crucial for defending against impersonation attacks and information forgery attacks. Most of the existing authentication technologies rely on complex cryptographic algorithms. However, they cannot be supported well on the devices with limited hardware resources. A fine-grained device authentication technology based on channel state information (CSI) provides a noncryptographic method, which uses the CSI fingerprints for authentication since CSI can uniquely identify the devices. But long-term authentication based on CSI fingerprints is a challenging work. First, the CSI fingerprints are environment-sensitive, which means that the local authenticator should be updated to adapt to the changing channel state. Second, the local authenticator trained with old CSI fingerprints is outdated when users reconnect to the network after being offline for a long time, thus, it needs to be retrained in the access phase with new fingerprints. To tackle these challenges, we propose a CSI-based enhancing Wi-Fi device authentication protocol and an authentication framework. The protocol helps to collect new CSI fingerprints for authenticator's training in access phase and performs the fingerprints' dispersion analysis for authentication. In the association phase, it provides packet-level authentication and updates the authenticator with valid CSI fingerprints. The authenticator consists of an ensemble of small-scale autoencoders, which has high enough time efficiency for packet-level authentication and authenticator's update. Experiments show that the accuracy of the framework is up to 98.7%, and the authenticator updating method can help the framework maintains high accuracy.

1. Introduction

Nowadays, Wi-Fi has become one of the most important wireless associated technologies [1]. However, there are serious security issues with Wi-Fi networks. Attackers can obtain the identity information of a valid device through wireless sniffing and then use this information to disguise themselves as legitimate devices [2, 3]. Attackers can steal confidential data or attack the internal websites after getting authorization [4, 5], or they can control other devices by sending spurious instructions [6]. Due to the widespread use of Wi-Fi networks in a range of critical services such as financial transactions and business management, attackers based on the identity of Wi-Fi networks can cause damage to public and private property and disrupt social order. Therefore, device authentication for Wi-Fi network security is indispensable [7, 8]. IEEE 802.11i provides cryptographic-based device authentication methods, but it has been proven to lack security [9–12]. What is more, Wi-Fi devices with limited hardware resources cannot support these authentication schemes well, which brings challenges to the usage of cryptographic-based device authentication technology.

In recent years, attempts have been made to use wireless channel characteristics for device authentication. One approach is to extract fine-grained fingerprints that can identify the device from the CSI and use fingerprint matching to authenticate the device [3, 13–16]. CSI refers to the channel frequency response (CFR) of a wireless multipath

channel, including the amplitude attenuation and phase shift of the channel. In a fast fading channel for indoor wireless communication, each radio path has different amplitude attenuation and time delay. As the radio signals arrive at the receiver along different paths, signals of different amplitudes and phases are superimposed to form the final signal, and the CSI reflects the amplitude fading and phase shift of the multipath channel. Because CSI reflects the amplitude and phase on all subcarriers, fine-grained device fingerprints can be generated with CSI. In a complex indoor environment, devices in different locations have different multipath channel states, and CSI is also expected to be different accordingly [17–23]. This feature allows CSI to be used for fingerprint extraction and device identification.

However, there are challenges to the application of this technology. It is well known that device authentication is usually a long-term task, and the channel state tends to change over time during the authentication process. As a result, the CSI fingerprint of the target device can also change, rendering the local authenticator ineffective. This will result in the rejection of legitimate devices. Therefore, it is significant to maintain a valid authenticator in an authenticated device during long-term operation for device authentication. More importantly, the authenticator often fails when a user logs back into the network after a long absence. This means that an alternative authentication method should be found to authenticate users at the access stage.

What is more, packet-level device authentication has high requirements on the computational complexity of the authentication algorithms. Packet-level device authentication requires extracting CSI fingerprints from each new data packet, data preprocessing, and fingerprint matching. Most of the CSI-based authentication technologies use large-scale neural networks such as ANN and CNN for fingerprint matching. Clustering algorithms such as K-means and SVM with high computational complexity are also used in authentication. Although these machine learning frameworks improve the accuracy of fingerprint matching, they are hard to apply for packet-level authentication in high throughput networks.

In this paper, we propose an enhanced Wi-Fi device authentication protocol based on CSI and authentication framework. We focus our attention on two working phases, including authentication in the access phase and authentication in the association phase. The protocol helps to collect CSI fingerprints in the access phase, and our framework performs fingerprint dispersion analysis for authentication. If the authentication in the access phase is successful, the newly collected CSI fingerprints are used to retrain the local authenticator. In the association phase, it provides packet-level authentication and updates the authenticator with valid CSI fingerprints. The local authenticator consists of a small-scale autoencoder rather than a neural network with high input dimensionality, which is computationally small enough for packet-level authentication and authenticator updates.

The main contributions of this paper are as follows:

(i) We propose an enhancing Wi-Fi device authentication protocol based on CSI, including the authentication procedure in access phase and association phase. It combines our authentication technology with the Wi-Fi communication protocol so that it can be applied in the existing Wi-Fi networks

(ii) We develop a Wi-Fi device authentication framework. In access phase, the framework is capable of the device authentication and the authenticator's retraining. In the association phase, it offers packet-level authentication service and updates the authenticator to make it fits the current channel state

(iii) We present an authentication method based on the CSI measurements' dispersion degree in access phase. We also provide a novel machine learning architecture for packet-level authentication in association phase. It consists of an ensemble of small-scale autoencoders, which has high enough time efficiency and accuracy for authentication and updating authenticator at a packet level

(iv) We implement our framework on Raspberry Pi and conduct real experiments in laboratory. Experimental results show that our framework can detect attackers and authenticate Wi-Fi devices with high accuracy under prolonged authentication. We also verify the effectiveness of the authenticator update method by conducting comparison tests

This paper is organized as follows: Section 2 shows the related work. Section 3 gives an overview of the authentication framework. Section 4 provides the authentication methods in access phase and association phase based on CSI fingerprints. Section 5 introduces the enhancing Wi-Fi device authentication protocol. Section 6 gives the evaluation. Section 7 concludes the paper.

2. Related Work

Recently, many wireless device authentication schemes based on CSI have been proposed. [17] applies K-means algorithm for authentication in access phase. It uses the number of clusters in the CSI fingerprints for judging whether there are fingerprints from attackers. [21] uses the correlation of neighbouring CSI fingerprints to determine whether the new CSI fingerprint comes from the valid device. [19] combined MIMO with CSI-based wireless device authentication technology, trying to use higher-dimensional CSI to improve authentication accuracy. Similarly, [18] also uses MIMO-CSI as the device fingerprint, but it proposes an authentication algorithm with a higher accuracy rate based on the LOF algorithm, which can achieve good accuracy even in an environment with low SNR.

Machine learning is widely used in wireless device authentication systems based on CSI. [13] uses CNN to classify new fingerprints and uses the classification results for subsequent device authentication. [17] uses K-means algorithm for access phase authentication. [24] leverages the SVM to obtain the similarity between the unknown CSI fingerprint and the local user profile, and the similarity is

used to determine whether the unknown fingerprint comes from a legitimate device. In addition, data dimensionality reduction algorithms are also used for the classification of high-dimensional data. Principal component analysis can extract the main feature components of high-dimensional data, thus, reducing the computational complexity of subsequent machine learning by downscaling the high-dimensional data to a low-dimensional space with the best linear combination. T-distributed stochastic neighbour embedding is also a dimensionality reduction algorithm. It can reduce the dimensionality of high-dimensional data to below 3 dimensions for data visualization.

However, all the abovementioned authentication systems do not provide a method for updating the authenticator during the authentication process. In this paper, we provide a method for updating the authenticator in real time, which can improve the performance of the authentication framework in the long run. In addition, none of the systems mentioned above show the time complexity of the authentication algorithms, and there is no guarantee that these techniques can be applied to high-throughput networks. However, the ensemble learning-based authenticator provided in this paper improves the time efficiency of authentication.

3. Framework Overview

We design a Wi-Fi device authentication framework that provides device authentication services in both the access phase and association phase, as shown in Figure 1. The framework can be implemented at the Wi-Fi access point (AP). Some work steps that require signaling interaction between the access point and the workstation (STA) can be accomplished by installing our authentication protocol on both devices, which will be shown in Section 5.

As shown in Figure 1, the Wi-Fi device authentication framework consists of two main parts, including authentication in the access phase and authentication in the association phase. The authenticator in between plays an important role in these two parts. The authentication framework will be described in terms of the device authentication workflow in the access phase and association phase as follows.

3.1. Authentication Workflow in Access Phase. According to the access process of the Wi-Fi device, STA needs to send an authentication request to AP. AP starts the CSI collection after receiving the request by sending a collection request (named collection REQ) to STA. STA then returns M measuring packets, which are used to extract CSI fingerprints. After extracting M CSI fingerprints, AP analyzes the dispersion degree of the fingerprints and judges whether the fingerprints are valid. If the fingerprints are invalid, the access request of STA will be rejected. Otherwise, the fingerprints are used to train the local authenticator for packet-level authentication in association phase. Before training the authenticator, the fingerprints should be preprocessed for noise reduction, which includes outlier elimination and smoothing. AP uses the processed fingerprints to train the local authenticator.

3.2. Authentication Workflow in Association Phase. For each data packet from STA, AP extracts the CSI fingerprint and sends it to the local authenticator. Then, AP judges whether the fingerprint is valid according to the output of the authenticator. It can be seen from Figure 1 that if the CSI fingerprint is judged to be valid, the AP accepts this data packet and uses the CSI fingerprint to train the authenticator. If the fingerprint fails the authentication, the data packet is dropped.

AP keeps a counter called F-Counter. It records the number of the packets that continuously fail the authentication. When the authentication fails, the F-Counter is incremented, and AP determines whether the connection should continue according to whether the threshold_f is reached. Otherwise, the F-Counter will be cleared.

4. Authentication Leveraging CSI Fingerprints

In this section, we focus on the device authentication based on CSI fingerprints. We first explain the basic idea of how to do authentication using CSI fingerprints, and then we describe the device authentication algorithms used in access phase and association phase.

4.1. Basic Idea. CSI is the CFR of a wireless multipath channel, including the amplitude attenuation and phase shift. In a Wi-Fi network, the wireless channel between each STA and AP is unique, which means that the CSI of the channel between each STA and AP is also unique. In the light of this idea, the AP can uniquely mark a STA through the CSI obtained from the packets sent by this device. Therefore, we refer to the CSI collected from the STA as CSI fingerprints. Figure 2 shows the amplitude image of the CSI fingerprints obtained from four STA placed in different positions. We can see that the fingerprints' distributions of the same device are concentrated, while the fingerprints' distributions between different devices are different. Considering that there is often interference in real environment, we collected CSI fingerprints under three interference conditions, and the results are shown in Figure 3. Even if there is interference, the CSI fingerprint is still stable. Therefore, it is reasonable to use CSI fingerprints for Wi-Fi device authentication.

In access phase, AP sends CSI fingerprints collecting requests to the target STA. If there is no attacker or the attacker does not do any response, the fingerprint set will only contain fingerprints with the same distribution, which belong to the target STA. However, if the identity-based attackers make response to AP, the fingerprint set will contain fingerprints of at least two different distributions. The framework takes advantage of this distinction to perform authentication. The framework quantifies the number of distributions by the dispersion degree of the fingerprint set.

In the association phase, the authenticator has been trained with new CSI fingerprints obtained in access phase. Our framework extracts CSI fingerprint from each packet received and perform authentication at the packet level. Autoencoders are used to build the local authenticator. The trained autoencoder is expected to rebuild the fingerprints of the target STA with low error, but it cannot rebuild the

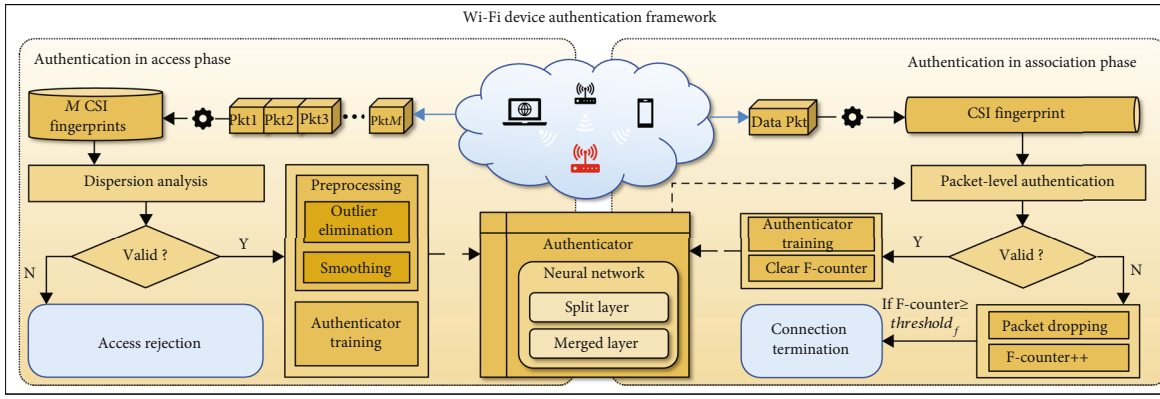


Figure 1: The framework of our Wi-Fi device authentication system.

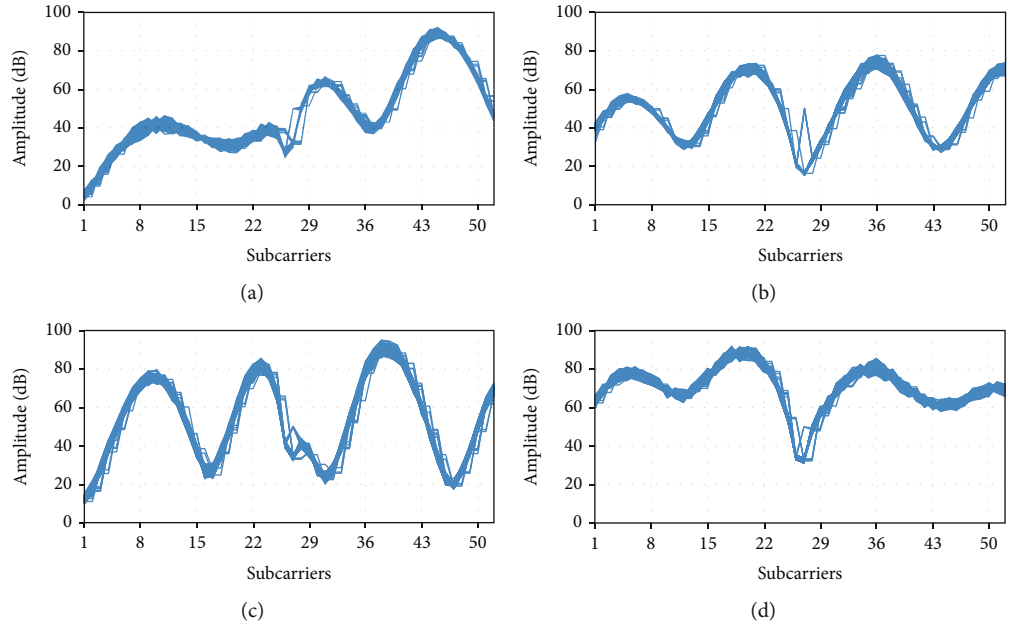


Figure 2: The amplitude of CSI collected in different locations. (a) Location 1. (b) Location 2. (c) Location 3. (d) Location 4.

unknown fingerprints collected from the attackers well. The framework makes use of this characteristic for authentication. To keep the authenticator adapt to the channel state, each valid fingerprint is used to train the authenticator. Moreover, we use an ensemble of small-scale autoencoders instead of a large-scale autoencoder, because the time efficiency of training and executing an ensemble of small-scale autoencoders is higher than that of a large-scale autoencoder. It is confirmed in Section 4.4 by theoretical derivation and Section 6.2 by evaluation results.

4.2. Authentication Scheme in Access Phase

4.2.1. Dispersion Analysis. Let C_D stands for the fingerprints set collected from the STA D and the k-th CSI fingerprint in C_D indicated by $C_D^k = \{C_{D,1}^k, \cdots, C_{D,N}^k\}$ $(k = 1, \cdots, K)$, where N is the number of subcarrier, and K is the number of samples obtained from D.

After collecting CSI fingerprints from STA, the framework uses standard deviation of C_D to quantify the dispersion degree:

$$\sigma = \left(\frac{1}{K}\sum_{k=1}^{K}\left(C_D^k - \bar{C}_D\right)^2\right)^{1/2}, \quad (1)$$

where $\bar{C}_D$ is the mean of the fingerprints in C_D.

If the standard deviation exceeds the predefined threshold, the STA is judged to be invalid and denied access to the network. Otherwise, the STA successfully access to the network, and the fingerprints are used for training the local authenticator. We designate the threshold as threshold_{ac}, which means the threshold used in access phase.

4.2.2. Fingerprint Preprocessing. We found that there are often fingerprints that far exceed the expected numerical fluctuation range, as shown in Figure 4(a). They do not

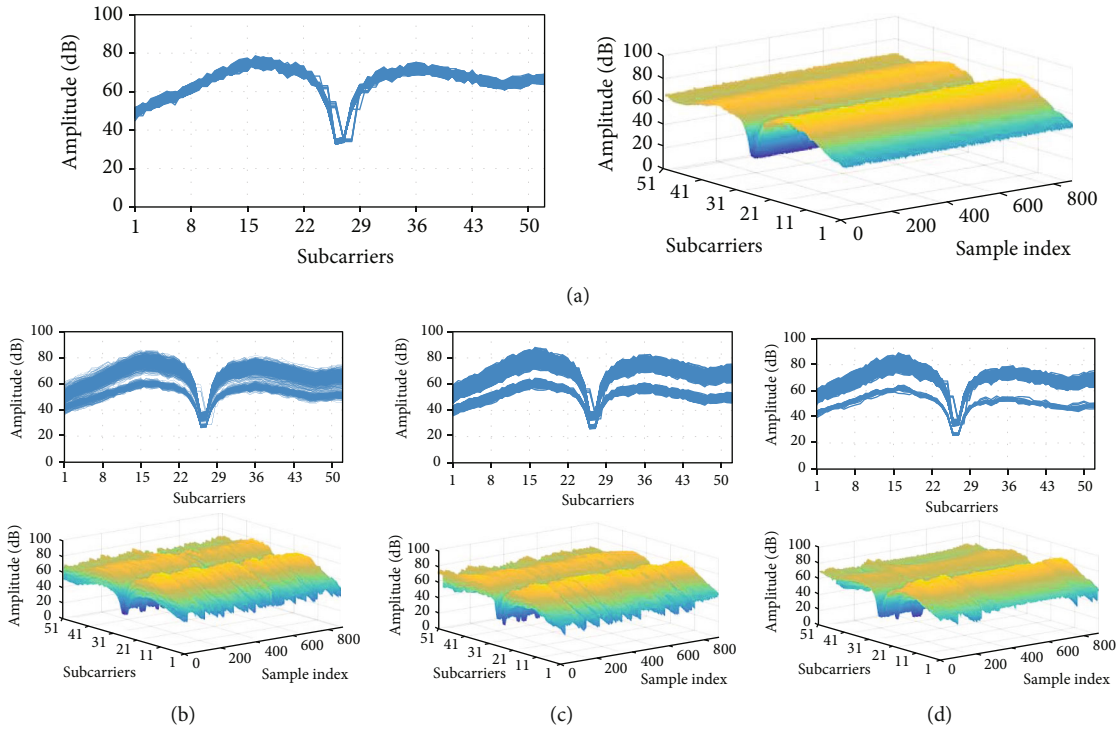


FIGURE 3: The amplitude of CSI collected in different interference environments. (a) Non-interference. (b) Interference -1 m. (c) Interference -2 m. (d) Interference -3 m.

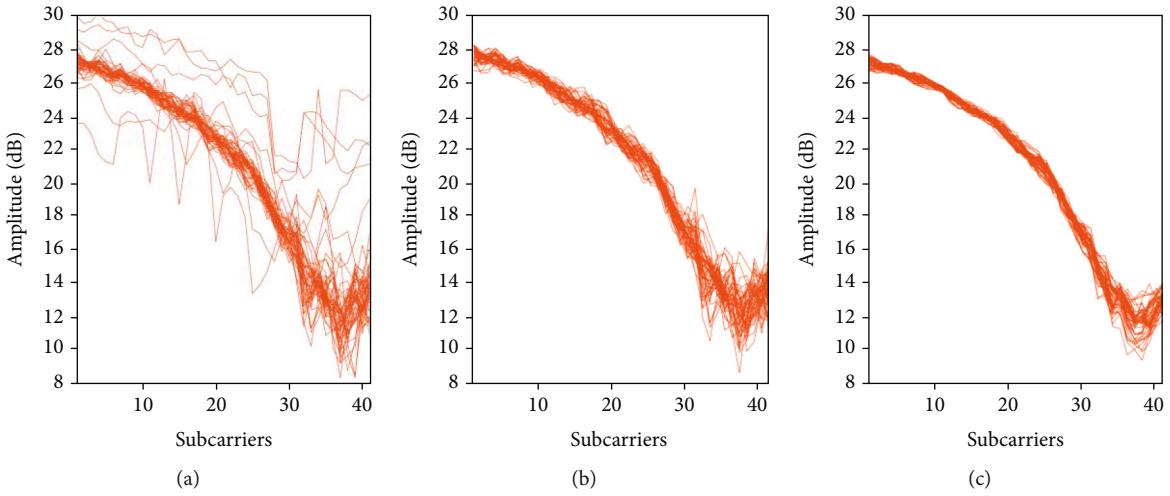


FIGURE 4: The amplitude of CSI in different preprocessing stages. (a) Before preprocessing. (b) After removing outliers. (c) After smoothing.

contain any information of device identity and will reduce the accuracy of system. Therefore, we use the Hampel identifier to exclude these outliers. In addition, the CSI fingerprints of the same device will change under noise interference. Even though the CSI images of the same device still have the same trend, but the dispersion increases. The local outlier factor describes the difference between fingerprints based on the Euclidean distance. It means the noise will cause the reference sample group become discrete and reduce the probability of abnormal CSI being detected. Therefore, we smooth the CSI fingerprints in the time domain. The fingerprint preprocessing consists of two stages, including the outlier elimination and the smoothing.

(1) Outlier Elimination. We perform outlier detection on each subcarrier individually. First, we arrange the amplitude samples of n-th subcarrier in chronological order, which can be presented as $L_{D,n} = (C^1_{D,n}, \cdots, C^K_{D,n})$. Then, $C^i_{D,n}\,(i = l+1, \cdots, K-l)$ and its $2l$ surrounding amplitude samples form a window W_i. In W_i, the absolute deviation

of each amplitude sample is used to estimate the standard deviation of the amplitude samples. The standard deviation is shown as follows:

$$\sigma_{D,n}^{I} = \frac{1}{\gamma}\text{median}\left(\left|C_{D,n}^{i} - C_{D,n}^{I}\right|\right), \tag{2}$$

where $C_{D,n}^{I}$ indicates the median of W_i and $\gamma = \sqrt{2}\text{erfinv}(0.5)$, erfinv on behalf of the inverse error function.

We then perform the following numerical substitution on each amplitude sample in W_i to exclude the outlier:

$$C_{D,n}^{i} = \begin{cases} C_{D,n}^{i}, \left|C_{D,n}^{i} - C_{D,n}^{I}\right| \le \eta\sigma_{D,n}^{I}, \\ C_{D,n}^{I}, \left|C_{D,n}^{i} - C_{D,n}^{I}\right| > \eta\sigma_{D,n}^{I}, \end{cases} \tag{3}$$

where η is a threshold used to judge whether the sample is an outlier.

We repeat the above work for $W_i\ (i = l+1, \cdots, K-l)$ on each subcarrier. Figure 4(b) shows the CSI fingerprints without the outliers.

(2) Smoothing. We smooth the amplitude of each subcarrier in the time domain to reduce this effect. The smoothing process is described as follows:

$$\tilde{C}_{D,n}^{k} = \frac{1}{\omega}\sum_{\max\left(0,k-\left\lfloor\frac{\omega}{2}\right\rfloor\right)}^{\min\left(K,k+\left\lfloor\frac{\omega-1}{2}\right\rfloor\right)} C_{D,n}^{k}, \tag{4}$$

where ω indicates the length of smoothing window. Figure 4(c) shows the CSI fingerprints after smoothing.

4.3. Packet-Level Authentication Scheme in Association Phase. We assume that the STA has been successfully connected and starts to transmit data. The successful access of the STA means that the fingerprints collected by the AP is valid, which means that we can successfully authenticate the data sent by the STA in association phase.

Before authenticating the STA in association phase, the authenticator should be trained with the processed fingerprints obtained in access phase.

4.3.1. Authenticator Training. Figure 5 shows the organization of the authenticator. The authenticator consists of two layers, including split layer and merged layer. We can see from Figure 5 that the basic functional unit of each layer is autoencoder, which is shown on the right of Figure 5. It consists of three layers of fully connected neurons. The input and output layers have the same number of neurons, and the middle layer, named hidden layer, has a smaller number of neurons. The autoencoder first compresses the input samples and then reconstructs them into the original dimensions. It tries to learn the distribution characteristics of the input sample set for reducing the reconstruction error.

In this paper, the autoencoders in split layer learn the distribution characteristics of the input CSI fingerprints and reconstruct them. The reconstruction error of each autoencoders is sent to the output autoencoder in merged layer. The root mean squared error (RMSE) between input and output is used for indicating the reconstruction error.

Before entering a new CSI fingerprint into split layer, authenticator first normalizes the fingerprint. The normalization algorithm is shown as follows:

$$S_{D,i}^{k} = \frac{C_{D,i}^{k} - C_{D,\min}^{k}}{C_{D,\max}^{k} - C_{D,\min}^{k}}, \tag{5}$$

where $C_{D,\min}^{k}$ and $C_{D,\max}^{k}$ represent the minimum and maximum values of $C_{D,i}^{k}\ (i = 1, \cdots, N)$.

The normalized fingerprints S_D^k are equally divided into I subfingerprints ($S_D^k = \{S_D^{k(1)}, \cdots, S_D^{k(I)}\}$), and they are sent to the autoencoders in split layer in order (pictured in Figure 5). Next, we focus on the training process of a single autoencoder.

Next, we focus on the training process of the authenticator. We initialize each autoencoder in two layers with random numbers with the distribution $\mathcal{U}(-1/\dim(x), 1/\dim(x))$ (x is the input of the autoencoder) before training. For clearly presenting the training process, we use $L1$ and $L2$ to represent split layer and merged layer. We use θ_i for the i-th autoencoder $L1$ and θ_0 for the autoencoder in $L2$. The training algorithm is shown in Algorithm 1, where v is the input of θ_0, $S_D^{k(i)\prime}$ is the output of θ_i, and w' is the output of θ_0.

Authenticator's update is important in association phase, because if the authenticator become outmoded, the data packets sent from STA will be refused. The updating method of the authenticator presented in this paper can improve this problem without compromising the safety of the system.

4.3.2. Packet-Level Authentication. For a new data packet from the STA, the framework first extracts the CSI fingerprint C_D from the packet and then normalizes it using equation (5). The normalized CSI fingerprint S_D is sent to the local authenticator, and the authentication algorithm is shown in Algorithm 2, where v is the input of θ_0, $S_D^{(i)\prime}$ is the output of θ_i and w' is the output of θ_0. If the CSI fingerprint is judged to be valid, it will be used to train the local authenticator.

4.4. Packet-Level Authentication Complexity. This subsection shows the time complexity of the packet-level authentication in association phase. To show the improvement of ensemble learning on time efficiency, we make a comparison between the time complexity of our authenticator (an ensemble of small-scale autoencoders) and a single large-scale autoencoder by mathematical calculations.

In the authentication framework, N is the number of subcarriers, and the number of subfingerprints (also the number of autoencoders in split layer) is I. Thus, the input dimension of the autoencoder in $L1$ and $L2$ is, respectively, equal to N/I and I. We use α as the dimensionality reduction

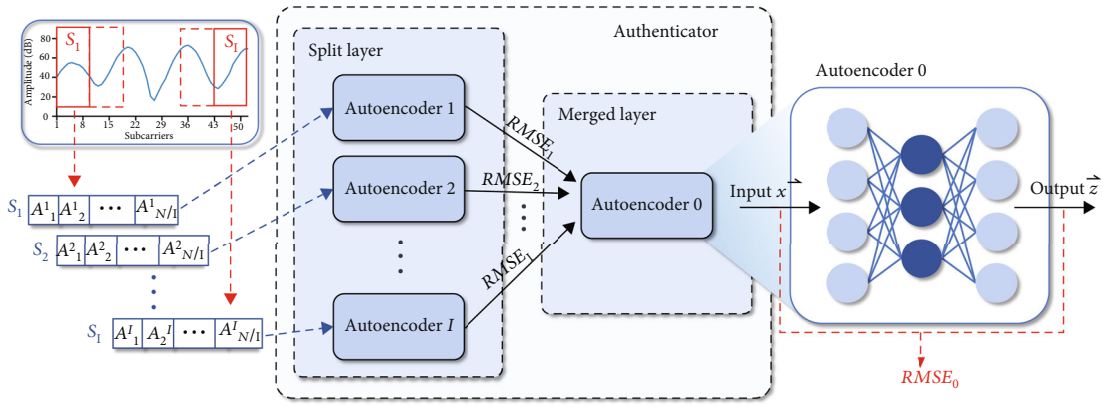


Figure 5: The local authenticator for authentication in association phase.

Input: S_D^k
1: $\boldsymbol{v} \longleftarrow \text{zeros}(I)$
//**t** rain Split layer
2: **For** θ_i in $L1$ **do**
3: $\theta_i, S_D^{k(i)\prime} \longleftarrow \text{SGD}(S_D^{k(i)})$
4: $\boldsymbol{v}[i] \longleftarrow \text{RMSE}(S_D^{k(i)}, S_D^{k(i)\prime})$
5: **End for**
//train merged layer
6: $\boldsymbol{w} \longleftarrow \text{norm}_{0-1}(\boldsymbol{v})$
7: $\theta_0, \boldsymbol{w}' \longleftarrow \text{SGD}(\boldsymbol{w})$
8: Return $\text{RMSE}(\boldsymbol{w}, \boldsymbol{w}')$

Algorithm 1: The authenticator training algorithm.

Input: S_D
1: $\boldsymbol{v} \longleftarrow \text{zeros}(I)$
2: **For** θ_i in $L1$ **do**
3: $S_D^{(i)\prime} \longleftarrow h_{\theta_i}(S_D^{(i)})$
4: $\boldsymbol{v}[i] \longleftarrow \text{RMSE}(S_D^{(i)}, S_D^{(i)\prime})$
5: **End for**
6: $\boldsymbol{w} \longleftarrow \text{norm}_{0-1}(\boldsymbol{v})$
7: $\boldsymbol{w}' \longleftarrow h_{\theta_i}(\boldsymbol{w})$
8: **If** $\text{RMSE}(\boldsymbol{w}, \boldsymbol{w}') < threshold_{as}$ **then**
//train Split layer and merged layer
9: **For** θ_i in $L1$ **do**
10: $\theta_i \longleftarrow \text{SGD}(S_D^{(i)})$
11: **End for**
12: $\theta_0 \longleftarrow \text{SGD}(\boldsymbol{w})$
13: **End if**
14: **Return** $\text{RMSE}(\boldsymbol{w}, \boldsymbol{w}')$

Algorithm 2: The authenticator execution algorithm.

ratio of the autoencoder's middle layer, thus, the neurons' number in the middle layer in $L1$ and $L2$ are $\alpha N/I$ and αI.

If the input of an autoencoder is u and the dimensionality reduction ratio is α, the activation of the middle layer and the output layer both requires $u \bullet \alpha u = \alpha u^2$ calculations. Therefore, the complexity of the activation is $O(\alpha u^2) = O(u^2)$. The backward propagation has the same complexity. Before calculating the complexity of the authenticator, we assume that the autoencoders in split layer operate serially in the framework. Therefore, the complexity of the authenticator's training is $O(N^2/I + I^2)$. Let $N = \beta I$, where β is the length of the subfingerprint. If we limit the size of the autoencoder in 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 association phase, we assume that all packets are authenticated successfully and the authenticator needs to be updated every time. Then, the complexity of the authenticator is $O(I^2)$.

If we use a single autoencoder instead, the complexity becomes $O(N^2)$. Our authenticator based on ensemble learning reduces the time complexity of the framework from $O(N^2)$ to $O(I^2)$.

5. The CSI-Based Authentication Protocol

It can be seen from the previous section that the association and cooperation between the AP and the STA are the key to do device authentication and fingerprint database update in both the access and the association phase. Therefore, we present an enhancing Wi-Fi device authentication protocol in this section, which is designed based on the existing Wi-Fi association process.

5.1. Authentication Procedure in Access Phase. This subsection describes the authentication procedure in access phase, which is shown in Figure 6(a). The steps are shown as follows:

(i) STA sends probe request to request access to the network. AP responds with probe response after listening to the request, indicating that it can provide access service

(ii) STA then sends authentication request for the authentication. AP returns collection request to inform STA that it is ready for CSI collection

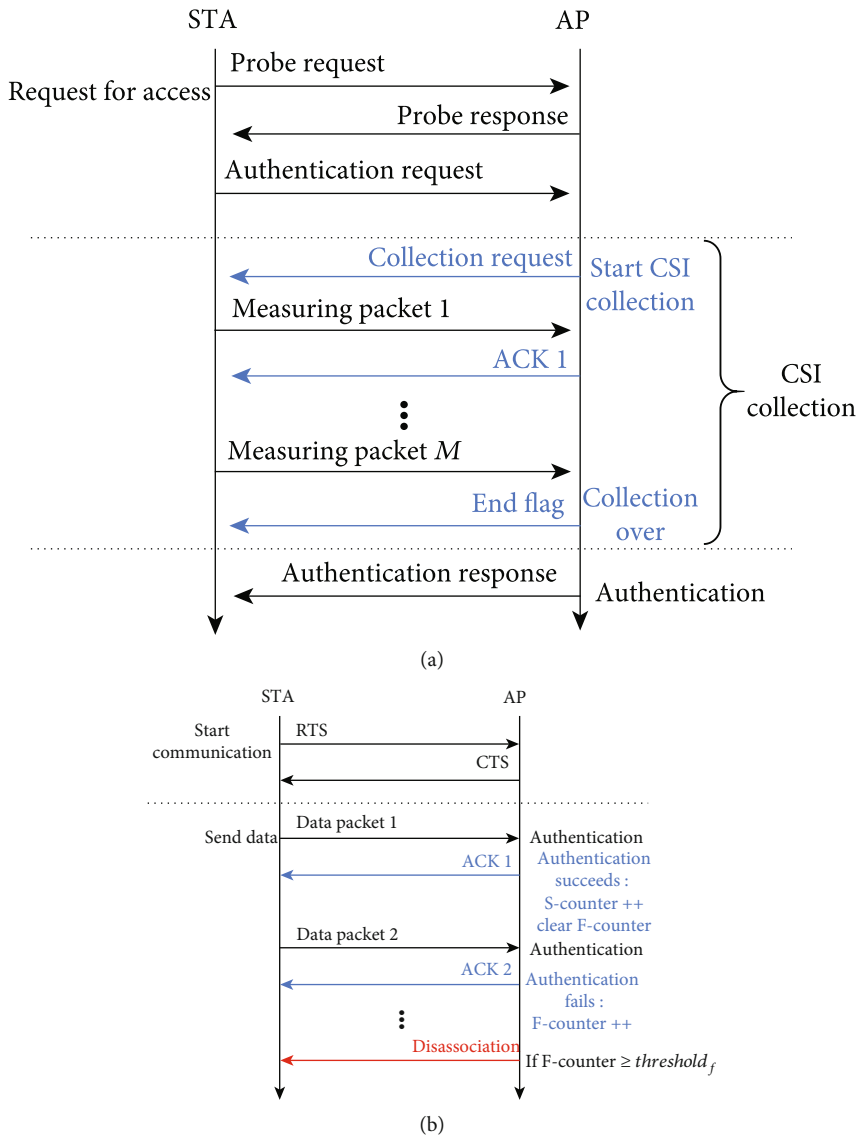


Figure 6: The procedure of CSI-based authentication protocol. (a) Access phase. (b) Association phase.

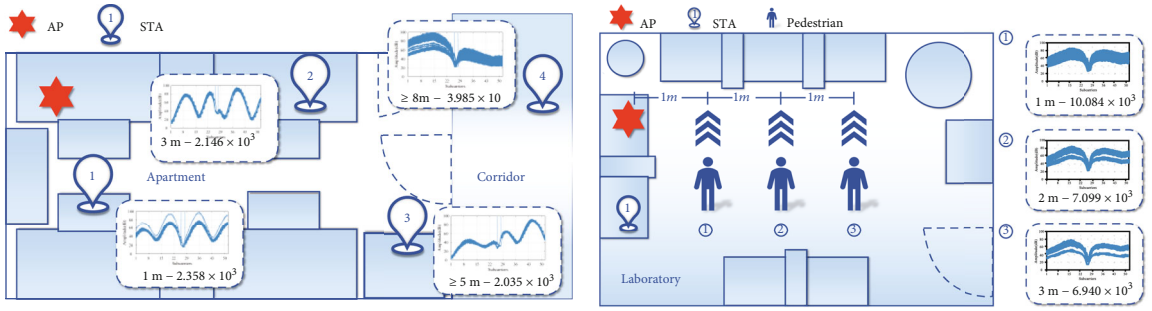
(iii) After receiving collection request, STA starts to send measuring packets with a fixed data length of 24 bytes. After receiving the data packets, AP responds with ACK to tell STA to continue or stop

(iv) After collecting enough CSI fingerprints, AP authenticates the target STA and judges whether the STA is valid. The authenticating result is sent to the target STA through authentication response

5.2. Packet-Level Authentication Procedure in Association Phase. Figure 6(b) shows the packet-level authentication in association phase. In Wi-Fi association, CSMA/CA is used to avoid the collisions among packets. The STA exchanges RTS/CTS with the AP before starting to transmit data to inform other devices to remain silent. After exchanging RTS/CTS, the AP performs CSI matching on each received data frame, judges whether the current connection is valid according to the matching situation, and decides whether to continue association or not (as shown in section 2).

The real-time update of the authenticator is based on the active communication between the AP and the STA. When the STA is in sleeping mode, keeping the CSI collection will cause greater power consumption and increase the network burden, so this is not recommended. After a device returns to be active, the local authenticator may become outmoded. To retrain the authenticator, we disconnect after threshold$_f$ consecutive data packets fail the authentication (see section 2); then, the authentication process will return to access phase.

In addition, we must also consider the existence of the attacker. When an attacker uses a forged identity to send a data packet, the AP will get the failed authentication result and discard the packet. If the AP continuously receives threshold$_f$ packets from the attacker, the connection is broken. We can find that the attacker cannot attack the AP effectively during this whole process.



(a) (b)

Figure 7: The test scenarios used in the evaluation. (a) Apartment. (b) Laboratory.

(i) STA sends RTS to tell AP that it is going to send data. AP responds with CTS after receiving the request, indicating that it is ready receiving and clearing the channel

(ii) STA starts to send data packets. When AP receives a data packet, it responds with ACK. AP extracts the CSI of the packet and performs authentication. If the authentication is successful, the S-Counter is increased by 1, and the F-Counter is cleared. Otherwise, the F-Counter is increased by 1

(iii) If F-Counter reaches threshold$_f$, AP sends disassociation to STA for disconnection

6. Performance Evaluation

In this section, we make an evaluation on the performance of our authentication framework.

6.1. Experiment Setup and Metrics. To simulate real application scenarios, we conducted experiments in both apartment and laboratory, which are complex multipath environments and in public networks with high traffic.

A commercial Wi-Fi device, HUAWEI TAS-AN00, works as STA transmits data packets at a rate of 100 pkt/sec in 20 MHz Wi-Fi channel on 2.4 GHz, which runs EMUI 11.0.0. We use the Raspberry Pi 3b+ as AP, which runs Raspbian Buster Lite 4.19.97 and is equipped with BCM43455c0 Wi-Fi card. By modifying the Wi-Fi card driver, it can pack the CSI fingerprints extracted from the specified devices' data packets in UDP datagrams and send it to the application layer. We then use TCPDump to grab the UDP datagrams and read the CSI fingerprints. The plug-in used to extract CSI from the Wi-Fi card of Raspberry Pi 3b+ is available in [25].

In the experiments performed in apartment, we place the STA in four locations fixedly, which is shown in Figure 7(a). The first three locations are distributed in the apartment at different distances, and the fourth location is in the corridor outside the apartment. We use these experiments to analyze the impact of distance on the performance of the authentication framework. In the experiments performed in laboratory, we put the AP and STA in fixed locations and walk back and forth in the positions shown in Figure 7(b). These three

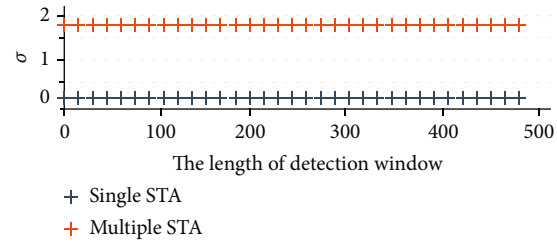


Figure 8: The standard deviation of CSI fingerprints with and without attacker.

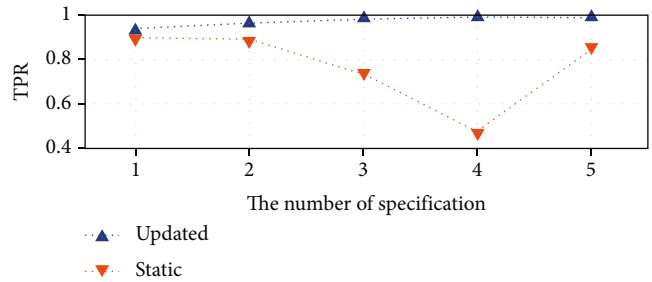


Figure 9: The authentication framework's accuracy using different specifications of authenticator.

modes of movement interfere with the collection of CSI to different degrees. These experiments help us observe the effects of different levels' interference on the authentication framework's performance.

In the apartment experiments, we collect 10,000 CSI fingerprints at each location, and each CSI fingerprint contains 52 subcarrier amplitudes. These datasets serve to evaluate the performance of our authentication framework in access phase and association phase. In the laboratory experiments, the same number of fingerprints is obtained in three modes of movement, which are used for evaluation in association phase. Since the AP will receive irrelevant frames from other unknown Wi-Fi devices, we record the MAC of each STA and filter the frames by MAC checking. For each valid frame received, the AP extracts CSI fingerprint and saves it locally.

To evaluate the performance of the authentication framework, we use the true positive rate (TPR) as the specific criteria for judging. 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.

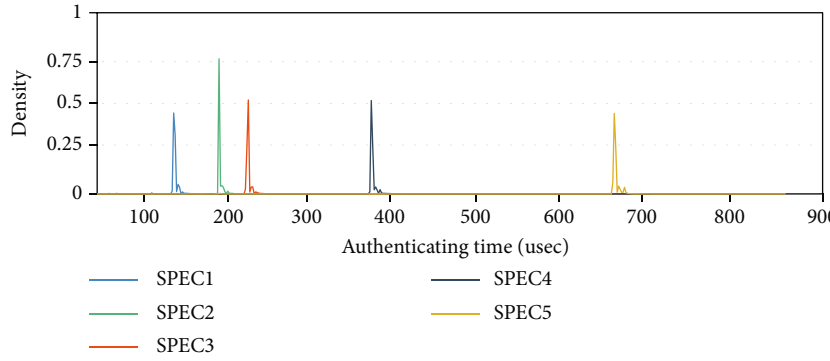


Figure 10: Per-packet authenticating time using different specifications of authenticator.

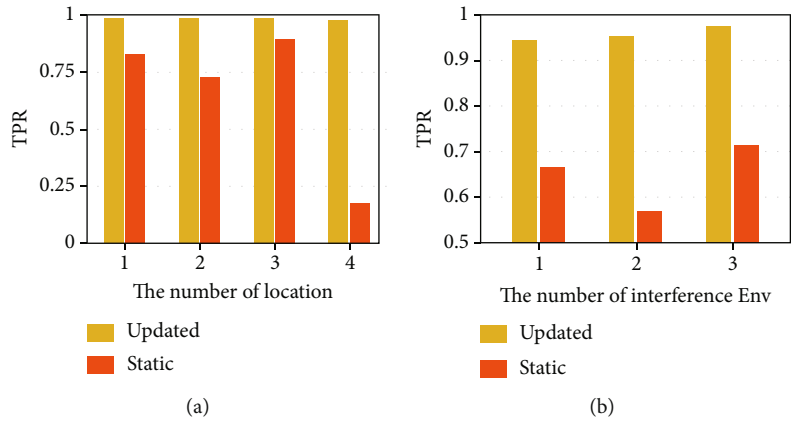


Figure 11: The authentication framework's accuracy in two test scenarios. (a) Apartment. (b) Laboratory.

6.2. Authentication Performance. In access phase, we need to determine the decision threshold for dispersion analysis. To this end, we use the CSI fingerprints obtained in the apartment experiments to test the degree of dispersion with and without the attacker. The test result is shown in Figure 8. Among them, the blue plus sign marks the standard deviation of the CSI fingerprints when only STA1 requests access, and the red plus sign marks the standard deviation when four STA request access at the same time. X represents the length of the detection window. Different detection window lengths have a stable standard deviation, and the standard deviation of a single STA is much lower than that of multiple STAs. Therefore, we set the decision threshold to 0.5, which can be used for access phase's authentication in the apartment experiments with an accuracy of 1. We set the detection window length to 100. Because on the one hand it can shorten the CSI collecting time compared with a longer window, on the other hand, it can reduce the probability of misjudgment caused by channel noise compared with a shorter window.

The number of autoencoders used in the authenticator's split layer and merged layer is an important factor affecting the overall performance of the authentication framework. We choose 5 different specifications of authenticators for performance evaluation, their I are 13, 6, 4, 2, and 1. Figure 9 shows the TPR of each specification. We can see that the TPR increases from 93.278% to 99.278% as the number of autoencoders in split layer increases. However, the increase in I means that the number of neurons in autoencoder increases, and the time efficiency of the authenticator will decrease, as analyzed in Section 4.4. To prove our theoretical analysis, we recorded the packet-level authentication time of 10,000 data packets under 5 different specifications and plotted the density map, which is shown in Figure 10. As we can see in the figure, the packet-level authenticating time is around 150 usec when $I = 13$. However, the authenticating time up to 660 usec when $I = 1$, which is 4.4 times as many as the former. Based on the above factors, we choose to use the third specification authenticator ($I = 4$) for the following evaluation.

In the apartment experiments, the authentication results of the first three locations are similar, and their TPR is higher than 98.6% (depicted in Figure 11(a)). However, the TPR of the position 4 drops to 97.87%. The CSI fingerprints' variance of location is up to 3.985×10^3 (as shown in Figure 7(a)), which is much higher than the variance of the first three locations. This shows that the distance between AP and STA is not an essential factor affecting the stability of CSI fingerprints. Although the increase of the distance will increase the possibility of interference, the distance has little effect on the stability of the CSI fingerprint in a stable indoor environment and will not affect the authentication

framework's performance. In the corridor, the CSI fingerprints show relatively poor stability, because in the absence of line of sight (LOS), any small disturbance will have a greater impact on the multipath channel state.

In the laboratory experiments, the third device with the least disturbance has the largest TPR, which is 97.53%. Meanwhile, the first device with the most interference had a TPR of 94.36% (as shown in Figure 11(b)). This is in line with our expectations. The CSI sample variances of the three interference environments are 10.084×10^3, 7.099×10^3, and 6.941×10^3 (Figure 7(b)). We can see that noise will reduce the stability of CSI fingerprints collected from Wi-Fi devices and then decrease the accuracy of the authentication framework. However, the results show that the accuracy of the authentication framework can still reach more than 94% even when there are disturbances such as people walking and objects moving at 1 m, which means that our authentication framework has strong robustness.

In each performance evaluation, we used two different methods for authentication. The first method is updating the authenticator with the update scheme shown in Section 4.3. The other one is keeping a static authenticator in the AP. In the environments with less interference such as locations 1, 2, and 3 in the apartment, the minimum TPR when using the static authenticator is 72.91%, which is 25.77% lower than using the updated authenticator; the maximum is 89.69% and is 9% lower than the second method (Figure 11(a)). In the environments with interference (location 4 in the apartment and laboratory), the minimum TPR when using the first authentication method is 17.37%, which is 80.5% lower than using the updated authenticator; the maximum is 71.49% and is 26.04% lower than the second method (depicted in Figures 11(a) and 11(b)). Therefore, the authenticator update scheme presented in this paper can improve the performance of the authentication framework, and the effect is more significant under interference environment.

However, there are some shortcomings in this scheme. First, because CSI is highly dependent on the environment in which the device is located, it faces frequent disconnections if the device moves quickly. Second, the scheme requires that the legitimate device is always within the range of the wireless network; otherwise, it will not be able to prevent effectively when the attacker appears. We will continue to study and overcome these possible problems in the future work.

7. Conclusion

In this paper, we propose an enhancing Wi-Fi device authentication protocol. We also give a complete Wi-Fi device authentication framework. The framework mainly contains two parts, including the authentication in access phase and the authentication in association phase. We provide different authenticating algorithms for both authentication phases. What is more, we present an authenticator composed of small-scale autoencoders and the authenticator update method. The evaluation shows that our authentication methods in both authentication phases have good performance. Also, our authenticator has a higher time efficiency than a single neural network, which can help our framework to do packet-level authentication.

Data Availability

The data set in this paper includes CSI samples collected in two experiment scenarios, which is available from the corresponding author upon request. The code of our authentication framework is available in https://github.com/Chinmize/CSIAuthenticator.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

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

References

[1] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.

[2] S. Liu, "Mac spoofing attack detection based on physical layer characteristics in wireless networks," in *2019 IEEE International Conference on Computational Electromagnetics (ICCEM)*, pp. 1–3, Shanghai, China, 2019.

[3] P. Madani, N. Vlajic, and S. Sadeghpour, "Mac-layer spoofing detection and prevention in IoT systems: randomized moving target approach," in *Proceedings of the 2020 Joint Workshop on CPS & IoT Security and Privacy (CPSIOTSEC'20)*, pp. 71–80, New York, NY, USA, 2020.

[4] J. S. Alshudukhi, B. A. Mohammed, and Z. G. Al-Mekhlafi, "An efficient conditional privacy-preserving authentication scheme for the prevention of side-channel attacks in vehicular ad hoc networks," *IEEE Access*, vol. 8, pp. 226624–226636, 2020.

[5] Y. Song, Q. Huang, J. Yang, M. Fan, A. Hu, and Y. Jiang, "IoT device fingerprinting for relieving pressure in the access control," in *Proceedings of the ACM Turing Celebration Conference (ACM TURC'19)*, New York, NY, USA, 2019.

[6] C. Yan and J. Ge, "Synchronous control of master-slave manipulator system under deception attacks," in *2020 Chinese Control and Decision Conference (CCDC)*, pp. 1778–1782, Hefei, China, 2020.

[7] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in dwsns," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.

[8] L. Yang, Y. Song, S. Gao, B. Xiao, and A. Hu, "Griffin: an ensemble of autoencoders for anomaly traffic detection in SDN," in *2020 IEEE Global Communications Conference (GLOBECOM 2020)*, pp. 1–6, Taipei, Taiwan, 2020.

[9] U. Chatterjee, R. Sadhukhan, D. Mukhopadhyay, R. Subhra Chakraborty, D. M. Mahata, and M. Prabhu, "Stupify: a hardware countermeasure of kracks in wpa2 using physically unclonable functions," in *Proceedings of the 2020 Companion Proceedings of the Web Conference*, pp. 217–221, Taipei Taiwan, 2020.

[10] A. Elhigazi, S. Abd Razak, M. Hamdan, B. Mohammed, I. Abaker, and A. Elsafi, "Authentication flooding dos attack detection and prevention in 802.11," in *2020 IEEE Student Conference on Research and Development (SCOReD)*, pp. 325–329, Batu Pahat, Malaysia, 2020.

[11] S. Gao, Z. Peng, B. Xiao, A. Hu, Y. Song, and K. Ren, "Detection and mitigation of dos attacks in software defined networks," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1419–1433, 2020.

[12] W. Kim, S. Kim, and H. Lim, "Malicious data frame injection attack without seizing association in IEEE 802.11 wireless LANs," *IEEE Access*, vol. 9, pp. 16649–16660, 2021.

[13] 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)*, pp. 231–235, Dalian, China, 2019.

[14] Q. Wang, H. Li, D. Zhao, Z. Chen, S. Ye, and J. Cai, "Deep neural networks for CSI-based authentication," *IEEE Access*, vol. 7, pp. 123026–123034, 2019.

[15] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in iiot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.

[16] J. Xiong, M. Zhao, M. Z. A. Bhuiyan, L. Chen, and Y. Tian, "An ai-enabled threeparty game framework for guaranteed data privacy in mobile edge crowdsensing of iot," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2021.

[17] H. Liu, Y. Wang, J. Liu, J. Yang, and Y. Chen, "Practical user authentication leveraging channel state information (CSI)," in *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security (ASIA CCS'14)*, pp. 389–400, New York, NY, USA, 2014.

[18] M. Liu, A. Mukherjee, Z. Zhang, and X. Liu, "TBAS: enhancing Wi-Fi authentication by actively eliciting channel state information," in *2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 1–9, London, UK, 2016.

[19] 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.

[20] K. S. Germain and F. Kragh, "Multi-transmitter physical layer authentication using channel state information and deep learning," in *2020 14th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1–8, Adelaide, SA, Australia, 2020.

[21] K. S. 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)*, pp. 1–8, Adelaide, SA, Australia, 2020.

[22] J. Xiong, J. Ren, L. Chen et al., "Enhancing privacy and availability for data clustering in intelligent electrical service of IoT," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1530–1540, 2019.

[23] J. Zhang, Z. Wang, K. Wang, S. Guo, B. Wang, and M. Guo, "Improving power efficiency for online video streaming service: a self-adaptive approach," *IEEE Transactions on Sustainable Computing*, vol. 4, no. 3, pp. 308–313, 2019.

[24] C. Shi, J. Liu, H. Liu, and Y. Chen, "Smart user authentication through actuation of daily activities leveraging WiFi-enabled IoT," in *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc '17)*, pp. 1–10, New York, NY, USA, 2017.

[25] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network ingetrusion detection," *Network and Distributed System Security Symposium*, 2018, http://arxiv.org/abs/:1802.09089.