DOI:10.7544/issn1000-1239.2021.20210676
58(11):2374-2399,2021

基于混合特征指纹的无线设备身份识别方法

宋宇波[1,2,3] 陈 冰[1,2,3,4] 郑天宇[1,2,3] 陈宏远[1,2,3] 陈立全[1,2,3] 胡爱群[3,4]
[1](东南大学网络空间安全学院 南京 211189)
[2](江苏省计算机网络技术重点实验室(东南大学) 南京 210096)
[3](紫金山实验室 南京 211111)
[4](东南大学信息科学与工程学院 南京 211189)
(songyubo@seu.edu.cn)

Hybrid Feature Fingerprint-Based Wireless Device Identification

Song Yubo[1,2,3], Chen Bing[1,2,3,4], Zheng Tianyu[1,2,3], Chen Hongyuan[1,2,3], Chen Liquan[1,2,3], and Hu Aiqun[3,4]
[1](*School of Cyber Science and Engineering*, *Southeast University*, *Nanjing* 211189)
[2](*Key Laboratory of Computer Network Technology of Jiangsu Province* (*Southeast University*), *Nanjing* 210096)
[3](*Purple Mountain Laboratories*, *Nanjing* 211111)
[4](*School of Information Science and Engineering*, *Southeast University*, *Nanjing* 211189)

Abstract Wireless networks transmit data over open wireless channels, so they are vulnerable to impersonation attacks and information forgery attacks. To prevent such attacks, accurate device identification is required. The device identification technology based on channel state information (CSI) fingerprinting uses the wireless channel characteristics of device for identification. Since CSI can provide fine-grained channel characteristics and can be easily obtained from OFDM wireless devices, this technology has received wide attention. However, since CSI fingerprints identify the wireless channel characteristics of device, they change with the location or the environment of device. What's more, the existing technologies usually use machine learning for fingerprint matching for increasing identification accuracy, but the computational complexity of fingerprint matching increases, which in turn cannot be implemented in embedded devices with limited computational ability. To address these problems, this paper proposes a hybrid feature fingerprint-based device identification scheme, which includes the identification in access stage and communication stage. Packet arrival interval distribution (PAID) fingerprint, which is independent of device's location, is introduced for identification in access stage to compensate for the shortcomings of the CSI fingerprint. In communication stage, CSI fingerprints are extracted from each data packet and identified in real time with the feature that CSI can be acquired packet by packet. In addition, this paper proposes a fingerprint matching scheme with low computational complexity to ensure fast and accurate device identification even in devices with limited computational ability. We implement the identification system on Raspberry Pi and perform some experiments, which show that the identification accuracy is up to 98.17% and 98.7% in access stage and communication stage, and the identification time of a single packet in communication stage is only 0.142 ms.

收稿日期:2021-06-11;**修回日期**:2021-08-13
基金项目:国家重点研发计划项目(2020YFE0200600)
This work was supported by the National Key Research and Development Program of China (2020YFE0200600).
通信作者:宋宇波(songyubo@seu.edu.cn)

Key words wireless network security; wireless device identification; hybrid feature fingerprint; channel state information (CSI); autoencoder

摘 要 无线网络利用开放性的无线信道传输数据,因此容易遭受设备假冒攻击和通信数据伪造攻击,而防范此类攻击需要精准的设备识别.基于信道状态信息(channel state information, CSI)指纹的设备识别技术利用无线信道特征来识别设备.由于CSI提供细粒度的信道特征,并且可以从OFDM无线设备中轻松获取,因此该技术受到广泛的关注.但是反映无线信道特征的CSI指纹会随着终端的位置和所处环境的改变而改变,并且现有技术通常将机器学习用于指纹匹配以追求高识别准确率,随之而来的高计算复杂度使其无法在计算能力有限的嵌入式设备中实现.针对上述问题,提出了一种基于混合特征指纹的设备身份识别方法,包含终端接入时和通信时的设备识别.在接入时,引入了与终端外界因素无关的数据包到达时间间隔分布(packet arrival interval distribution, PAID)指纹进行识别,以弥补CSI指纹的缺陷;在通信时,借助CSI可以逐包获取的特点,从每个报文中提取CSI指纹并进行实时识别.同时,提出了一种计算复杂度较低的指纹匹配方案,以保证在计算能力有限的设备中也能快速且准确地识别终端.在树莓派上实现了设备识别原型系统并开展了实验,实验表明:该系统在接入时和通信时的识别准确率最高可达98.17%和98.7%,通信时单个数据包的识别时间仅需0.142 ms.

关键词 无线网络安全;无线设备身份识别;混合特征指纹;信道状态信息;自动编码器

中图法分类号 TP309

近年来,无线网络技术发展迅速,无线用户数量激增.随着无线网络在金融交易、商业管理等一系列重要业务上的广泛应用,无线网络安全成为安全研究的重点.由于无线网络使用开放性的无线信道传输数据,因此易遭受设备假冒攻击和通信数据伪造攻击.攻击者可以通过窃听无线传输数据来获取合法用户的身份信息,进而利用该身份信息假冒合法用户接入网络,或者通过发送携带合法用户身份信息的伪造数据来欺骗无线网络中的其他设备[1-3].IEEE 802.11i提供的基于密码学的无线设备认证技术一直是防范上述攻击的重要技术,但是已经出现有效的攻击方案[4-7].此外,基于密码学的无线设备认证技术需要无线终端进行复杂的密钥计算和密钥协商,一方面具有较大的时间开销,另一方面难以在计算能力有限的嵌入式设备中实现.

基于信道状态信息(channel state information, CSI)指纹的设备识别技术是一种非密码学的设备识别技术[8-18].这类技术从无线设备的CSI中提取指纹,该指纹能够标识设备的无线信道特征.CSI通常为无线信道的信道频率响应,在OFDM无线通信系统中,CSI由各个子载波的幅度和相位组成,能够在多频点上提供细粒度的无线信道状态信息.由于空间的多样性,每台设备的无线信道都是独一无二的,因此反映细粒度无线信道特征的CSI指纹能够用于标识设备身份.在设备识别过程中,识别设备获取待识别终端的CSI并生成CSI指纹,该CSI由识别设备通过信道估计获得,反映了设备的物理属性,因此CSI指纹难以被伪造,进而大大增加了设备假冒攻击和通信数据伪造攻击的难度.

重要的是,CSI可以从现有的支持OFDM无线通信技术的802.11设备(如802.11a/g/n/ac/ax设备)中直接获取,而无需定制特殊的设备或进行额外的信号处理,因此能够轻松地在现有的无线设备中部署基于CSI指纹的无线设备身份识别系统.同时,在机器学习算法的帮助下,这类技术往往能够获得较高的设备识别准确率.但是这类技术面临着2个问题:

1) 由于CSI指纹标识终端的无线信道特征,因此它会随着终端的位置或所处环境的改变而改变.也就是,现有的技术通常需要提前采集终端的大量CSI指纹以用于后续的指纹匹配,但是若无线终端的位置发生变化,或周围环境发生较大的改变,提前采集的CSI将无法代表最新的无线信道状态,进而无法再用于后续的设备身份识别.尤其当长期处于离线状态的无线终端请求接入网络时,其无线信道状态往往已经发生较大的变化,因此除非重新采集CSI指纹,否则后续的指纹匹配将无法成功进行.

2) 现有的技术通常将机器学习用于指纹匹配,以追求系统的高识别准确率,但是指纹匹配的计算复杂度也会随之增加,进而无法在计算能力有限的嵌入式设备中实现.也就是,现有的基于CSI指纹的

设备身份识别技术往往借助机器学习来进行特征提取、指纹分类以及指纹匹配等工作,进而获得较好的特征提取效果和高识别准确率.但是,CSI 指纹识别的计算复杂度也随之增大,因此可能会对设备硬件造成巨大的负担,同时无法在计算能力有限的嵌入式设备中实现.

以上 2 个问题给基于 CSI 指纹的无线设备身份识别技术的实际应用带来了困难.为了解决这些问题,本文提出了一种基于混合特征的无线设备指纹生成方案,该方案在使用 CSI 指纹的同时引入了数据包到达时间间隔分布(packet arrival interval distribution, PAID)指纹,该指纹能够标识设备的硬件特征.2 种指纹被分别用于两阶段的设备识别:一是终端请求接入时,利用 PAID 指纹进行识别,并重新采集终端的 CSI 指纹;二是终端成功接入并开始通信时,从终端的每个数据包中提取 CSI 指纹并进行实时的逐包设备识别.此外,本文提出了一种计算复杂度较低的指纹匹配方案,以保证在计算能力有限的设备中也能快速且准确地识别设备.本文的主要贡献有 3 个方面.

1) 针对 CSI 指纹会随着终端的位置或所处环境的改变而改变的问题,本文提出了一种基于混合特征指纹的设备身份识别方案.该方案从无线终端的 PAID 和 CSI 中提取 2 类指纹:前者为 PAID 指纹,标识设备的硬件特征;后者为 CSI 指纹,标识设备的无线信道特征.该方案包含两阶段的设备识别:①当终端请求接入无线网络时,识别设备捕获该终端发送的若干数据包,从中提取 PAID 指纹和 CSI 指纹,并用 PAID 指纹进行识别,识别成功则用新获取的 CSI 指纹进行后续的 CSI 指纹匹配;②当终端接入并开始通信时,识别设备从终端的每个数据包中提取 CSI 指纹并进行实时的逐包身份识别.

2) 针对现有的技术利用机器学习提高识别准确率的同时计算复杂度较高,无法在计算能力有限的嵌入式设备中实现的问题,本文提出了一种改进的指纹匹配方案,该方案利用基于自动编码器的指纹匹配网络来进行指纹匹配.该网络包括匹配层和输出层,前者由小规模自动编码器并联组成,后者由线性变换器构成,在保证高识别准确率的同时降低了计算复杂度.

3) 在树莓派 3B+上实现了基于混合特征指纹的设备识别原型系统,主要包括无线终端接入时的 PAID 指纹提取、PAID 指纹匹配、CSI 指纹采集,以及通信时的逐包 CSI 指纹提取和 CSI 指纹匹配.利用该原型系统在 2 种测试场景下分别进行了接入时和通信时的设备身份识别测试,并根据测试结果分析了设备识别系统的性能.

1 相关工作

无线通信技术的发展为人们的生产和生活带来了极大的便利.相比于有线网络,无线网络利用其无线传输的特点有效地联通了互联网的“最后一百米”,并且在公共服务、商业活动、金融交易、军事、医疗等重要领域中得到了充分的应用.但是,一些恶意攻击者的存在给无线网络的安全性造成了威胁.攻击者可以窃听和拦截无线流量并获取用户的身份信息,然后通过将自身伪装成合法设备来发起假冒攻击欺骗无线接入点(access point, AP),实现以合法身份接入网络.这意味着无线网络必须对未知用户进行精准的身份验证以防御这些可能存在的攻击,进而增强自身的安全性.

近年来,基于信道特征指纹的无线设备身份识别技术获得了越来越多的关注.在信道特征指纹发展的早期,接收信号强度指示(received signal strength indication, RSSI)被用于提取信道特征指纹[19-22].该技术的基本思想是:不同设备的信号发射端会根据其信道状态来设置信号的发射强度,因此 RSSI 指纹能够反映设备的信道状态.但是 RSSI 已被证实在较小的环境噪声干扰下会产生较大的波动,并不能作为稳定的设备身份标识[23].因此,人们尝试从信道冲激响应(channel impulse response, CIR)中提取用于标识设备的信道特征指纹.Mahmood 等人[24]提出了基于 CIR 指纹的分布式无线设备身份识别方法,该方法利用在不同的位置部署多个测量设备的方式来提高系统的识别准确率.此后,CSI 被用于生成信道特征指纹[8-18],这种指纹识别技术相比于 RSSI 指纹和 CIR 指纹具有重大的突破:首先,CSI 包含比 RSSI 和 CIR 更细粒度的信道状态信息,能够更好地标识设备的无线信道状态,即使 2 台设备具有近似的无线信道状态(如 2 台设备距离很近),CSI 也能捕捉设备之间的细微差异;其次,CSI 可以很容易地从支持 OFDM 无线通信技术的 802.11 设备中获得(如 802.11a/g/n/ac/ax 设备),而 CIR 的获取则依赖于特殊的设备[25].

Liao 等人[8]从 CSI 的幅度信息中提取指纹,并利用提前训练的 CNN 将待识别的 CSI 指纹分类,若该指纹的分类结果与其携带的身份信息一致,

并且输出超过设定的阈值,则判定该 CSI 指纹携带的身份信息为真.但是文献[8]并没有给出安全地获取 CNN 训练集的方法,恶意设备可能会用假冒的身份污染 CNN 训练集.为了解决该问题,Liu 等人[11]提供了一种安全的 CSI 指纹采集方式,该文使用 K-means 算法将采集的 CSI 指纹进行聚类,根据聚类结果判断指纹是否来自恶意终端.其同样提取 CSI 的幅度信息作为指纹,并使用 SVM 算法对待识别的 CSI 指纹进行分类,若分类结果与 CSI 携带的身份一致则识别成功.以上技术均从单收发天线的 CSI 中提取指纹,Xie 等人[12]将 MIMO 与基于 CSI 指纹的无线设备身份识别技术相结合,尝试使用更高维度的 MIMO-CSI 指纹来提高身份识别的准确性.Liu 等人[18]同样使用 MIMO-CSI 指纹进行设备识别,但是该文提出了一种基于 LOF 的改进设备识别算法,该识别算法即使在低信噪比无线通信环境下也可以达到高准确率.

文献[8-18]所述的方法虽然利用 CSI 指纹实现了高准确率的设备身份识别,但是没有考虑设备位置改变或环境改变对 CSI 指纹带来的影响.Liu 等人[11]虽然提供了一种在设备位置改变后重新采集 CSI 指纹的方法,但是这种方法安全有效的前提是合法设备必须能够随时响应识别设备的指纹采集请求,这在大部分应用场合中是无法满足的(如办公场景下,人们时常需要携带笔记本电脑离开).

为了解决待认证设备的移动和合法设备的离线给 CSI 指纹认证带来的问题,本文提出了一种基于混合特征指纹的设备识别技术.该技术在使用 CSI 指纹为无线终端提供实时的逐包设备识别服务的同时引入了另外一种指纹.该指纹从终端的 PAID 中提取,能够反映设备的硬件特征,进而与设备所处的位置和环境无关.但是,由于获取 PAID 指纹需要采集大量的数据包,因此无法用来实现实时的设备识别.因此,本文将 PAID 指纹用于接入时的设备识别,并将 CSI 指纹用于通信中实时的逐包设备识别.

2 相关技术介绍

本节介绍了基于混合特征指纹的无线设备识别技术所使用的相关技术,包括无线网络的端到端时延组成、数据包时间间隔的影响因素、CSI 的基本概念和测量方式,以及自动编码器的相关概念和算法.

2.1 数据包到达时间间隔

为了评估一个分组交换无线网络的数据速率和吞吐量,需要研究影响数据包递送速率的因素.在一个单跳分组交换无线网络中,用户通信时数据包的递送步骤可以概括为:

1) 发送设备组建数据包;
2) 发送设备将数据包发送到无线信道上;
3) 数据包从发送设备传输到路由器;
4) 路由器接收数据包并将其放入处理队列;
5) 路由器解析数据包并组建新数据包;
6) 路由器将新数据包发送到无线信道上;
7) 数据包从路由器传输到接收设备;
8) 接收设备接收并解析数据包.

其中,步骤 1,5,8 均为设备在本地处理数据包,因此这些步骤产生的时延可以归纳为处理时延;步骤 2,6 为设备将数据包发送到物理信道,将产生传输时延;步骤 3,7 为数据包在无线信道中传播,此时存在传播时延;步骤 4 为数据包在路由器中排队等待处理,可视为存在排队时延.

在由 2 台直接相连的无线终端组成的网络中利用 Ping 探讨数据包时间间隔的时延组成.在本场景中,无线终端 A 为 Ping 请求设备,无线终端 B 为 Ping 响应设备,图 1 显示了当 Ping 请求发送间隔设置为 10 ms 时无线终端 A 和无线终端 B 的 Ping 时序关系.本研究中发送间隔为 10 ms 的含义为:无线终端 A 提前组建 Ping Request 数据包,并在 10 ms 的中断处理程序中将已经组件好的请求包通过系统调用发送到无线信道中.如图 1 所示,每组(Ping Request,Ping Response)都被限制在 10 ms 以内,相邻 2 组之间没有时域上的交叠.这种限制是合乎常理的,因为通过实际测试可知,当两端设备中间无路由器、交换机等中转设备时,Ping 的往返时延(round-trip time, RTT)将小于 5 ms.

图 1 给出了 2 组(Ping Request,Ping Response)的时序关系,并分别编号为(Ping Request(1),Ping Response(1))和(Ping Request(2),Ping Response(2)).如图 1 所示,无线终端 A 在发送时间节点 0 ms,10 ms,…上将数据包传输到信道中,该传输时延记作 $t_{A\text{-trans}}$;数据包从无线终端 A 传播到无线终端 B,其传播时延记作 $t_{A\text{-prop}}$;无线终端 B 接收、解析 Ping Request 并组建、传输 Ping Response 产生的时延包括处理时延和传输时延,记作 $t_{B\text{-proc}}+t_{B\text{-trans}}$;Ping Response 从无线终端 B 传播到无线终端 A 产生的传播时延记作 $t_{B\text{-prop}}$.

为了计算数据包接收时间间隔,将 2 组 Ping 数据包的时延分别用(1),(2)进行编号,因此数据包

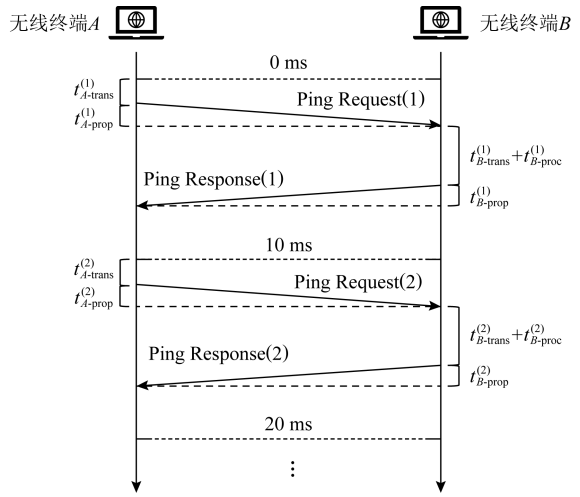


Fig. 1 Ping timing diagram and delay composition

图 1 Ping 时序图与时延组成

接收时间间隔 t 可计算为:

$$t^1=t^{(1)}_{A\text{-trans}}+t^{(1)}_{A\text{-prop}}+t^{(1)}_{B\text{-trans}}+t^{(1)}_{B\text{-proc}}+t^{(1)}_{B\text{-prop}}, \quad (1)$$

$$t^2=t^{(2)}_{A\text{-trans}}+t^{(2)}_{A\text{-prop}}+t^{(2)}_{B\text{-trans}}+t^{(2)}_{B\text{-proc}}+t^{(2)}_{B\text{-prop}}. \quad (2)$$

若令 $\Delta t_{A\text{-trans}}=t^{(2)}_{A\text{-trans}}-t^{(1)}_{A\text{-trans}}$ 并以此类推,则 t 还可以表示为

$$t=10+\Delta t_{A\text{-trans}}+\Delta t_{A\text{-prop}}+\Delta t_{B\text{-trans}}+\Delta t_{B\text{-prop}}+\Delta t_{B\text{-proc}}. \quad (3)$$

式(3)还可以被进一步简化.首先,普通商用无线设备的数据包的传输时延通常为毫秒到微妙级,而数据包的处理时延仅为微秒级甚至更小的数量级,因此 $t_{B\text{-proc}}\ll t_{B\text{-trans}}$;再者,无线传输的距离通常为 100 m 左右甚至更小,因此数据包的传播时延将为微秒级甚至更小的数量级,因此 $t_{B\text{-prop}}\ll t_{B\text{-trans}}$.相应地,由于设备软硬件因素和无线信道争用造成的处理、传播时延抖动($\Delta t_{B\text{-proc}}$,$\Delta t_{B\text{-prop}}$)也会远小于传输时延抖动 $\Delta t_{B\text{-trans}}$.由此,数据包接收时间间隔 t 可以被简化为

$$t=10+\Delta t_{A\text{-trans}}+\Delta t_{B\text{-trans}}. \quad (4)$$

因此,t 的主要影响因素为发送设备和接收设备的传输时延.

2.2 信道状态信息

在无线通信中,发射机发出的无线信号将沿着多条无线路径传播到接收机,这些路径组成了用于设备通信的无线多径信道.CSI 通常指该无线多径信道的信道频率响应(channel frequency response, CFR),因此当传输频率为 f 时,CSI 可以表示为

$$H(f)=\sum_{n=1}^{N}a_n e^{-j2\pi f\tau_n}. \quad (5)$$

由式(5)可知,CSI 反映了无线信道整体的幅度衰减和相位偏移.在实际无线通信中,组成无线信道的每条传播路径都具有不同的幅度衰减和时间延迟.当无线信号沿着不同的路径抵达接收机时,不同幅度和相位的信号叠加组成了最终的接收信号,CSI 则反映了多条传播路径叠加后的幅度衰减和相位偏移.

基于 OFDM 的无线网络(如 IEEE 802.11a/g/n)利用多载波传输技巧实现频带的高效利用和系统吞吐量的提升.当设备在 2.4 GHz 频段上以 20 MHz 的带宽传输数据时,其所使用的信道将包含 64 个子载波.这种情况下,接收端测量得到的 CSI 包含每个子载波的频率响应,进而组成 64 维的 CSI 复数向量.在 MIMO-OFDM 无线通信系统(假设该系统具有 V 根发射天线和 G 根接收天线)中,由发射天线和接收天线组成的 $V\times G$ 个天线对各有一个 CSI,进而将 CSI 拓展为 $V\times G\times 64$ 的 3 维复数矩阵.

相比于 RSSI,CIR 用来表征无线信道物理特性的物理量,CSI 借助 OFDM 多载波传输的特点,在多个频点上记录信道的幅度衰减和频率偏移,因此包含了更加细粒度的无线信道信息.同时,基于 OFDM 的 IEEE 802.11a/g/n 无线设备在信道估计模块提供 CSI 的测量和提取,其根据 CSI 测量值来优化接收器参数,使得接收机适应当前的信道状态.因此,能够从现有的商用 IEEE 802.11a/g/n 无线网卡中直接获取 CSI 测量值,无需添加新固件或定制的硬件.

OFDM 无线通信系统利用信道估计来获取 CSI,而信道估计则依赖于无限物理层 PLCP 子层协议数据单元(presentation protocol data unit, PPDU)的长训练序列.在采用 OFDM 技术的 IEEE 802.11 协议(如 IEEE 802.11 a/g/n/ac/ax)中,物理层包含物理层汇聚过程子层(physical layer convergence procedure, PLCP)和物理介质相关子层(physical medium dependent, PMD),其负责将介质访问控制层(medium access control, MAC)发来的数据发送到无线介质.该过程可以概括为:PLCP 接收到 MAC 发来的 MAC 层业务数据单元后,将其封装为 MAC 层协议数据单元,并作为 PLCP 子层业务数据单元发送到 PLCP 子层,PLCP 子层将其封装为 PPDU.

以工作在 2.4 GHz 频段、带宽为 20 MHz 的无线通信系统为例,PPDU 的帧格式如图 2 所示,其中,PLCP 前导码包含发送端和接收端共享的长训练序列 L,L 在 64 个子载波上的幅度为:L 在编号为

$-32\sim-27$,0,$27\sim31$ 的子载波上幅度为 0,在其余子载波上的幅度为 1.当发送信号为 X、接收信号为 Y 时,有:

$$Y=HX+N, \tag{6}$$

其中,H 为信道频率响应,N 为噪声.当忽略噪声 N 时,信道频率响应可估计为

$$H=YX^{-1}. \tag{7}$$

通过该信道估计方法,接收端利用长训练序列 L 可以获取无线信道的 CSI.需注意的是,L 仅在 52 个子载波上的幅度不为 0,CSI 的有效值分布在 L 的幅度不为 0 的子载波上.

PLCP前导码 12符号	SIGNAL 单OFDM符号	DATA 若干OFDM符号

Fig. 2 Frame structure of PPDU

图 2 PPDU 的帧结构

2.3 自动编码器

自动编码器作为一种特殊的人工神经网络,其学习目标为寻找一种数据映射,该映射首先将输入样本压缩得到低维度样本,然后以尽可能低的误差将低维样本重建为原始维度的样本.

该数据映射分为编码映射和解码映射.其中,编码映射负责输入样本的降维,解码映射负责将编码得到的低维样本重建为与输入样本维度相同的输出样本.降维后的样本具有比输入样本更少的特征,其携带的信息量通常少于输入样本所携带的信息量.为了根据低维样本重建出高维样本,自动编码器需要从训练样本中学习样本的分布特征.这种重建方式类似于大脑根据已掌握的不完整信息和经验来预测未知的结果.

本文所用的自动编码器为具有单一隐藏层的 3 层自动编码器,图 3 展示了一个典型的单隐藏层自动编码器结构.由图 3 可知,自动编码器的每一层

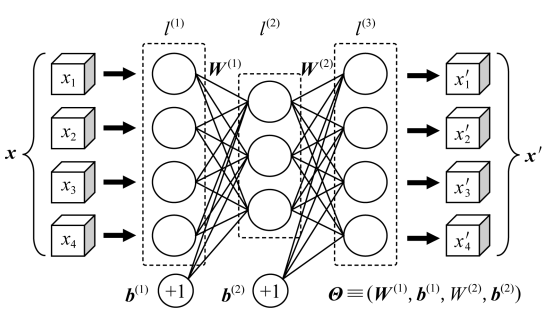


Fig. 3 The structure of autoencoder

图 3 自动编码器结构示意图

由若干个神经元构成,相邻 2 层的神经元通过突触两两连接.图 3 所示的自动编码器输入维度为 4,输出维度与输入维度一致.其具有一个隐藏层,该隐藏层对输入向量的压缩比为 4∶3.为了后续表示的方便,分别用 $l^{(1)},l^{(2)},l^{(3)}$ 来表示自动编码器的输入层、隐藏层、输出层,并用 $dim^{(1)},dim^{(2)},dim^{(3)}$ 表示 $l^{(1)},l^{(2)},l^{(3)}$ 的神经元数量($dim^{(1)}=dim^{(3)}$).自动编码器的压缩率 ρ 定义为

$$\rho=\frac{dim^{(2)}}{dim^{(1)}}. \tag{8}$$

此外,$\boldsymbol{W}^{(1)}$ 和 $\boldsymbol{W}^{(2)}$ 分别表示连接(输入层,隐藏层)和(隐藏层,输出层)的突触的权重.其中,$\boldsymbol{W}^{(1)}$ 为 $dim^{(1)}\times dim^{(2)}$ 的矩阵,$\boldsymbol{W}^{(2)}$ 为 $dim^{(2)}\times dim^{(3)}$ 的矩阵.$\boldsymbol{b}^{(1)}$ 和 $\boldsymbol{b}^{(2)}$ 分别表示隐藏层和输出层的偏差向量,它们的维度分别为 $dim^{(2)}$ 和 $dim^{(3)}$.

自编码器是数据相关的,具体表现为:使用特定样本集训练的自动编码器掌握了该样本集的分布特征,因此对于具有相同分布特征的样本具有良好的重建能力;但是对于具有不同分布特征的样本,该自动编码器的重建能力将很差.用输入样本和输出样本的均方根误差来表示自动编码器对该输入样本的重建能力:

$$RMSE(\boldsymbol{x},\boldsymbol{x}')=\left(\frac{1}{4}\sum_{i=1}^{4}(x_i-x'_i)^2\right)^{\frac{1}{2}}. \tag{9}$$

在本文中,利用自动编码器的数据相关特性对无线设备的 CSI 指纹进行标记和识别.具体方式为:首先利用目标设备的 CSI 指纹集训练自动编码器,然后将新的属于未知设备的 CSI 指纹输入自动编码器,最后获得该 CSI 指纹的重建误差.若重建误差大于设置好的阈值,则该 CSI 指纹被判定为不具有训练集特征,进而不属于目标设备.

自动编码器的执行过程为输入向量在神经网络中逐层传递的过程,这与人类的末梢神经接收信号并沿着神经网络将其传递到深层神经类似,将其称为神经网络的前向传播过程.假设输入向量为 $\boldsymbol{x}$,则 $l^{(2)}$ 接收到的向量为 $\boldsymbol{a}^{(2)}=g(\boldsymbol{W}^{(1)}\cdot\boldsymbol{x}+\boldsymbol{b}^{(1)})$ 接收到的向量为 $\boldsymbol{a}^{(3)}=\boldsymbol{x}'=g(\boldsymbol{W}^{(2)}\cdot\boldsymbol{a}^{(2)}+\boldsymbol{b}^{(2)})$.其中,$g$ 为激活函数.本文所用的激活函数为 sigmoid 函数,因此 g 定义为

$$g(\boldsymbol{x})=\frac{1}{1+\mathrm{e}^{\boldsymbol{x}}}. \tag{10}$$

通过使用激活函数,每一层接收到的信号均为上层信号的非线性组合,因此整个神经网络可以拟合出复杂的非线性模型.若不使用激活函数,则整个

前向传播过程等价于将输入向量进行线性组合，因此失去了拟合复杂模型的能力.将整个前向传播过程表示为

$$h_{\boldsymbol{\Theta}}(\boldsymbol{x})=\boldsymbol{x}', \tag{11}$$

其中，$\boldsymbol{\Theta}\equiv(\boldsymbol{W}^{(1)},\boldsymbol{b}^{(1)},\boldsymbol{W}^{(2)},\boldsymbol{b}^{(2)})$，在下文中用于代指自动编码器.

在自动编码器的训练过程中，初始的自动编码器被赋予随机的权重，并利用训练集中的样本进行权重的更新.其中，用于训练自动编码器的训练集由具有相同分布特征的样本组成.对于本文所用的对称自动编码器，可以在编码映射和解码映射 2 个部分使用镜像的权重矩阵，因此可以将权重矩阵简化为 $\boldsymbol{W}^{(1)}=\boldsymbol{W},\boldsymbol{W}^{(2)}=\boldsymbol{W}^{\mathrm{T}}$.

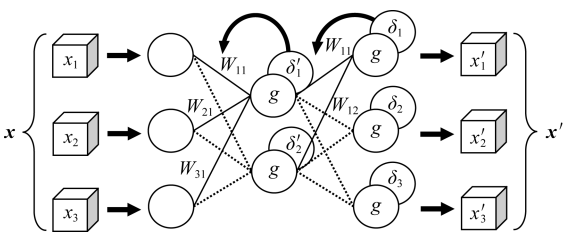


Fig. 4 The backward propagation process of the autoencoder

图 4 自动编码器的后向传播过程示意图

自动编码器的训练依据为所有神经元的激活误差 $deltas=[\delta_{ij}]_{dim^{(1)}\times dim^{(2)}}$，而获得 $deltas$ 的通用方法为反向传播算法.为了清晰地展示反向传播算法的计算过程，提供了一个 $dim^{(1)}=dim^{(3)}=3,dim^{(2)}=2$ 的单隐藏层自动编码器，并在图 4 中绘制了后向传播过程.由图 4 可知，$deltas$ 由隐藏层、输出层中所有神经元的激励误差组成，其计算方式为

$$\delta_i=x_i-x_i', \tag{12}$$

$$\delta_j'=\sum_{i=1}^{dim^{(1)}}W_{ij}\delta_j, \tag{13}$$

其中，$W_{ij}(i=1,2,\cdots,dim^{(1)};j=1,2,\cdots,dim^{(2)})$ 为权重矩阵 $\boldsymbol{W}$ 的第 i 行、第 j 列元素.由式(12)(13)可知，每一层的误差均由下一层神经元的激励误差反向加权得来，因此该计算过程称为反向传播.

本文选择使用随机梯度下降算法(stochastic gradient descent, SGD)作为自动编码器的训练算法.原始的梯度下降算法(gradient descent, GD)在每次更新权重时需要使用所有训练样本计算下降梯度，SGD 仅使用单个样本，相比于 GD 具有更快的收敛速度.本文提出的设备认证系统需要在通信时为设备提供数据包级别的认证服务，对神经网络的训练速度要求达到微秒级，因此选用更为迅速的 SGD 算法.在利用 SGD 算法更新权重之前，首先需要利用式(12)(13)计算出 $deltas$，然后利用式(14)更新权重：

$$W_{ij}'=W_{ij}+\varepsilon\delta_j'^{\,a_j^{(2)}(1-a_j^{(2)})x_i}+\varepsilon\delta_i x_i'^{\,(1-x_i')a_j^{(2)}}, \tag{14}$$

其中，ε 为自动编码器的学习率.式(14)的计算利用了 sigmoid 函数求导公式：

$$\frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}=\boldsymbol{x}(1-\boldsymbol{x}). \tag{15}$$

3 基于混合特征的无线设备指纹生成方案

本节描述了基于混合特征的无线设备指纹生成方案，包括 PAID 指纹和 CSI 指纹的生成方案，并利用理论分析和实验测试 2 种方式验证了 PAID 指纹和 CSI 指纹的有效性.

3.1 PAID 指纹生成方案

为了获取待识别无线终端的数据包到达时间间隔，终端需要以恒定速率 Q pkts/s 发送无线数据包，每个数据包都应当携带发送设备的身份信息(如 MAC，IP 等)，便于识别设备筛选.处于监听模式的识别设备接收到来自该终端的数据包 P_i(i 为接收数据包的编号)时，记录下数据包的到达时间 t_i，并将其添加在数据包到达时间序列 $PA=(t_1,t_2,\cdots,t_{i-1})$ 的末尾.当识别设备接收到 $L+1$ 个来自该终端的数据包 $P_1,P_2,\cdots,P_{L+1}$ 后，利用 $PA=(t_1,t_2,\cdots,t_{L+1})$ 计算数据包到达时间间隔序列 $PAI=(\Delta t_1,\Delta t_2,\cdots,\Delta t_L)$.其中，$\Delta t_i(i=1,2,\cdots,L)$ 的计算方式为

$$\Delta t_i=t_{i+1}-t_i. \tag{16}$$

随后，通过计算 PAI 的分段密度来获取其分布特征.PAI 的分段密度的计算方法为：将时间区间 $[t_{\mathrm{beg}},t_{\mathrm{end}}]$ 平均分为 SEG 个子时间区间($[t_{\mathrm{beg}},t_{\mathrm{end1}}]$，$(t_{\mathrm{end2}},t_{\mathrm{end3}}]$，$\cdots$，$(t_{\mathrm{end}(SEG-1)},t_{\mathrm{end}}]$).计算 PAI 中的时间间隔落在每个子区间中的数量，记为 $PN=(pn_1,pn_2,\cdots,pn_{SEG})$.令 pn 为 PAI 中的时间间隔落在时间区间 $[t_{\mathrm{beg}},t_{\mathrm{end}}]$ 中的总数，则分段密度计算方法为

$$D_{PAI}=\left(\frac{pn_1}{pn},\frac{pn_2}{pn},\cdots,\frac{pn_{SEG}}{pn}\right). \tag{17}$$

该 SEG 维序列即为 PAID 指纹 $PFP=(h_1,h_2,\cdots,h_{SEG})$.

3.2 CSI 指纹生成方案

为了方案描述的统一性，以带宽为 20 MHz、使用 64 个 OFDM 子载波进行数据传输的无线网络和具有单收发天线的无线终端进行讲解.接收机接收到无线终端发来的数据包后利用信道估计算法获取

CSI(见 2.2 节),该 CSI 是一个长度为 64 的复向量,表示为 $\boldsymbol{M}=(m_1,m_2,\cdots,m_{64})$.由 2.2 节可知,用于信道估计的长训练序列在编号为 $-32,-31,-30,-29,-28,-27,0,27,28,29,30,31$ 的 12 个子载波上幅度为 0,因此在生成 CSI 指纹前需将这些值剔除.剔除后获得长度为 52 的复向量 $\boldsymbol{U}=(u_1,u_2,\cdots,u_{52})$,每个值代表了相应子载波的幅度和相位.

CSI 的相位信息需要经过特殊处理才能更加稳定,这种处理将会增加认证系统的计算复杂度.相比之下,未经处理的 CSI 的幅度已经具有较高的稳定性和设备差异(见 3.4 节).出于降低设备识别方案的数据处理复杂度、提高设备识别时间效率的需求,本文将使用 CSI 的幅度来生成设备的 CSI 指纹.将 CSI 幅度序列表示为 $A_{\mathrm{CSI}}=(a_1,a_2,\cdots,a_{52})$,则 $a_i=|u_i|(i=1,2,\cdots,52)$.

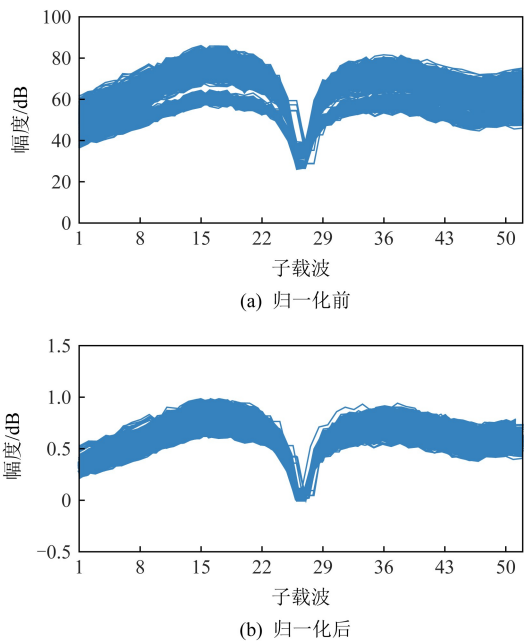


Fig. 5 CSI fingerprints before and after normalization

图 5 归一化前后的 CSI 指纹

图 5(a)展示了从受人员走动干扰的终端采集的 10 000 个 CSI 幅度序列 A_{CSI}.可以看到,由于存在环境干扰,该终端的 CSI 幅度在测量时间内存在较为明显的变化:A_{CSI} 发生整体上下偏移.但是尽管如此,所有的 A_{CSI} 仍然具有相近的形状.为了削弱幅度变化对指纹识别性能带来的负面影响,需要提取 A_{CSI} 的形状信息.该形状信息通过将每个 CSI 幅度序列进行最大-最小值归一化来获取.对于 A_{CSI},令 $a_{\max}=\max\{a_1,a_2,\cdots,a_{52}\}$,$a_{\min}=\min\{a_1,a_2,\cdots,a_{52}\}$,则最大-最小值归一化方法为

$$c_i=\frac{a_i-a_{\min}}{a_{\max}-a_{\min}}. \tag{18}$$

归一化后的 CSI 幅度序列即为 CSI 指纹 $CFP=(c_1,c_2,\cdots,c_{52})$.图 5(b)展示了图 5(a)中的 CSI 幅度序列归一化后得到的 CSI 指纹,可以看出,CSI 指纹仅保留了 CSI 幅度序列的形状信息,相比于未经处理的 CSI 幅度序列具有更小的离散度.

3.3 PAID 指纹有效性分析

本节将分析 PAID 指纹的有效性,即 PAID 指纹确实能够标识设备的硬件特征,并用于设备身份识别.

设备利用无线网卡发送数据包是一个复杂的过程.当网络层构建好一个数据包后,CPU 在主存储器(dynamic random access memory,DRAM)中创建缓冲区描述符,该描述符包含数据包在存储器中的存储地址以及数据的长度(当数据包存储在多个不连续的虚拟内存块时,CPU 需要创建多个缓冲区描述符).然后,CPU 将新建的缓冲区描述符信息通过外部数据总线、北桥芯片、PCI 总线写入网络接口卡(network interface card, NIC)的内存映射寄存器.NIC 检测到新数据包到来后,启动若干个直接内存访问(direct memory access, DMA)检索描述符并读取数据包,这些数据仍然通过外部数据总线、北桥芯片、PCI 总线传输.最终,NIC 通过 MAC 单元发送数据包.

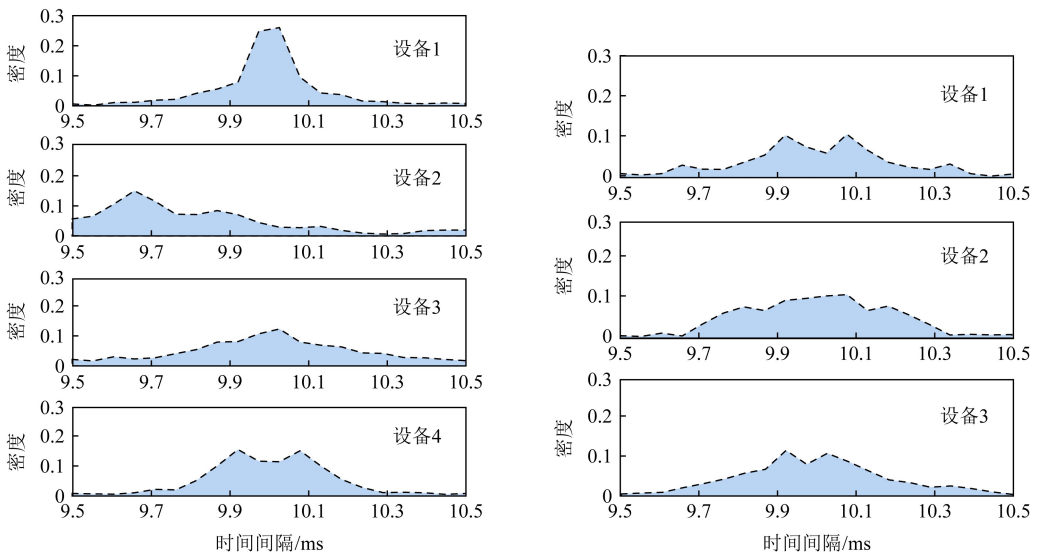
由上述过程可以看出,数据包的发送主要依赖于 CPU、DRAM、L1/L2 Cache、外部数据总线、北桥芯片、PCI 总线、NIC、DMA 控制器、MAC 单元.进一步地,这些硬件是影响利用无线网卡发送数据包所需时间的主要因素,而该时间正是数据包的传输时延(见 2.1 节).不同型号的设备将使用不同的硬件,因此传输时延能够用来反映不同型号的设备之间的差异.此外,即使是型号相同的设备,硬件的电气特性也会因为生产误差而不同,这会导致不同的设备具有不同的时钟偏移,因此数据包的传输时延也能够用来反映相同型号的设备之间的差异.由 2.1 节的讨论可知,数据包到达时间间隔主要由收发设备的数据包传输时延组成,因此数据包达到时间间隔也能够反映设备硬件的差异.

重要的是,由于操作系统调度算法和无线信道退避算法的随机性,单个数据包达到时间间隔也具有随机性.可以将数据包到达时间间隔看作服从某种分布的随机数,虽然单个数值受软件影响具有随机性,但是整体分布特征由设备硬件特征决定.因此,

可以利用大量数据包时间间隔的分布特征来生成PAID指纹.分布特征通常包含差异性特征(如极差、标准偏差和方差)、规律性特征(如算术平均值、中位数、众数)和概率密度.前两者从单一维度描述分布特征,概率密度则能够完全描述分布情况.因此,分段密度作为一种概率密度的粗粒度近似计算方式,能够很好地反映数据包到达时间间隔的分布特征,进而标识设备的硬件特征.

以上通过理论分析证明了基于数据包时间间隔分布的PAID指纹的有效性,下面从2个方面结合实际测试结果来证实PAID指纹的有效性.

1) 基于数据包时间间隔分布的指纹能够标识设备的硬件:为了证实这一点,从2类测试设备集提取了PAID指纹.第1类设备集包含4台不同型号的设备,第2类设备集则包含3台同型号的设备.为了更清晰地看出设备间指纹的差异,用折线图来绘制指纹.图6(a)展示了第1类设备集的PAID指纹,并标有每个指纹对应的设备型号.可以看出,除了设备2的数据包达到时间间隔集中分布在9.65 ms左右外,其余3台设备均集中分布在10 ms左右,这表明规律性特征无法反映设备硬件差异;此外,设备2与设备3的数据包时间间隔样本都较为离散,因此差异性特征无法反映设备硬件差异.对比之下,用分段密度表示的粗粒度概率密度则能够很好地反映此差异.图6(b)展示了第2类设备集的PAID指纹,可以看出,虽然该组设备集的PAID指纹差异较小,但是仍具有不同的分布趋势,借助机器学习算法可以感知同型号设备之间的差异并进行有效的身份识别.



(a) 不同型号 (b) 相同型号

Fig. 6 PAID fingerprints for different devices

图6 不同设备的PAID指纹

2) 基于数据包时间间隔分布的指纹与设备所处的物理环境无关:为了证实这一点,利用图6(a)所示的4台设备进行了测试.测试时,每台设备被分别放置在4个不同的位置,图7显示了4台设备在4个不同位置上的PAID指纹.可以看出,即使设备的位置发生了改变,设备的PAID指纹依旧相似,而不同的设备之间则具有很大的差异.这表明PAID指纹与设备所处的物理环境无关.

3.4 CSI指纹有效性分析

本节将分析CSI指纹的有效性,即CSI指纹确实能够标识设备的无线信道特征,并用于设备身份识别.

在无线设备工作时,设备的物理位置决定了无线信道的多径环境.这意味着,处于不同物理位置的设备将具有不同的多径环境.如2.2节所示,CSI反映了通过多条无线路径传输的无线信号后在接收机端叠加后的整体幅度衰减和相位偏移,所以CSI刻画了无线信道的多径效应.当接收机被放置在固定位置时,每个发射机与接收机之间的无线信道都是独特且相对稳定的,因此能够通过匹配CSI的方式来验证发射机的身份.

上一段通过理论分析证明了CSI指纹的有效性,下面从2个方面结合实际测试结果来证实CSI指纹的有效性.

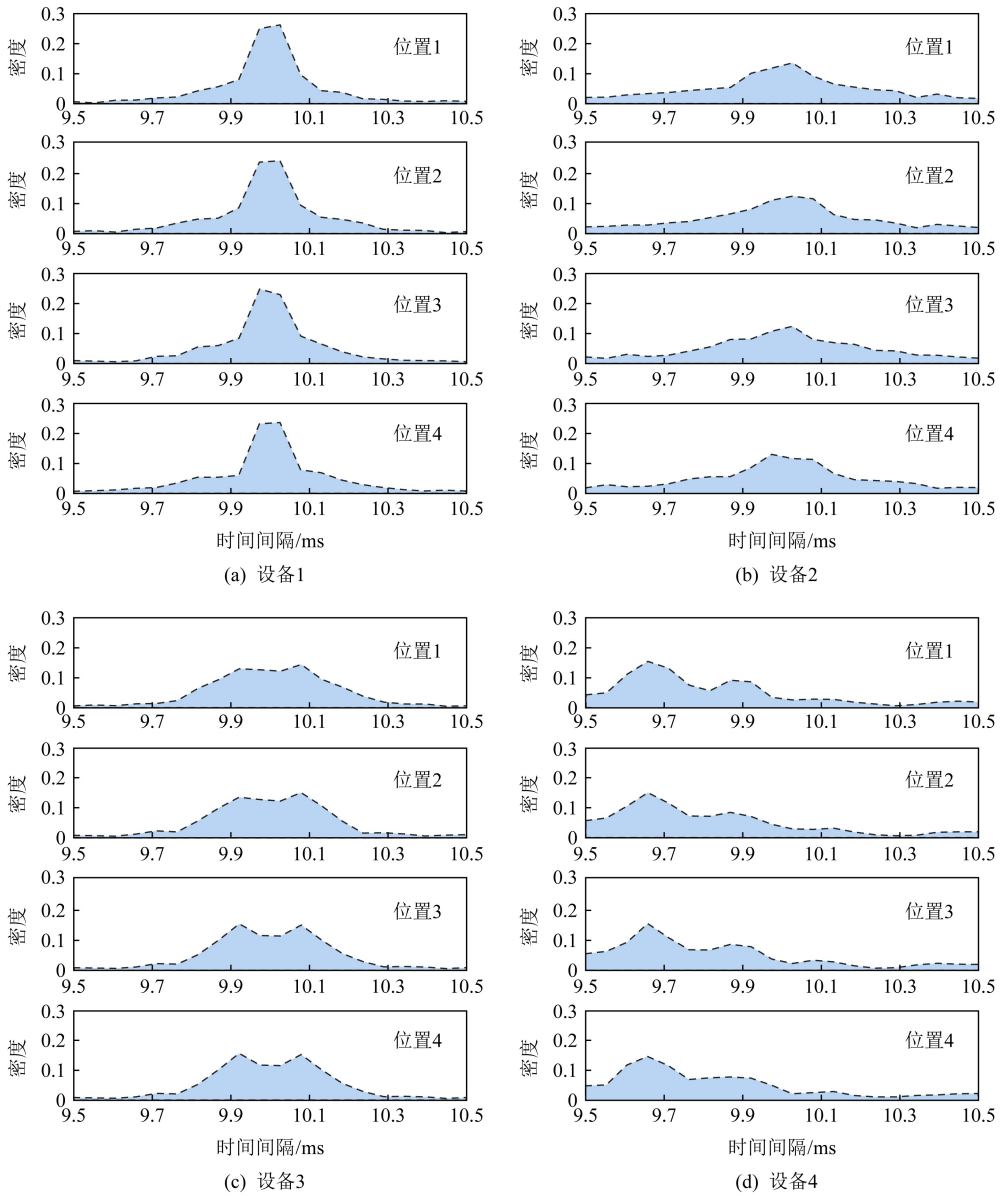


Fig. 7 PAID fingerprints collected at different locations

图 7 设备在不同位置处采集的 PAID 指纹对比

1) CSI 指纹能够标识设备的无线信道,进而能够用于设备身份识别:图 8 展示了 4 台放置在室内不同位置的无线终端的 CSI 幅度曲线,每张子图均包含 1 000 个 CSI 样本.由于测试场地面积较小,因此在采集 CSI 的过程中,终端的两两距离最小仅有 0.5 m 左右.可以看到,即使终端间的距离较小,4 台终端的 CSI 幅度曲线仍然具有肉眼可见的形状差异.这种差异存在的根本原因为不同的终端使用不同的无线信道来传输数据,而且即使终端位置相邻,其无线信道仍然具有很大的差异,这些差异体现在 CSI 幅度曲线中,因此 CSI 指纹能够作为设备的信道标识,进而实现设备身份识别.

2) 处于干扰环境中的固定无线设备的 CSI 指纹具有稳定性:在与图 8 相同的 CSI 采集场景中,无线终端和监听设备被放置在固定的位置,在 3 种干扰条件下进行了时长 1 min 的 CSI 采集,并绘制了 CSI 幅度曲线,如图 9 所示.3 种干扰分别为人员在距终端 1 m,2 m,3 m 远的位置为中心来回走动(不经过无线终端与监听设备的视距路径),CSI 测量结果分别如图 9(a)~(c)所示.由图 9 可知,当无线终端受到最大程度的干扰(第 1 种干扰条件)时,在不同时间测得的 CSI 幅度曲线依然保持近似的形状,只是所有子载波的幅度统一地增大或减小,这种差异可以通过指纹提取时的归一化来削弱.

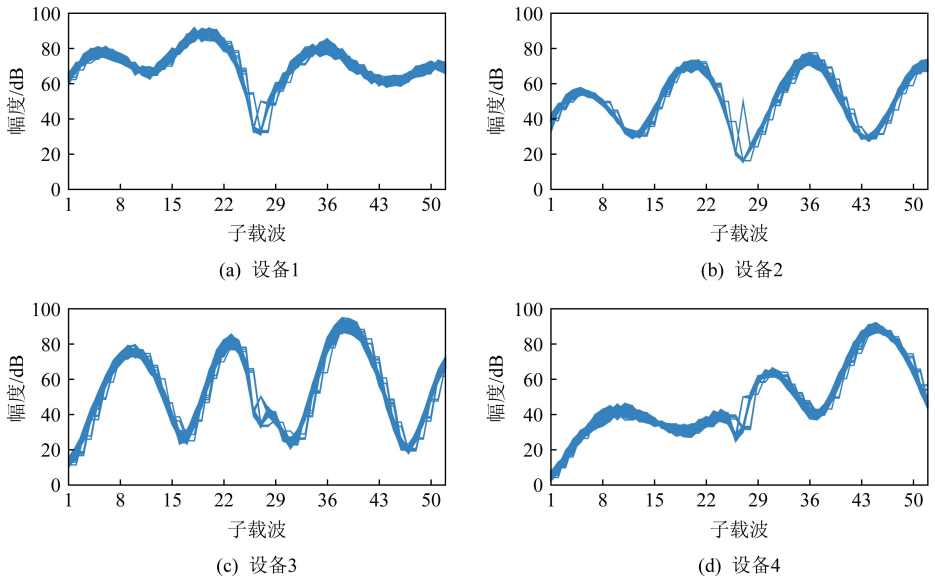


Fig. 8 CSI amplitude waveforms of different devices

图 8 不同设备的 CSI 幅度波形

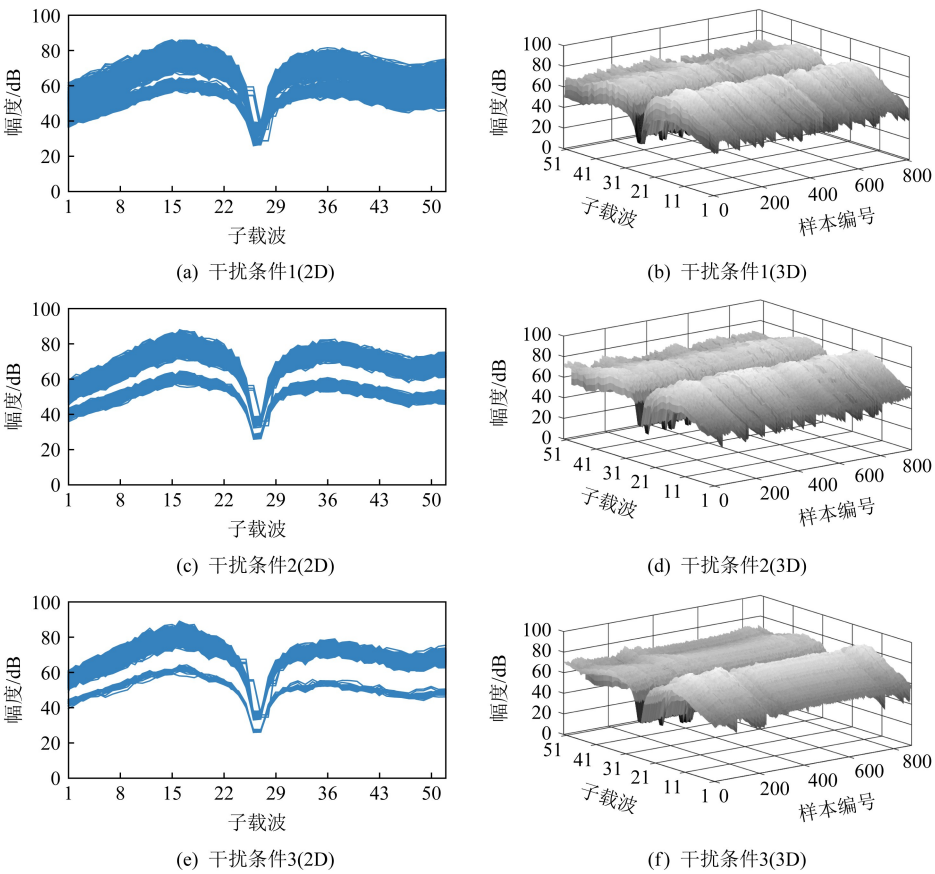


Fig. 9 CSI amplitude waveforms under different interference

图 9 不同干扰条件下的 CSI 幅度波形

4 基于混合特征指纹的无线设备识别方案

本节主要介绍基于混合特征指纹的无线设备识别方案,包括识别方案的基本设计思想、设备识别方案的整体流程.

4.1 方案设计思想与流程总览

本文的无线设备识别方案主要包括 2 个阶段的

设备身份识别:

1) 无线终端接入时的设备识别.该阶段发生在处于离线状态的无线终端发出请求接入的数据报文、识别设备接收到该报文时,此时识别设备需要获取该无线终端的设备身份信息并进行身份识别,若身份识别为合法则允许该终端接入网络.

2) 无线终端通信时的设备识别.该阶段发生在无线终端成功接入无线网络后、开始不断地发送数据包时,此时识别设备从每一个来自该终端的数据包中提取身份信息并进行身份识别,若身份识别合法则判定该数据包确实来自合法终端并接受之,否则判定该数据包来自非法的终端并丢弃之.

总而言之,本文的设备识别方案包括接入时和通信时的指纹获取与身份识别.第 3 节讨论了 PAID 指纹和 CSI 指纹,这 2 种指纹具有 2 个重要的区别:

1) CSI 指纹反映了设备的无线信道特征,PAID 指纹则反映了设备的硬件特征.CSI 指纹标记了无线终端的无线信道,因此即使是同一台终端,当该终端所处的环境改变时,其 CSI 指纹也会发生变化;与此不同的是,PAID 指纹标记了设备的硬件特性,无论设备所处的环境如何变化,PAID 指纹都将保持稳定.

2) CSI 指纹可以从单个数据包中提取,PAID 指纹则需要从大量数据包中提取.首先,由 2.2 节可知,每个 PPDU 帧都包含一个完整的 CSI,而单个数据包通常由一个或多个 PPDU 帧传输,因此 CSI 指纹可以从单个数据包中提取;相比之下,PAID 是大量数据包的到达时间间隔的分段密度,因此为了获取 PAID 指纹,必须接收来自指定无线终端的大量数据包.

当长期处于离线状态的无线终端重新请求接入无线网络时,其位置往往已经改变.此时,识别设备提前采集的用于指纹匹配的 CSI 指纹已经不能反映当前的信道状态.因此,接入时的设备身份识别应使用与外界因素无关的 PAID 指纹而非 CSI 指纹.在采集 PAID 指纹的过程中,需要接收来自待识别终端的大量数据包,如果接入时设备识别成功,则从这些数据包中提取的 CSI 指纹可以被用于通信时设备识别的指纹匹配,因为这些 CSI 指纹反映了最新的信道状态.若不利用 PAID 指纹进行接入时设备识别,而是直接重新采集终端的 CSI 指纹再进行接入时设备识别,则无法确定该无线终端是否伪造了身份.因为如果非法终端利用伪造的合法身份请求接入,识别设备就会采集到来自非法终端的指纹,识别设备并没有判断新采集的 CSI 指纹是否来自合法终端的能力.当采用了 PAID 指纹时,若 PAID 指纹匹配成功,则表明这些数据包来自合法终端.

当无线终端成功接入并开始通信时,应采用 CSI 指纹而非 PAID 指纹.因为 CSI 可以从单个数据包中提取,因此相比于 PAID 指纹,CSI 指纹可以用于在终端通信时实现实时的逐包设备识别,并及时发现非法终端发来的伪造数据包,进而过滤携带虚假身份的数据.

本文使用指纹匹配网络来分别进行接入时和通信时的指纹匹配和设备识别,分为 PAID 指纹匹配网络和 CSI 指纹匹配网络.设备识别时,首先将指纹输入匹配网络,再将输出的匹配结果和阈值进行比较来判定识别是否成功.PAID/CSI 指纹匹配网络具有相同的基本结构,其主要组成单元为自动编码器,在进行指纹匹配之前需要提前采集大量的 PAID/CSI 指纹来训练匹配网络.

设备身份识别方案的总流程见图 10.可以看到,设备识别流程主要分为 3 个步骤:

1) 接入时的设备识别.当接收到无线终端 D 发来的接入请求后,开始执行接入时设备识别,主要包括 PAID 指纹提取、CSI 指纹提取与 PAID 指纹匹配;

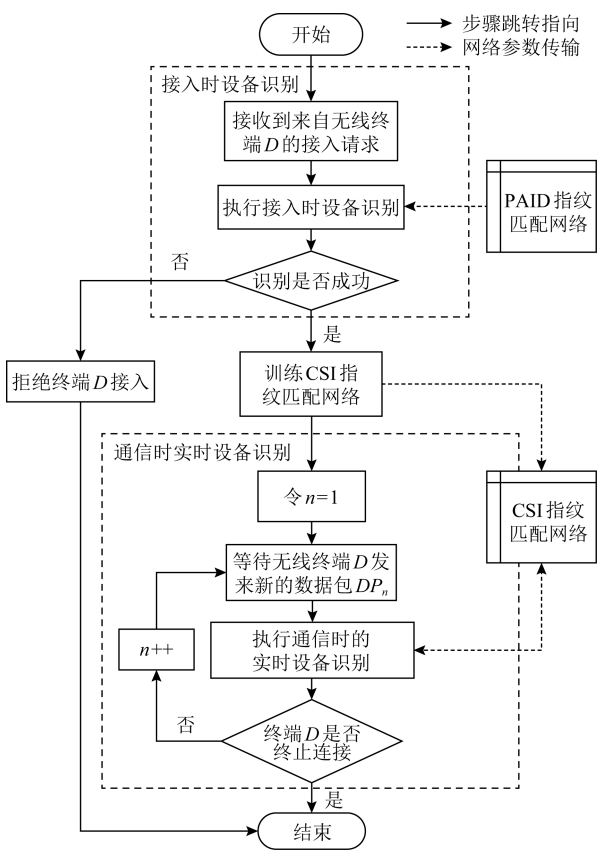


Fig. 10 Flow chart of the device identification scheme

图 10 设备身份识别方案的总流程图

获取指纹匹配结果后,将其与设定的阈值进行比较,并获得识别结果;若识别成功,利用新获得的 CSI 指纹更新 CSI 指纹匹配网络,否则拒绝终端 D 的接入请求并结束本次识别工作.

2) 更新 CSI 指纹匹配网络.接入时设备识别成功,说明新采集的 CSI 指纹携带的身份信息是真实的,因此以这些指纹为训练集重新训练 CSI 指纹匹配网络.重新训练的 CSI 指纹匹配网络能够反映终端 D 的最新无线信道状态,因此可以用于后续的实时设备识别.

3) 通信时的实时设备识别.当 CSI 指纹匹配网络训练结束后,对终端 D 发来的数据包进行实时的设备识别.也就是,从每个数据包中提取 CSI 指纹并进行 CSI 指纹匹配,若匹配结果超过预设的阈值,则数据包的身份识别成功,否则身份识别失败,将该数据包丢弃.

此外,图 11 给出了 PAID 指纹匹配网络的训练流程,该流程需要在开始识别无线终端之前完成,可以视作设备识别的准备工作.下文将按照顺序详细介绍各步骤的流程.

4.2 训练 PAID 指纹匹配网络

图 12 显示了 PAID 指纹匹配网络的基本结构.由图 12 可知,PAID 指纹匹配网络由匹配层和输出层组成.其中,匹配层由 I 个自动编码器并联组成,输出层则有一个线性变换器构成.假设匹配层的每个自动编码器的输入维度均为 Dim,压缩率均为 ρ,则自动编码器隐藏层的维度为 ρDim.线性变换器的输入 I 个自动编码器的重建误差 $RMSE$,其将所有自动编码器的 I 个重建误差进行线性变换得到最终匹配结果 γ,变换方法为

$$\gamma=\varepsilon_0+\varepsilon_1 RMSE_1+\cdots+\varepsilon_I RMSE_I. \quad (19)$$

在进行设备识别之前,需要提前采集无线终端的 PAID 指纹并训练 PAID 指纹匹配网络.图 11 给出了 PAID 指纹匹配网络的训练流程,具体描述为:

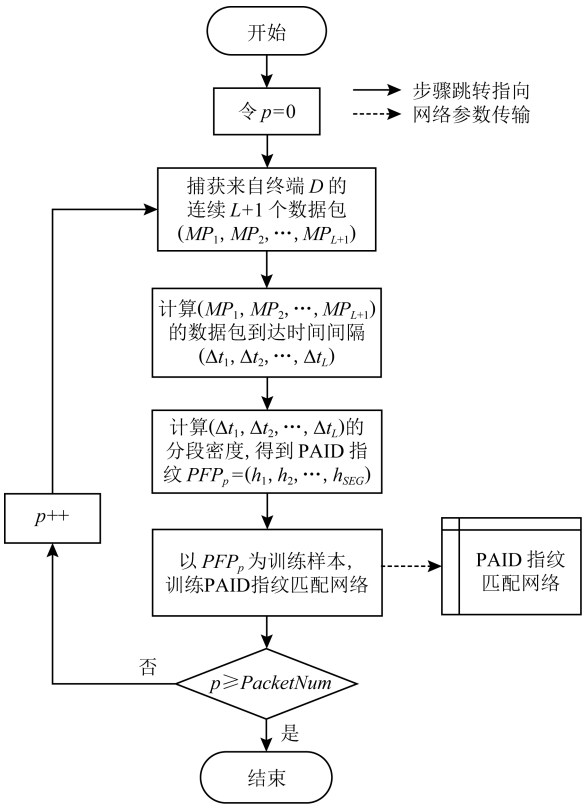


Fig. 11 Training flowchart of PAID fingerprint matching network

图 11 PAID 指纹匹配网络的训练流程图

1) 令 PAID 指纹计数器 $p=0$,该计数器用于记录当前已采集的 PAID 指纹数量.令目标指纹数量为 $PacketNum$,当已采集的 PAID 指纹数量达到 $PacketNum$ 后停止采集.

2) 初始化 PAID 指纹计数器后,识别设备进入监听状态,捕获来自无线终端 D 的连续 $L+1$ 个数据包,记作($MP_1, MP_2, \cdots, MP_{L+1}$).这些数据包被按照到达的时间排序,因此 MP_{i+1} 的到达时间一定大于等于 MP_i 的到达时间,其中 $i=1,2,\cdots,L$.

3) 识别设备在接收数据包 MP_i($i=1,2,\cdots$,

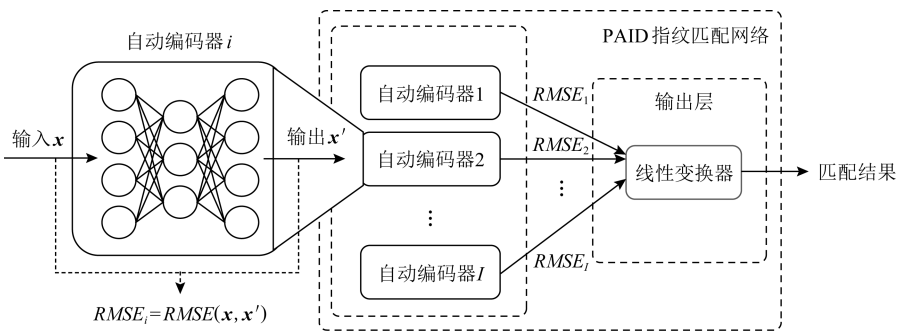


Fig. 12 PAID fingerprint matching network structure

图 12 PAID 指纹匹配网络结构

L)时,记录下其到达时间 t_i.当接收到 $L+1$ 个数据包后,识别设备获取无线终端 D 的数据包到达时间序列($t_1,t_2,\cdots,t_{L+1}$).然后,利用 3.1 节提供的 PAID 指纹生成方法计算数据包到达时间间隔($\Delta t_1,\Delta t_2,\cdots,\Delta t_L$).

4) 根据 3.1 节给出的分段密度计算方法计算($\Delta t_1,\Delta t_2,\cdots,\Delta t_L$)的分段密度,生成 PAID 指纹 $PFP_p=(h_1,h_2,\cdots,h_{SEG})$.以 PFP_p 为训练样本,对 PAID 指纹匹配网络进行单次训练.

5) PAID 指纹匹配网络的单次训练完成后,判断已采集的 PAID 指纹数量是否达标(达标即 $p\geqslant PacketNum$).若 $p\geqslant PacketNum$,结束 PAID 指纹匹配网络的训练;若 $p<PacketNum$,首先将指纹计数器 $p++$,然后返回数据包采集阶段,开始采集下一个 PAID 指纹.

图 13 为 PAID 指纹匹配网络的单次训练过程示意图.如图 13 所示,PAID 指纹匹配网络需要训练的部分为匹配层的 I 个自动编码器,输出层的线性变换器通常选用固定的参数,因此无需训练或进行数值更改.

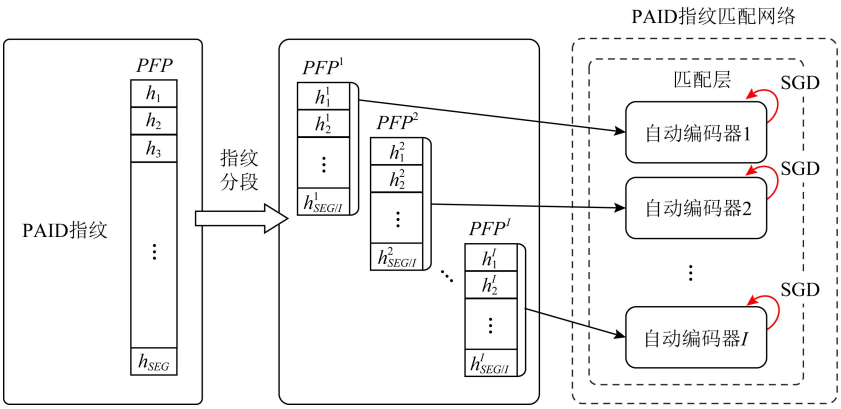


Fig. 13 Training process of PAID fingerprint matching network

图 13 PAID 指纹匹配网络的训练过程示意图

图 13 中所示的训练过程主要包括 2 个方面.

1) PAID 指纹分段.在进行 PAID 指纹匹配网络训练时,首先需要将用于训练网络的 PAID 指纹 $PFP=(h_1,h_2,\cdots,h_{SEG})$ 分割为等长的 I 个子 PAID 指纹,记为($PFP^1,PFP^2,\cdots,PFP^I$),其中 $PFP^i=(h_1^i,h_2^i,\cdots,h_{SEG/I}^i)$,$i=1,2,\cdots,I$.PAID 指纹分段的原因是:匹配层由多个输入维度一致的自动编码器组成,这些自动编码器分别负责匹配 PAID 指纹的每个子段,因此需要将 PAID 指纹分段为子指纹,再作为各个自动编码器的输入.

2) 匹配层自动编码器的训练.令 θ 表示匹配层中所有自动编码器的集合,$\theta_i(i=1,2,\cdots,I)$表示匹配层中的第 i 个自动编码器,则训练过程为:将 PFP^i 输入 $\theta_i(i=1,2,\cdots,I)$并进行前向传播,获得所有神经元的激活 A_i 和输出 y_i;利用后向传播算法计算所有神经元的激活误差 $delta_i$;结合 A_i 和 $delta_i$,利用随机梯度下降算法更新 θ_i.以上所使用的前向传播算法、后向传播算法和随机梯度下降算法如 2.3 节所示.

PAID 指纹匹配网络的单次训练算法描述为:

1) 将 PAID 指纹 PFP_p 分割为等长的 I 个子 PAID 指纹,记为($PFP_p^1,PFP_p^2,\cdots,PFP_p^I$),其中 $PFP_p^i=(h_1^i,h_2^i,\cdots,h_{SEG/I}^i)$,$i=1,2,\cdots,I$;

2) 针对子 PAID 指纹 $PFP_p^i(i=1,2,\cdots,I)$,首先利用最大-最小值归一化法进行归一化,得到 $PFP_{p,0\text{-}1}^i$,具体归一化方法为

$$h_k^{i\prime}=\frac{h_k^i-h_{k\text{-}\min}^i}{h_{k\text{-}\max}^i-h_{k\text{-}\min}^i},\tag{20}$$

其中,$h_{k\text{-}\max}^i$为 $PFP_q^i(q=0,1,\cdots,p-1)$对应的 h_k^i 的最大值,$h_{k\text{-}\min}^i$则为 $PFP_q^i(q=0,1,\cdots,p-1)$对应的 h_k^i 的最小值;

3) 将 $PFP_{p,0\text{-}1}^i$ 输入 θ_i 并进行前向传播,获得神经元的激活 A_i 和输出 $y_i=(o_1^i,o_2^i,\cdots,o_{SEG/I}^i)$;

4) 将 $PFP_{p,0\text{-}1}^i$ 和 y_i 输入 θ_i 并进行后向传播,获得激活误差 $delta_i$;

5) 利用神经元的激活 A_i 和激活误差 $deltas_i$ 训练 θ_i.

4.3 基于 PAID 指纹的接入时设备识别

图 14 的前 3 个主要步骤描述了 PAID 指纹提取的流程.当识别设备接收到来自无线终端 D 的接入请求后,需要获得终端 D 的 PAID 指纹,以进行接入时的设备识别.为了获取终端 D 的 PAID 指纹,

识别设备进入监听状态,捕获来自无线终端 D 的连续 $L+1$ 个数据包,记作($P_1,P_2,\cdots,P_{L+1}$).这些数据包按照到达的时间排序,因此 P_{i+1}的到达时间一定大于等于 P_i 的到达时间,其中 $i=1,2,\cdots,L$.

识别设备在接收数据包 $P_i(i=1,2,\cdots,L)$时,记录下其到达时间 t_i.当接收到 $L+1$ 个数据包后,识别设备获取无线终端 D 的数据包到达时间序列($t_1,t_2,\cdots,t_{L+1}$).然后,利用 3.1 节提供的 PAID 指纹生成方法计算数据包到达时间间隔($\Delta t_1,\Delta t_2,\cdots,\Delta t_L$).

根据 3.1 节给出的分段密度计算方法计算($\Delta t_1,\Delta t_2,\cdots,\Delta t_L$)的分段密度,生成 PAID 指纹 $PFP=(h_1,h_2,\cdots,h_{SEG})$.

图 14 的后 2 个主要步骤描述了 PAID 指纹匹配的流程.获取无线终端 D 的 PAID 指纹 PFP 后,识别设备利用提前训练好的 PAID 指纹匹配网络来匹配 PFP,并根据匹配结果来判断接入时识别成功与否.首先,识别设备将 PFP 输入 PAID 指纹匹配网络.匹配网络输出匹配结果 γ_0,识别设备将 γ_0 与预设的阈值 $threshold_{ac}$进行比较来判断识别是否成功:若 $\gamma_0\leqslant threshold_{ac}$则识别成功,即判断无线终端 D 使用了真实的身份信息;若 $\gamma_0>threshold_{ac}$则识别失败,即判断无线终端 D 使用了虚假的身份信息.

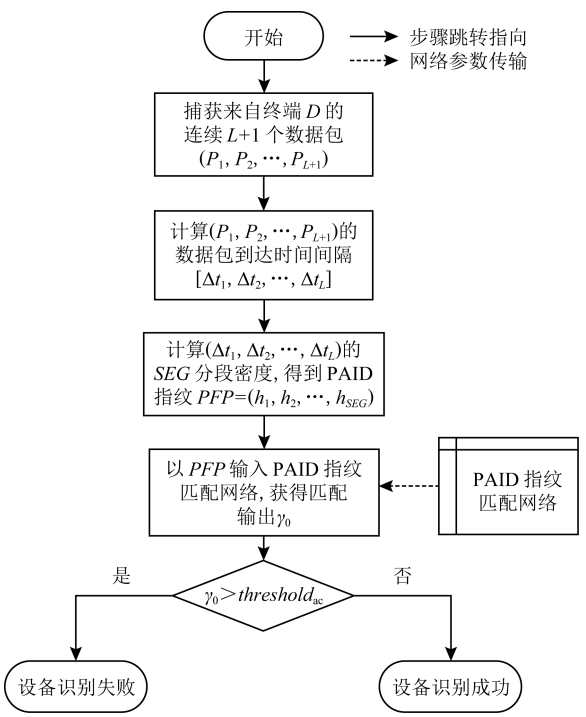


Fig. 14 Flow chart of PAID fingerprint based device authentication in access phase

图 14 基于 PAID 指纹的接入时设备识别流程图

PAID 指纹匹配算法描述如下:

1) 定义一个 I 维向量 $\boldsymbol{v}$,并初始化为 $\mathbf{0}$.向量 $\boldsymbol{v}$ 为线性变换器的输入,其内部存储匹配层的 I 个自动编码器的匹配输出.

2) 将 PAID 指纹 PFP 分割为等长 I 个子 PAID 指纹,记为($PFP^1,PFP^2,\cdots,PFP^I$),其中 $PFP^i=(h_1^i,h_2^i,\cdots,h_{SEG/I}^i)$,$i=1,2,\cdots,I$.

3) 针对子 PAID 指纹 $PFP^i(i=1,2,\cdots,I)$,首先利用最大-最小值归一化法进行归一化,得到 $PFP_{0\text{-}1}^i$,具体归一化方法为

$$h_k^{i\prime}=\frac{h_k^i-h_{k\text{-}\min}^i}{h_{k\text{-}\max}^i-h_{k\text{-}\min}^i},\tag{21}$$

其中,$h_{k\text{-}\max}^i$为 $PFP_p^i(p=0,1,\cdots,PakcetNum-1)$对应的 h_k^i 的最大值,$h_{k\text{-}\min}^i$则为 $PFP_p^i(p=0,1,\cdots,PakcetNum-1)$对应的 h_k^i 的最小值.

4) 将 $PFP_{0\text{-}1}^i$输入 θ_i 并进行前向传播,获得神经元的激活 A_i 和输出 $y_i=(o_1^i,o_2^i,\cdots,o_{SEG/I}^i)$.

5) 计算 $PFP_{0\text{-}1}^i$和 y_i 的均方根误差,写入 $\boldsymbol{v}$ 的第 i 个值.均方根误差计算方法为

$$RMSE_i=RMSE(PFP_{0\text{-}1}^i,y_i)=\left(\frac{1}{SEG/I}\sum_{k=1}^{SEG/I}(h_k^{i\prime}-o_k^i)^2\right)^{\frac{1}{2}}.\tag{22}$$

6) 所有自动编码器的前向传播完成后,将 $\boldsymbol{v}$ 输入线性变换模块,得到最终匹配结果 γ_0:

$$\gamma_0=\varepsilon_0+\varepsilon_1RMSE_1+\cdots+\varepsilon_IRMSE_I.\tag{23}$$

4.4 训练 CSI 指纹匹配网络

CSI 指纹匹配网络具有与图 11 所示的 PAID 指纹匹配网络相同的基本结构,由匹配层和输出层组成.其中,匹配层由 J 个自动编码器并联组成,输出层则有一个线性变换器构成.其中,每个自动编码器的压缩率均为 ρ.线性变换器的输入为 J 个自动编码器的重建误差 $RMSE$,其将所有自动编码器的 J 个重建误差进行线性变换得到最终匹配结果 γ,变换方法为

$$\gamma=\varepsilon_0+\varepsilon_1RMSE_1+\cdots+\varepsilon_JRMSE_J.\tag{24}$$

CSI 指纹匹配网络的训练过程与 PAID 指纹匹配网络近似,最大的区别在于:CSI 指纹在生成时已经归一化,因此输入匹配网络之前无需再次归一化.

在进行通信时的实时设备识别之前,需要训练 CSI 指纹匹配网络,以获取无线终端最新的无线信道状态.图 15 给出了 CSI 指纹匹配网络的训练流程,具体描述为:

1) 从在接入时获取的 $L+1$ 个数据包($P_1,P_2,\cdots,P_{L+1}$)中提取出 $L+1$ 个 CSI 指纹($CFP_1,CFP_2,\cdots,CFP_{L+1}$),指纹提取方法见第 3.2 节;

2) 对 $L+1$ 个 CSI 指纹(CFP_1,CFP_2,…,CFP_{L+1})进行降噪处理,得到(CFP'_1,CFP'_2,…,CFP'_{L+1});

3) 以(CFP'_1,CFP'_2,…,CFP'_{L+1})为训练集训练 CSI 指纹匹配网络;

4) 将训练好的 CSI 指纹匹配网络存储在本地,用于通信时的实时设备识别.

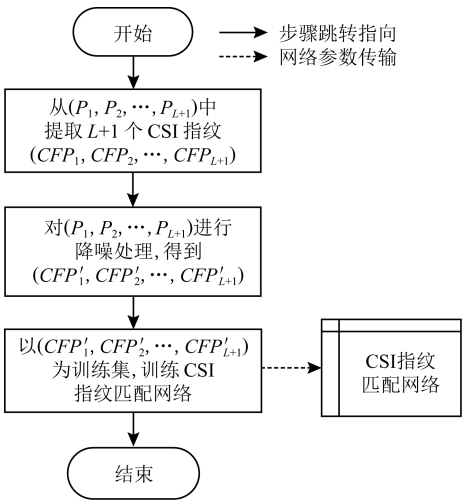


Fig. 15 Training flowchart of CSI fingerprint matching network

图 15 CSI 指纹匹配网络的训练流程图

CSI 指纹的降噪处理包括 2 个步骤,分别为异常值剔除和时域平滑.

异常值剔除的步骤为:在短时间间隔内,CSI 指纹中的每个子载波对应的数值应在有限的范围内浮动.但是在现实中往往由于存在软硬件缺陷等原因而出现远超预期数值波动范围的数值.这种数值无法反映设备的无线信道特征,保留下来会降低 CSI 指纹的设备识别性能.因此需要将这些异常值剔除.

本文选用 Hampel Identifier 算法来剔除异常值.需要注意的是,剔除异常值时并不将整个指纹作为剔除的对象,而是利用更细粒度的方法,对每个子载波分别进行异常值检测.将所有 CSI 指纹(CFP_1,CFP_2,…,CFP_{L+1})中的第 i 个子载波对应的数值组成序列(c_i^1,c_i^2,…,c_i^{L+1}).对于 c_i^j($j=l+1,l+2,\cdots,L-l+1$),将 c_i^j 及其左右各 l 个数值视作长度为 $2l+1$ 的窗口 W_i^j.从中选取出中位数 med_i^j,并用每个样本与中位数绝对偏差的均值来估计 W_i^j 中样本的标准差.其中,W_i^j 的标准差为

$$\sigma_i^j=\frac{1}{\sqrt{2}\,\mathrm{erfinv}(0.5)}\mathrm{median}(|c_i^j-med_i^j|),\tag{25}$$

其中,erfinv 为逆误差函数.

最后,对 c_i^j($j=1,2,\cdots,L+1$)进行数值代换来剔除异常值:

$$c_i^j=\begin{cases}c_i^j, & |c_i^j-med_i^j|\leqslant\eta\sigma_i^j,\\ med_i^j, & |c_i^j-med_i^j|>\eta\sigma_i^j,\end{cases}\tag{26}$$

其中,η 为判断数值是否为异常值的阈值权重.

时域平滑的步骤为:在时域上对(c_i^1,c_i^2,…,c_i^{L+1})($i=1,2,\cdots,L+1$)进行平滑滤波,滤波公式为

$$c_i^{k\prime}=\frac{1}{\omega}\sum_{i=\max\left(0,k-\left\lfloor\frac{\omega}{2}\right\rfloor\right)}^{\min\left(L+1,k+\left\lfloor\frac{\omega-1}{2}\right\rfloor\right)}c_i^k,\tag{27}$$

其中,ω 为平滑滤波窗口的长度.

CSI 指纹匹配网络的训练算法描述为:

1) 将 CSI 指纹 CFP 分割为等长的 J 个子 CSI 指纹,记为(CFP^1,CFP^2,…,CFP^J),其中 $CFP^j=(c_1^j,c_2^j,\cdots,c_{N/J}^j)$,$j=1,2,\cdots,J$;

2) 将子 CSI 指纹 CFP^j($j=1,2,\cdots,J$)输入 ϑ_j 并进行前向传播,获得神经元的激活 A_j 和输出 $y_j=(o_1^j,o_2^j,\cdots,o_{N/J}^j)$;

3) 将 CFP^j 和 y_j 输入 ϑ_j 并进行后向传播,获得激活误差 $deltas_j$;

4) 结合 2.3 节的随机梯度下降算法,利用神经元的激活 A_j 和激活误差 $deltas_j$ 训练 ϑ_j.

4.5 基于 CSI 指纹的通信时实时设备识别

图 16 的第 1 个主要步骤为 CSI 指纹提取.当无线终端 D 通过接入时的设备识别后,终端 D 成功接入无线网络,并开始不断地发送数据包,识别设备则利用 CSI 指纹来识别每个数据包携带的身份信息是否真实.

识别设备首先令数据包计数器 $n=1$.数据包计数器用于记录从通信时的实时识别开始到目前为止当前已经识别的数据包的数量.然后,等待无线终端 D 发来新的数据包 DP_n.在终端 D 没有发送新的数据包时,识别设备将处于阻塞状态;当接收到终端 D 发来新的数据包 DP_n 后,识别设备结束阻塞状态,并针对该包进行身份识别.

获取新数据包 DP_n 的 CSI 后,生成 CSI 指纹 CFP_n,具体的 CSI 指纹生成方法如 3.2 节所述:首先,从所有子载波中筛选出可用子载波,得到 CSI';然后提取 CSI' 的幅度,得到 CSI 幅度序列 A_{CSI};利用最大-最小归一化法对 A_{CSI} 进行归一化,得到 CSI 指纹 CFP_n.

图 16 的带背景颜色的框为 CSI 指纹匹配的 3 个主要步骤.获取数据包 DP_n 的 CSI 指纹 CFP_n 后,将其输入 CSI 指纹匹配网络进行身份识别.将匹配

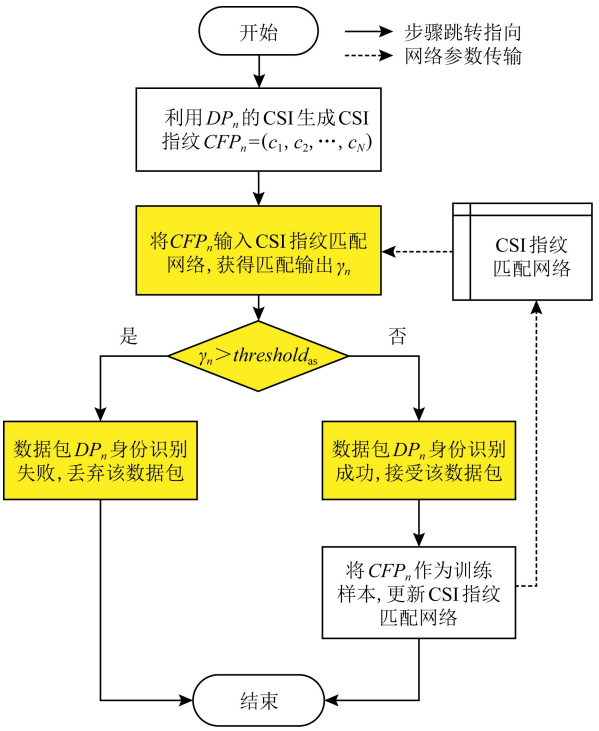


Fig. 16 Flow chart of real-time CSI fingerprint based device authentication in communication phase

图 16 基于 CSI 指纹的通信时实时设备识别流程图

网络的输出 γ_n 与阈值 $threshold_{as}$ 比较来获得识别结果:若 $\gamma_n \leqslant threshold_{as}$ 则数据包 DP_n 的身份识别成功,接受该数据包;否则识别失败,判定 DP_n 携带了虚假的设备身份并丢弃该数据包.

CSI 指纹匹配算法描述为:

1) 定义一个 J 维向量 $\boldsymbol{v}$,并初始化为 0.向量 $\boldsymbol{v}$ 为线性变换器的输入,其内部存储匹配层的 J 个自动编码器的匹配输出.

2) 将从数据包 DP_n 中获取的 CSI 指纹 CFP_n 分割为等长的 J 个子 CSI 指纹,记为 $(CFP_n^1, CFP_n^2, \cdots, CFP_n^J)$,其中 $CFP_n^j=(c_1^j, c_2^j, \cdots, c_{N/J}^j)$, $j=1,2,\cdots,J$.

3) 将子 CSI 指纹 $CFP_n^j (j=1,2,\cdots,J)$ 输入 ϑ_j 并进行前向传播,获得神经元的激活 A_j 和输出 $y_j=(o_1^j, o_2^j, \cdots, o_{N/J}^j)$.

4) 计算 CFP_n^j 和 y_j 的均方根误差,写入 $\boldsymbol{v}$ 的第 j 个值.均方根误差计算方法为

$$RMSE_j = RMSE(CFP_n^j, y_j) = \left(\frac{1}{N/J}\sum_{k=1}^{N/J}(c_k^j - o_k^i)^2\right)^{\frac{1}{2}}. \tag{28}$$

5) 所有自动编码器的前向传播完成后,将 $\boldsymbol{v}$ 输入线性变换模块,得到最终匹配结果 γ_n:

$$\gamma_n = \varepsilon_0 + \varepsilon_1 RMSE_1 + \varepsilon_2 RMSE_2 + \cdots + \varepsilon_J RMSE_J. \tag{29}$$

图 16 的最后一个主要步骤为更新 CSI 指纹匹配网络.在无线终端通信过程中,即使终端位置没有发生改变,终端所处环境的细微变化的叠加会体现在终端的 CSI 指纹上,表现为在一段时间内(如 10 min)采集的 CSI 指纹发生"连续"的变化,该"连续"是指相邻的 CSI 指纹之间的差异很小.这种变化虽然在短期内不会使 CSI 指纹产生较大的改变,但是设备识别通常是一项长期的工作,因此在接入时设备识别过程中重新训练的 CSI 指纹匹配网络很可能再次失效(即无法反映最新的信道状态,进而使得合法设备识别失败).

利用从新数据包中获取的 CSI 指纹不断地训练 CSI 指纹匹配网络可以使匹配网络始终反映最新的信道状态.在实际通信过程中,每秒发送的数据包通常为数百个,因此当无线终端活跃时,每秒都可以获取足够的 CSI 指纹,用于更新 CSI 指纹匹配网络.重要的是,仅使用识别成功的 CSI 指纹来更新匹配网络,因为识别失败的 CSI 指纹被判定为来自携带虚假身份信息的非法终端.更新 CSI 指纹匹配网络的算法与 4.4 节中给出的单次训练算法一致.

4.6 指纹匹配网络时间复杂度分析

本节主要分析 PAID 指纹匹配网络和 CSI 指纹匹配网络的时间复杂度.分析时间复杂度时将会使用的变量有:SEG 和 N 分别表示 PAID 指纹和 CSI 指纹的长度,I 和 J 分别表示 PAID 指纹和 CSI 指纹的分段数量(即匹配层中的自动编码器的数量),则 SEG/I 和 N/J 分别表示 PAID 指纹和 CSI 指纹匹配网络的自动编码器的输入维度.此外,ρ 表示自动编码器的压缩率,则硬件和 CSI 指纹匹配网络中自动编码器隐藏层的神经元数量分别为 $\rho SEG/I$ 和 $\rho N/J$.

对于单个自动编码器,假设其输入维度为 u,中间层的降维比例为 ρ.在前向传播过程中,中间层的激活需要 $u \times \rho u = \rho u^2$ 次计算,输出层的激活同样需要 ρu^2 次计算.因此,单个自动编码器的前向传播复杂度为 $O(\rho u^2)=O(u^2)$.此外,自动编码器的后向传播与前向传播具有相同的复杂度,因此后向传播复杂度也为 $O(u^2)$.

假设匹配层的所有自动编码器均为串行运算,则利用 PAID 指纹匹配网络进行一次指纹匹配的复杂度为 $O(I \times (SEG/I)^2 + SEG) = O(SEG^2/I)$.令 $I=SEG/\beta$,β 为子 PAID 指纹的长度.如果对匹配层

中自动编码器的规模进行限制(限制在 6 或 6 以下),则 β 可以被视为常量.则复杂度表达为 $O(SEG^2/I)=O(SEG^2/(SEG/\beta))=O(\beta SEG)=O(SEG)$.

同样地,利用 CSI 指纹匹配网络进行一次指纹匹配的复杂度为 $O(J\times(N/J)^2+N)=O(N^2/J)$.令 $J=N/\alpha$,α 为子 CSI 指纹的长度.如果对匹配层中自动编码器的规模进行限制(限制在 6 或 6 以下),则 α 可以被视为常量.则复杂度表示为 $O(N^2/J)=O(N^2/(N/\alpha))=O(\alpha N)=O(N)$.

当不采用多个自动编码器组成的网络,而采用单个大规模自动编码器进行指纹匹配时,PAID 指纹匹配网络和 CSI 指纹匹配网络将分别由输入维度为 SEG 和 N 的自动编码器构成,根据上文所推导的自动编码器的时间复杂度可知,此时 PAID 指纹匹配网络和 CSI 指纹匹配网络的识别时间复杂度为 $O(SEG^2)$和 $O(N^2)$.由此可见,如果将单个自动编码器的输入维度控制在固定值以下,那么本文提供的指纹匹配网络能够将平方时间复杂度降低到线性时间复杂度.

5 设备身份识别框架的实现与评估

本节讲述了基于混合特征指纹的设备识别框架的实现与评估.首先整体描述了基于混合特征值的识别框架,然后描述了识别框架中的各个部分的具体实现方法,接着给出了实际测试的场景设置、评估指标以及 2 个设备识别阶段下的设备识别结果.

5.1 设备识别框架的设计与实现

图 17 展示了设备识别框架.从组成来看,该框架可分为 3 个部分,分别为:指纹匹配网络、接入时设备识别和通信时实时设备识别.其中,接入时设备识别部分模拟了无线终端接入时的设备识别过程,通信时实时设备识别则模拟了终端成功接入后、开始通信时实时的逐包识别过程.

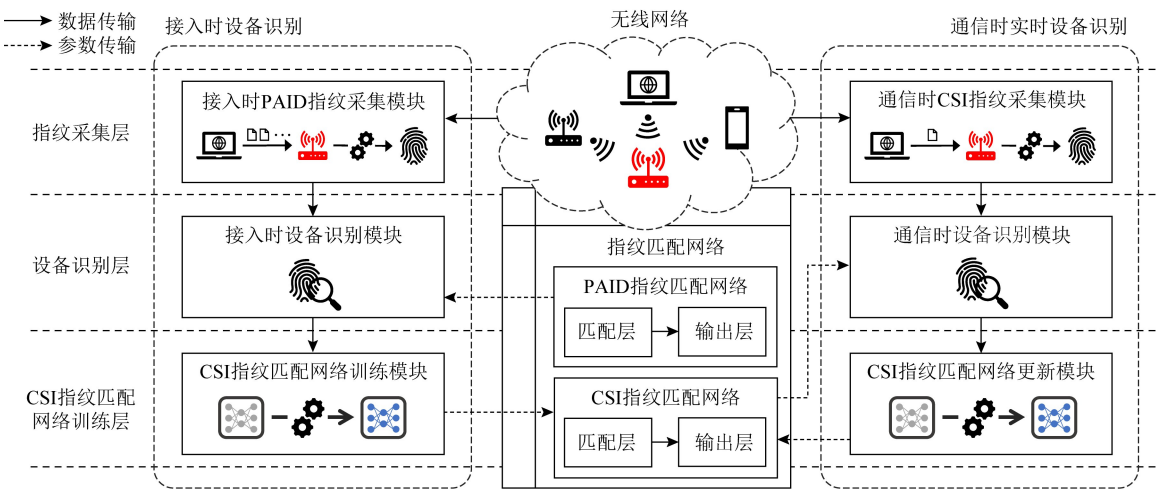


Fig.17 Wireless device identification framework based on hybrid feature fingerprint

图 17 基于混合特征指纹的无线设备识别框架

从功能来看,该框架可分为 3 个层次:指纹采集层、设备识别层和 CSI 指纹匹配网络训练层.

指纹采集层包括:

1) 接入时 PAID 指纹采集模块,负责在无线终端请求接入时采集 PAID 指纹;

2) 通信时 CSI 指纹采集模块,负责在无线终端通信时,从终端发来的每一个数据包中提取 CSI 指纹.

设备识别层包括:

1) 接入时设备识别模块,负责利用无线终端的 PAID 指纹来识别其身份;

2) 通信时设备识别模块,负责利用数据包的 CSI 指纹来识别其身份,进而判断数据包是否携带真实的身份信息.

CSI 指纹匹配网络训练层包括:

1) CSI 指纹匹配网络训练模块,负责利用接入时采集的新 CSI 指纹来训练网络;

2) CSI 指纹匹配网络更新模块,负责利用识别成功的 CSI 指纹单次训练网络.

为了更清晰地描述整个设备识别框架,给出一台无线终端的完整识别流程:识别设备接收到来自终端的接入请求时,接入时 PAID 指纹采集模块采集该终端的 PAID 指纹;获得 PAID 指纹后,接入时设备识别模块读取该终端的 PAID 指纹匹配网络并

进行指纹匹配,根据匹配结果得到识别结果;若识别成功,CSI 指纹匹配网络训练模块利用接入时 PAID 指纹采集模块中获得的数据包生成 CSI 指纹集,将其作为训练集来训练 CSI 指纹匹配网络,训练后的网络覆盖原网络;通信时 CSI 指纹采集模块等待终端发来的数据包,捕获到新数据包后,从中提取 CSI 指纹;通信时设备识别模块读取该终端的 CSI 指纹匹配网络并进行指纹匹配,匹配成功则接受该数据包,否则丢弃;若数据包识别成功,CSI 指纹匹配网络更新模块利用该 CSI 指纹单次训练 CSI 指纹匹配网络.

为了更方便地描述指纹匹配网络以及 3 个层次的实现方式,假设设备识别框架的应用场景为带宽为 20 MHz、使用 64 个 OFDM 子载波的无线网络,识别设备和无线终端均具有单收发天线.

指纹匹配网络是无线设备识别框架的核心,它包含 PAID 指纹匹配网络和 CSI 指纹匹配网络,负责在 2 个设备识别阶段(接入时设备识别、通信时逐包设备识别)提供用于指纹匹配的匹配网络.

其具体实现方式为:对于每个需要使用无线网络的合法无线终端,识别设备在内部新建一个 PAID 指纹匹配网络文件和一个 CSI 指纹匹配网络文件,文件用 JSON 格式保存用于描述指纹匹配网络的参数.通过读取和写入文件中的相应参数来实现匹配网络的读取和更改.

下面将讨论指纹匹配网络的参数.回顾第 4 节可知,PAID 指纹匹配网络和 CSI 指纹匹配网络均由自动编码器和线性变换器组成.本文所使用的线性变换器选用均值变换器,即对于 PAID 指纹匹配网络有 $\varepsilon_1=\varepsilon_2=\cdots=\varepsilon_I=1/I$,$\varepsilon_0=0$;对于 CSI 指纹匹配网络有 $\varepsilon_1=\varepsilon_2=\cdots=\varepsilon_J=1/J$,$\varepsilon_0=0$.以上参数均为固定值,因此无需存储在文件中.此外,本文将 PAID/CSI 指纹匹配网络中所有自动编码器的学习率设为 0.05,压缩率 ρ 设为 0.75,激活函数设为 sigmoid 函数,匹配层自动编码器的个数 I 和 J 均设为固定值,这些参数均无需存储.因此,需要存储在文件中的参数仅为:PAID 指纹匹配网络的匹配层中所有自动编码器的参数 $\theta=\{\theta_1,\theta_2,\cdots,\theta_I\}$,以及 CSI 指纹匹配网络的匹配层中所有自动编码器的参数 $\boldsymbol{\vartheta}=(\vartheta_1,\vartheta_2,\cdots,\vartheta_J)$.

指纹采集层主要负责接入时和通信时的指纹采集,包括接入时 PAID 指纹采集模块、通信时 CSI 指纹采集模块.这 2 个模块分别负责接入时识别和通信时逐包识别的指纹采集.

1) 接入时 PAID 指纹采集模块的实现:首先定义一个空的数据包到达序列 PA,然后令该终端以 100 pkts/s 的发送速度连续发出 $L+1$ 个数据包 $(P_1,P_2,\cdots,P_{L+1})$.每接收到一个数据包 P_i,模块提取其到达时间和 CSI 序列,分别记作 t_i 和 C_i,将 t_i 添加到 PA 的末尾,将 C_i 放入 CSI 队列,以备后续使用.捕获完成后,按照 3.1 节给出的 PAID 指纹生成方案,利用 PA 生成 PAID 指纹 PFP,并放入 PAID 指纹队列,以备后续使用.其中,在接入时让终端以指定速度连续发送数据包需要修改现有的协议,本文为了简化设计,选择利用 ICMP 协议来模拟这一过程:识别设备以 100 pkts/s 的速度向终端发送 ICMP 回送请求(echo request)数据包,并捕获终端卡来的 ICMP 回送响应(echo reply)数据包.

2) 通信时 CSI 指纹采集模块的实现:等待无线终端发来数据包;接收到一个数据包 DP_n 后,获取该数据包的 CSI 序列,记作 C'_n;按照 3.2 节给出的 CSI 指纹生成方案,利用 C'_n 生成 CSI 指纹 CFP_n,并放入 CSI 指纹队列,以备后续使用.

设备识别层主要负责接入时和通信时的设备识别,包括接入时设备识别模块、通信时设备识别模块.这 2 个模块分别负责接入时和通信时的指纹匹配,并根据匹配结果来获得识别结果.

1) 接入时设备识别模块的实现.开始工作后,首先进入阻塞等待状态,等待 PAID 指纹队列非空;PAID 指纹队列出现新指纹后等待结束,从队列中取出 PAID 指纹 PFP;读取 PAID 指纹匹配网络参数,并新建一个匹配网络;将 PFP 输入网络,获得匹配结果 γ_0;比较 γ_0 与 $threshold_{\mathrm{ac}}$,若 $\gamma_0>threshold_{\mathrm{ac}}$ 则识别失败并向所有模块发出终止信号,否则识别成功并向 CSI 指纹匹配网络训练模块发出识别成功信号.

2) 通信时设备识别模块的实现.开始工作后,首先进入阻塞等待状态,CSI 指纹匹配网络训练模块发来训练完成信号;接收到信号后,读取 CSI 指纹匹配网络参数,并新建一个匹配网络;阻塞等待 CSI 指纹队列非空;CSI 指纹队列出现新指纹后等待结束,从队列中取出 CSI 指纹 CFP_n;将 CFP_n 输入网络,获得匹配结果 γ_n;比较 γ_n 与 $threshold_{\mathrm{as}}$,若 $\gamma_n>threshold_{\mathrm{as}}$ 则识别失败并向所有模块发出终止信号,否则识别成功并将 CFP_n 放入训练队列;进入阻塞等待状态,继续等待 CSI 指纹队列非空.

CSI 指纹匹配网络更新层主要负责接入时和通信时的 CSI 指纹匹配网络训练,包括 CSI 指纹匹配

网络训练模块、CSI 指纹匹配网络更新模块.这 2 个模块分别负责接入时和通信时的网络训练以及更新.

1) CSI 指纹匹配网络训练模块的实现.开始工作后,首先进入阻塞状态,等待通信时设备识别模块发来识别成功信号;接收信号后,模块从 CSI 指纹队列中读取 $L+1$ 个 CSI,并生成 $L+1$ 个 CSI 指纹(CFP_1,CFP_2,…,CFP_{L+1});对这些指纹进行降噪,得到(CFP'_1,CFP'_2,…,CFP'_{L+1});读取 CSI 指纹匹配网络参数,并新建一个匹配网络;以(CFP'_1,CFP'_2,…,CFP'_{L+1})为训练集训练 CSI 指纹匹配网络;训练完成后将参数写回文件;向通信时设备识别模块发送训练完成信号.

2) CSI 指纹匹配网络更新模块的实现.开始工作后,首先进入阻塞等待状态,等待训练队列非空;训练队列出现新指纹后等待结束,从队列中取出用于更新的 CSI 指纹 CFP;用 CFP 单次训练通信时识别模块的 CSI 指纹匹配网络.

需要注意的是,该模块与通信时设备识别模块共用一个 CSI 指纹匹配网络.若两模块不共用网络,那么当该模块更新网络后,需要将新的参数写入文件,然后通信时设备识别模块需要读取文件并重新建立匹配网络.前者避免了频繁的文件读写,提高了识别的时间效率.

5.2 测试场景与评估指标

本节所提供的无线设备身份识别框架包含接入时识别和通信时逐包识别这 2 个工作阶段,2 个阶段使用不同类型的指纹、不同的指纹采集和处理方案以及不同的指纹匹配方案.为了评估每种指纹识别方案的性能,本节先后测试了基于 PAID 指纹的接入时设备识别和基于 CSI 指纹的逐包设备识别.

在接入时设备识别测试中,选取了 18 台无线设备作为被识别的对象,设备信息如表 1 所示.其中包含 3 台型号相同的路由器、3 台型号相同的树莓派以及 3 台型号相同的台式电脑,这 3 组设备用于评估接入时设备识别方案识别相同型号的设备性能.其余还包括手机、平板电脑以及笔记本电脑,用于测试该识别方案能否用于多种设备类型.此外,用 1 台树莓派(Raspberry Pi 3 Model B+)作为识别设备,负责采集待测无线终端的 PAID 指纹.该设备通过安装 hostapd 服务和 dhcp 服务实现无线网络共享,进而作为无线路由器与无线终端直接相连,方便采集指纹的同时消除了数据包转发时延对 PAID 指纹带来的干扰.

Table 1 The Information of the Devices Used for Evaluation in Access Stage

表 1 用于测试接入时设备识别的设备信息

设备编号	设备名称
1	HUAWEI WS5106
2	HUAWEI WS5106
3	HUAWEI WS5106
4	Raspberry Pi 3 Model B+
5	Raspberry Pi 3 Model B+
6	Raspberry Pi 3 Model B+
7	Lenovo 启天 M610-D329
8	Lenovo 启天 M610-D329
9	Lenovo 启天 M610-D329
10	HUAWEI Mate 30 5G
11	iPad Air 3
12	DELL Inspiron 15-7560
13	HUAWEI P10 Plus
14	HONOR V40
15	Lenovo 拯救者 R7000
16	Lenovo 小新 15
17	360 安全路由 5G P2
18	PHICOMM 路由器

接入时设备识别测试流程为:识别设备 R 开启无线网络共享,待测终端 D 接入 R 的共享网络,此时 R 为 D 的网关;R 以 100 pkts/s 的速度向 D 发送 ICMP 回送请求包,并监听 D 的 ICMP 回送响应包,记录响应包到达时间;每采集到 $L+1$ 个到达时间,生成 PAID 指纹并写本地文件;采集 8 000 个 PAID 指纹后停止;新建 1 个 PAID 指纹匹配网络,读取本地文件中的指纹,以前 1 000 个 PAID 指纹为训练集训练匹配网络,以后 7 000 个指纹为测试集测试 PAID 指纹识别的准确率.对所有待测设备重复以上步骤.

在通信时逐包设备测试中,选取 2 个测试场景进行 CSI 指纹采集和设备识别,分别为公寓测试场景和实验室测试场景.2 个测试场景均使用带宽为 20 MHz 的 WiFi 网络,并且均用一台固定放置的树莓派(Raspberry Pi 3 Model B+)作为识别设备,负责采集待测无线终端的 CSI 指纹.该树莓派安装了 Gringoli 等人[26]提供的插件,该插件通过修改 WiFi 网卡驱动程序来监听指定 MAC 地址的无线数据包,并将该数据包的 CSI 发送到应用层.公寓测试场景中,将 4 台无线终端作为待测设备放置在 4 个

不同的位置,其中 3 台放置在无物体移动、无人员走动的室内,1 台放置在无人员走动、但是离识别设备较远且有墙壁相隔的走廊.在实验室测试场景中,将 1 台无线终端作为待测设备放置在固定的位置,人员在距其 1 m,2 m,3 m 位置处来回走动.

通信时逐包设备识别测试流程为:在公寓测试场景中,识别设备创建 4 个空文件,分别用于存储 4 台无线终端的 CSI;4 台终端同时以 100 pkts/s 的速率对任意 IP 发送 ICMP 回送请求包;识别设备捕获数据包,同时获取数据包的 MAC 地址和 CSI,根据 MAC 地址将 CSI 存入相应的文件;每台设备均采集 10 000 组 CSI,采集完成后停止;对于每台设备,对前 $L+1$ 个指纹进行降噪,降噪后的指纹作为训练集;剩余的指纹作为测试集.

为了评估身份验证框架的性能,使用真正例率 TPR 和真反例率 TNR 作为两阶段的设备识别准确率的评估指标.其中,$TPR=TP/(TP+FN)$:TP 是真正例的数量,在本文中即某终端的所有指纹中识别成功的指纹数量;FN 是假反例的数量,即某终端的所有指纹中识别失败的指纹数量.$TNR=TN/(TN+FP)$:TN 是真反例的数量,即所有不属于某终端的指纹中识别失败的指纹数量;FP 是假正例的数量,即所有不属于某终端的指纹中识别成功的指纹数量.TPR 被用来评估指纹标识设备身份的能力,TNR 则被用来评估设备识别框架检测携带虚假身份信息的设备的能力.

为了清晰地描述测试指标的计算方法,以接入时设备识别中设备 1、设备 2 和设备 3 的 PAID 指纹构成的测试集为例.当测试设备 1 的 TPR 时,先用设备 1 的前 10 00 个 PAID 指纹训练 PAID 指纹匹配网络,然后利用该网络识别剩余 7 000 个 PAID 指纹,此时 $TP+FN=7\,000$,TP 为识别成功的 PAID 指纹的数量.当测试设备 1 的 TNR 时,首先用设备 1 的前 1 000 个 PAID 指纹训练 PAID 指纹匹配网络,然后利用该网络识别设备 2 和设备 3 的 16 000 个 PAID 指纹,此时 $TN+FP=16\,000$,TN 为识别失败的指纹的数量.

5.3 接入时的设备识别性能分析

由 3.1 节的 PAID 指纹生成方案可知,影响指纹生成的变量有数据包到达时间间隔序列 PAI 的长度 L,以及 PAID 指纹长度 SEG. L 越大,则用于提取 PAID 指纹的数据包($P_1,P_2,\cdots,P_{L+1}$)就越多,因此采集 PAID 指纹所需的时间就越长,进而增加接入时设备识别的用时.但是,L 越小则数据包就越少,数据包到达时间的统计特征可能会淹没在退避算法和操作系统调度带来的随机性当中,因此需要选择合适的 L 来生成 PAID 指纹.为此,在 $SEG=20$ 时利用不同的 L 开展了识别准确率测试,测试数据集为设备 10、设备 11 和设备 12 的 PAID 指纹,TPR,TNR 分别为 3 台设备的 TPR,TNR 的平均值.测试结果如图 18 所示.可以看出,L 不同时 $TNR>0.99$,TPR 仅在 0.01 的范围内变化:当 $L\leqslant 1\,400$ 时,TPR 超过 0.97,而当 $L>1\,400$ 时,TPR 下降至 0.97 以下.考虑到 L 越小,PAID 指纹采集速度就越快,本文在后续测试中令 $L=1\,000$.

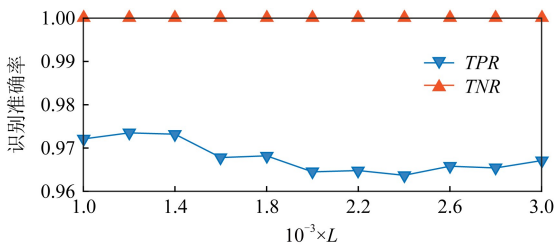


Fig. 18 Identification accuracy corresponding to different L

图 18 不同 L 对应的识别准确率

SEG 为 PAID 指纹的长度,也为计算分段密度时数据包到达时间间隔序列 PAI 的分段数量.较小的 SEG 会导致分段密度失真,无法体现出 PAI 的密度特征,进而降低不同设备间 PAID 指纹的区分度;过大的 SEG 则会增大 PAID 指纹匹配网络的规模,进而降低接入时设备识别的时间效率.因此,需要选择合适的 SEG 来生成 PAID 指纹.为此,利用不同的 SEG 测试了识别准确率,测试方法与图 18 相同.图 19 给出了不同 SEG 对应的识别准确率(每个 SEG 对应的 I 均为 5).可以看出,当 $10\leqslant SEG\leqslant 15$ 时,TNR 始终接近 1,TPR 呈现上升趋势;当 $SEG\leqslant 20$ 时,TPR 的增长较快;而当 $SEG>20$ 时,TPR

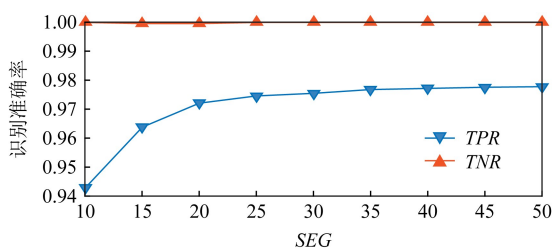


Fig. 19 Identification accuracy corresponding to different SEG

图 19 不同 SEG 对应的识别准确率

则缓慢增长,且始终低于 0.98.考虑到 SEG 越小,PAID 指纹匹配速度就越快,因此在后续测试中令 $SEG=20$.

为选取接入时设备识别的判别阈值 $threshold_{ac}$,计算了任意 2 台设备的 PAID 指纹的匹配结果,计算方法为:针对设备 $i(i=1,2,\cdots,18)$和设备 $j(j=1,2,\cdots,18)$组成的测试设备对(设备 i,设备 j),以设备 i 的前 1000 个 PAID 指纹为训练集训练 PAID 指纹匹配网络,然后以设备 j 的 PAID 指纹作为测试集进行指纹匹配,将 8 000 个指纹的匹配结果的均值作为最终结果.图 20 显示了所有测试设备对的匹配结果,可以看出,设备与自身的匹配结果为 0.15 左右,与其他设备的匹配结果则大于 0.3,因此将 $threshold_{ac}$ 设置为 0.2.

	设备1	设备2	设备3	设备4	设备5	设备6	设备7	设备8	设备9	设备10	设备11	设备12	设备13	设备14	设备15	设备16	设备17	设备18
设备1	0.16	0.50	0.32	1.33	1.30	1.21	1.07	1.23	1.28	1.02	1.38	1.31	1.73	1.63	1.73	1.51	1.68	1.57
设备2	0.49	0.16	0.50	1.72	1.62	1.55	1.35	1.48	1.45	1.28	1.72	1.70	1.99	1.83	1.88	1.69	1.84	1.66
设备3	0.32	0.50	0.16	1.34	1.28	1.24	1.04	1.27	1.34	1.04	1.40	1.26	0.83	1.60	1.71	1.40	1.61	1.44
设备4	1.39	1.65	1.44	0.15	0.49	0.49	0.50	0.34	0.53	0.58	0.44	0.39	0.59	0.73	0.68	0.88	0.89	0.97
设备5	1.29	1.53	1.30	0.49	0.15	0.30	0.27	0.38	0.47	0.37	0.46	0.40	0.49	0.47	0.58	0.29	0.43	0.31
设备6	1.12	1.38	1.16	0.50	0.30	0.15	0.35	0.34	0.37	0.35	0.49	0.49	1.13	0.40	0.46	0.36	0.36	0.48
设备7	1.35	1.61	1.32	0.68	0.42	0.52	0.14	0.52	0.68	0.53	0.77	0.42	0.36	0.78	0.80	0.47	0.55	0.55
设备8	1.20	1.50	1.27	0.47	0.51	0.44	0.45	0.14	0.40	0.51	0.47	0.38	0.74	0.49	0.35	0.67	0.52	0.76
设备9	1.07	1.30	1.13	0.69	0.53	0.40	0.52	0.40	0.14	0.36	0.62	0.65	0.74	0.69	0.72	0.75	0.70	0.79
设备10	1.12	1.38	1.15	0.66	0.44	0.39	0.45	0.45	0.41	0.16	0.58	0.62	0.50	0.69	0.78	0.62	0.64	0.66
设备11	1.61	1.86	1.67	0.43	0.44	0.47	0.56	0.34	0.52	0.60	0.15	0.52	1.02	0.61	0.81	0.92	0.91	0.96
设备12	1.18	1.42	1.17	0.50	0.53	0.58	0.36	0.33	0.50	0.56	0.66	0.14	1.75	0.64	0.56	0.69	0.66	0.80
设备13	1.66	1.98	1.90	0.83	0.59	0.48	0.45	0.79	0.69	0.65	0.92	1.06	0.14	0.42	0.45	0.67	0.57	0.76
设备14	1.53	1.80	1.66	0.79	0.52	0.44	0.77	0.49	0.74	0.56	0.69	0.63	0.49	0.13	0.27	0.50	0.42	0.66
设备15	1.58	1.90	1.74	0.72	0.62	0.49	0.71	0.35	0.66	0.64	0.87	0.56	0.48	0.27	0.14	0.54	0.38	0.68
设备16	1.35	1.88	1.55	0.92	0.33	0.41	0.49	0.47	0.76	0.50	0.98	0.63	0.70	0.51	0.52	0.14	0.36	0.45
设备17	1.51	1.78	1.66	0.93	0.46	0.40	0.46	0.35	0.75	0.56	0.97	0.62	0.60	0.43	0.38	0.36	0.14	0.24
设备18	1.42	1.82	1.40	1.02	0.35	0.52	0.44	0.52	0.78	0.52	1.03	0.67	0.79	0.68	0.66	0.44	0.23	0.15

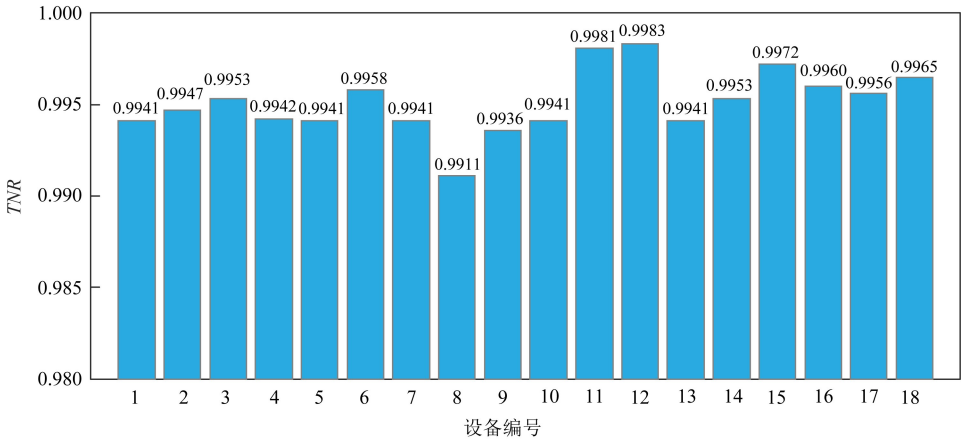
Fig. 20 Matching results of all devices pairs

图 20 所有测试设备对的匹配结果

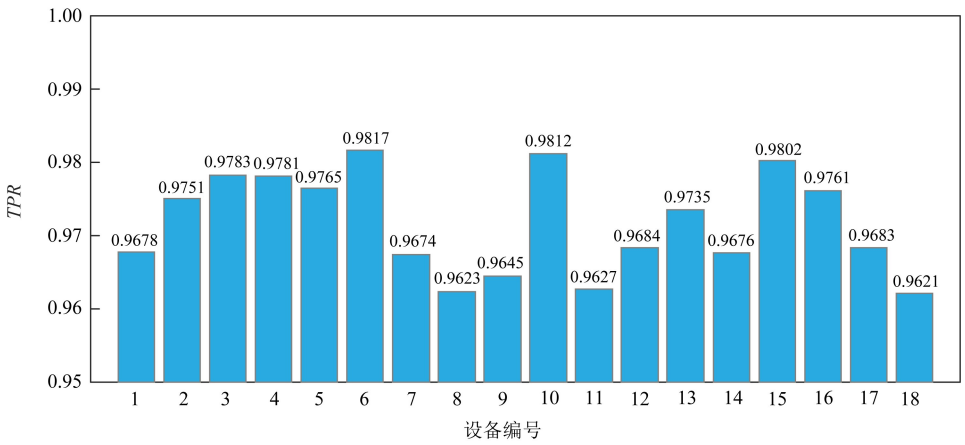
最终,对 18 台设备分别进行了准确率测试,测试数据集为所有设备的 PAID 指纹集.图 21 为测试结果.设备 6 的 TPR 最高,约为 0.981 7;设备 8 的 TPR 最低,约为 0.962 3.进一步地,设备 4、设备 5 和设备 6 的 TPR 均高于 0.976,设备 7、设备 8 和设备 9 的 TPR 均低于 0.968,设备 1、设备 2 和设备 3 的 TPR 则约为 0.97,因此相比于台式机,本文所提供的 PAID 指纹能够更好地标识树莓派和路由器的硬件特征.此外,设备 10~16 的 TPR 均高于 0.96,因此该 PAID 指纹同样适用于手机、平板电脑和笔记本电脑.此外,所有设备的 TNR 均高于 0.99,表明该 PAID 指纹能够标识设备间的硬件特征差异,进而能够以高准确率检测出携带虚假身份信息的无线终端.

5.4 通信时的实时设备识别性能分析

在通信时的实时设备身份识别中,由于 CSI 指纹具有固定的维度,在带宽为 20 MHz 的 WiFi 网络中,CSI 指纹的维度为 52,因此影响识别准确率的主要因素为 CSI 指纹匹配网络中自动编码器的个数 J.图 22(a)展示了 $J=13,6,4,2,1$ 时的识别准确率,随着 J 的增加,TNR 近似于 1,TPR 则逐渐增大.CSI 指纹匹配网络中的 J 个自动编码器分别用于匹配 J 个子 CSI 指纹,当 J 增加时,CSI 指纹被拆分为更多的子指纹,因此匹配目标更加局部,噪声干扰导致的 CSI 指纹波动则被进一步地放大,进而降低了识别准确率;当 J 减小时,子指纹长度增加,匹配目标趋于指纹整体,因此能够更好地学习指纹的特征,并且削弱噪声的干扰.



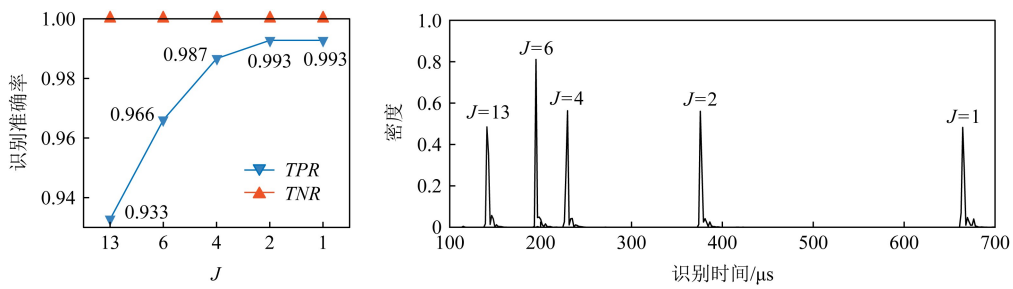
(a) 设备的*TNR*



(b) 设备的*TPR*

Fig. 21 Identification accuracy of all devices

图 21 所有测试设备的识别准确率



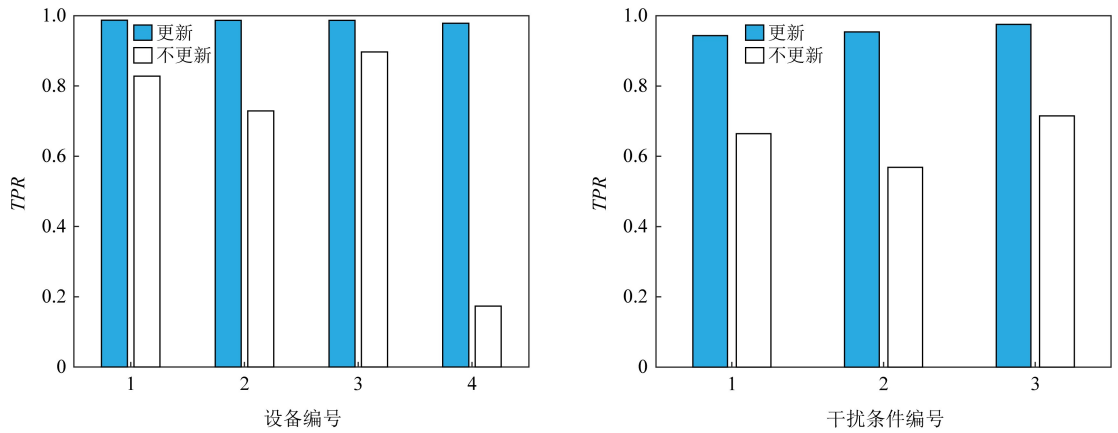
(a) 识别准确率 (b) 单个数据包在不同J下的识别时间密度图

Fig. 22 Identification accuracy for different J and identification time distribution of a single packet

图 22 不同 J 对应的识别准确率和单个数据包的识别时间分布图

但是,由 4.6 节的指纹匹配网络时间复杂度分析可知,随着 J 的降低,匹配网络的匹配时间会随之增大,单个数据包的识别时间也随之增加.图 22(b)显示了 10 000 个 CSI 指纹的指纹匹配时间的分布密度图,与 4.6 节的理论推导一致.过长的匹配时间可能会导致数据包到达速率与识别速度不匹配,进而无法实现实时的逐包识别,因此需要选取合适的 J,在提高准确率的同时保证识别的实时性.考虑到 $J=4$ 时的 TPR 仅比 $J=2$ 时低 0.006,识别时间则近似为后者的 1/2,在后续测试中令 $J=4$,此时子 CSI 指纹的长度为 13.

图 23(a)展示了公寓测试场景中位于 4 个不同位置的无线终端的 TPR.前 3 个位置在公寓内,3 台终端的 TPR 均高于 0.986;被放置在走廊中的终端



(a) 公寓测试场景 (b) 实验室测试场景

Fig. 23 Identification accuracy in two test scenarios

图 23 2 种测试场景下的识别准确率

的 *TPR* 下降到 0.978 7.在公寓测试场景中,终端 4 的 CSI 样本的方差为 3.985×10^3,远高于前 3 个位置的方差.以上测试结果表明,终端与识别设备的距离不是影响通信时设备识别准确率的直接因素.尽管随着距离的增大,存在干扰的可能性也会增加,但是在物体移动、人员走动较少的室内环境中可以忽略这种影响.走廊中的终端的 CSI 指纹稳定性较差,这是因为 2 个原因:

1) 终端与识别设备没有视距(line of sight, LOS)无线传播路径;

2) 走廊有行人走动干扰.因此,本文给出的通信时逐包设备识别框架更适用于存在 LOS 路径、较为稳定的室内环境.

图 23(b)展示了实验室测试场景中,3 种不同干扰条件下无线终端的 *TPR*.随着人员走动路径的远离,*TPR* 从 0.943 6 增长到 0.975 3.在实验室测试场景下,3 种干扰条件对应的 CSI 样本方差分别为 10.084×10^3,7.099×10^3 和 6.941×10^3,因此,当人员走动路径远离时,无线终端和识别设备受到的干扰降低,同一终端的 CSI 指纹更加稳定,通信时设备识别准确率也更高.这表明环境干扰会降低通信时逐包设备识别框架的性能,但是在受到较大干扰(行人在距终端 1 m 处走动)时,通信时逐包识别准确率仍能够达到 0.94.以上的测试均使用了 CSI 指纹匹配网络更新模块.为了测试该模块对通信时逐包设备识别准确率的改善效果,在不启用该模块的条件下测试了无线终端的 *TPR*,测试结果如图 23 所示.在干扰较小的环境中(公寓测试场景中的位置 1,2,3),*TPR* 最低降至 0.729 1,比启用更新模块时降低了 0.257 7.在干扰较大的环境中(公寓测试场景中的位置 4 和实验室测试场景),*TPR* 最低降至 0.173 7,比启用更新模块时降低了 0.805.因此,CSI 指纹匹配网络更新模块虽然增大了通信时逐包设备识别时间,但是能够改善设备识别的性能,这种改善效果在干扰环境下更为显著.

6 总 结

本文提出了一种基于混合特征指纹的设备身份识别方案,该方案将 2 种流量特征指纹分别用于 2 种设备识别场景:1)在终端请求接入时,该方案利用 PAID 生成指纹并进行识别;2)在终端成功接入并开始通信时,该方案利用来自终端的每个数据包的 CSI 生成指纹,从而实现实时的逐包设备识别.同时,本文提出了一种计算复杂度较低的指纹匹配网络,以保证在计算能力有限的设备中也能快速且准确地识别设备.

为了解决 CSI 指纹会随着终端的位置或所处环境的改变而改变的问题,本文提出了一种基于混合特征指纹的设备身份识别方案.该方案从无线终端的 PAID 和 CSI 中提取 2 类指纹:前者为 PAID 指纹,标识设备的硬件特征;后者为 CSI 指纹,标识设备的无线信道特征.该方案包含两阶段的设备识别:一是当终端请求接入无线网络时,识别设备捕获该终端发送的若干数据包,从中提取 PAID 指纹和 CSI 指纹,并用 PAID 指纹进行识别,识别成功则用新获取的 CSI 指纹进行后续的 CSI 指纹匹配;二是当终端接入并开始通信时,识别设备从终端的每个数据包中提取 CSI 指纹并进行实时的逐包身份识别.

为了解决现有的利用机器学习进行指纹匹配的设备识别技术计算复杂度较高,无法在计算能力有限的嵌入式设备中实现的问题,本文提出了一种改进的指纹匹配方案,该方案利用基于自动编码器的指纹匹配网络来进行指纹匹配.该网络包括匹配层和输出层,前者由小规模自动编码器并联组成,后者由线性变换器构成,在保证高识别准确率的同时降低了计算复杂度.

在树莓派 3B+上实现了基于混合特征指纹的设备识别原型系统,具体实现了无线终端接入时的 PAID 指纹提取、CSI 指纹提取、PAID 指纹匹配,以及通信时的逐包 CSI 指纹提取和 CSI 指纹匹配.利用树莓派和无线终端在公寓和实验室开展了设备识别实验,实验表明:该系统在接入时和通信时均能达到较高的设备识别准确率.

参 考 文 献

[1] Bendale S P, Prasad J R. Security threats and challenges in future mobile wireless networks [C] //Proc of IEEE Global Conf on Wireless Computing and Networking. Piscataway, NJ: IEEE, 2018: 146-150

[2] Shah S, Bendale S P. An intuitive study: Intrusion detection systems and anomalies, how AI can be used as a tool to enable the majority, in 5G era [C] //Proc of the 5th Int Conf On Computing, Communication, Control and Automation. Piscataway, NJ: IEEE, 2019: 1-8

[3] Vashist A, Keats A, Pudukotai D S M, et al. Securing a wireless network-on-chip against jamming-based denial-of-service and eavesdropping attacks [J]. IEEE Transactions on Very Large Scale Integration Systems, 2019, 27(12): 2781-2791

[4] Chatterjee U, Sadhukhan R, Mukhopadhyay D, et al. Stupify: A hardware countermeasure of KRACKs in WPA2 using physically unclonable functions [C] //Proc of the Web Conf 2020 (WWW'20). New York: ACM, 2020: 217-221

[5] Kim W, Kim S, Lim H. Malicious data frame injection attack without seizing association in IEEE 802.11 wireless LANs [J]. IEEE Access, 2021, 9: 16649-16660

[6] Lounis K, Zulkernine M. Attacks and defenses in short-range wireless technologies for IoT [J]. IEEE Access, 2020, 8: 88892-88932

[7] Pimple N, Salunke T, Pawar U, et al. Wireless security—An approach towards secured WiFi connectivity [C] //Proc of the 6th Int Conf on Advanced Computing and Communication Systems. Piscataway, NJ: IEEE, 2020: 872-876

[8] Liao Runfa, Wen Hong, Pan Fei, et al. A novel physical layer authentication method with convolutional neural network [C] //Proc of the 2019 IEEE Int Conf on Artificial Intelligence and Computer Applications. Piscataway, NJ: IEEE, 2019: 231-235

[9] Lin Yuxiang, Gao Yi, Li Bingji, et al. Accurate and robust rogue access point detection with client-agnostic wireless fingerprinting [C] //Proc of IEEE Int Conf on Pervasive Computing and Communications. Piscataway, NJ: IEEE, 2020: 1-10

[10] Perazzone J B, Yu P L, Sadler B M, et al. Artificial noise-aided MIMO physical layer authentication with imperfect CSI [J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 2173-2185

[11] Liu Hongbo, Wang Yan, Liu Jian, et al. Authenticating users through fine-grained channel information [J]. IEEE Transactions on Mobile Computing, 2018, 17(2): 251-264

[12] Xi Wei, Qian Chen, Han Jinsong, et al. Instant and robust authentication and key agreement among mobile devices [C] //Proc of the 2016 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2016: 616-627

[13] Pan Fei, Pang Zhibo, Luvisotto M, et al. Authentication based on channel state information for industrial wireless communications [C] //Proc of the 44th Annual Conf of the IEEE Industrial Electronics Society. Piscataway, NJ: IEEE, 2018: 4125-4130

[14] Wang Wei, Chen Yingjie, Zhang Qian. Privacy-preserving location authentication in WiFi networks using finegrained physical layer signatures [J]. IEEE Transactions on Wireless Communications, 2016, 15(2): 1218-1225

[15] Liu Pengfei, Yang Panlong, Song Wenzhan, et al. Real-time identification of rogue WiFi connections using environment-independent physical features [C] //Proc of IEEE Conf on Computer Communications (INFOCOM 2019). Piscataway, NJ: IEEE, 2019: 190-198

[16] Jiang Zhiping, Zhao Jizhong, Li Xiangyang, et al. Rejecting the attack: Source authentication for WiFi management frames using CSI information [C] //Proc of the IEEE INFOCOM. Piscataway, NJ: IEEE, 2013: 2544-2552

[17] Liu Rui, Li Yang, Zhang Mingxuan, et al. The wireless IoT device identification based on channel state information fingerprinting [C] //Proc of the 9th Joint Int Information Technology and Artificial Intelligence Conf. Piscataway, NJ: IEEE, 2020: 534-541

[18] Liu Muye, Mukherjee A, Zhang Zhenghao, et al. Tbas: Enhancing WiFi authentication by actively eliciting channel state information [C] //Proc of the 13th Annual IEEE Int Conf on Sensing, Communication, and Networking. Piscataway, NJ: IEEE, 2016: 1-9

[19] Wan Xiaoyue, Xiao Liang, Li Qiangda, et al. FHY-layer authentication with multiple landmarks with reduced communication overhead [C] //Proc of IEEE Int Conf on Communications (ICC). Piscataway, NJ: IEEE, 2017: 1-6

[20] Cheema A R, Alsmadi M, Ikki S. Survey of identity-based attacks detection techniques in wireless networks using received signal strength [C] //Proc of IEEE Canadian Conf on Electrical & Computer Engineering. Piscataway, NJ: IEEE, 2018: 1-6

[21] Zhang Aiguo, Yuan Ying, Wu Qunyong, et al. Wireless localization based on RSSI fingerprint feature vector [J]. International Journal of Distributed Sensor Networks, 2015, 11(11): 528-747

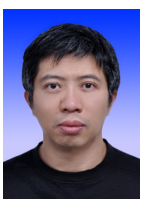
[22] Li Zhen, Luo Lingen, Sheng Gehao, et al. UHF partial discharge localisation method in substation based on dimension-reduced RSSI fingerprint [J]. IET Generation, Transmission & Distribution, 2018, 12(2): 398-405

[23] Xie Ning, Chen Junjie, Huang Lei. Physical-layer authentication using multiple channel-based features [J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 2356-2366

[24] Mahmood A, Aman W, Iqbal M O, et al. Channel impulse response-based distributed physical layer authentication [C] //Proc of the 85th Vehicular Technology Conf. Piscataway, NJ: IEEE, 2017: 1-5

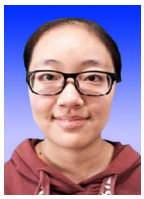
[25] Iqbal A, Jeoti V, Drieberg M, et al. A time-domain channel impulse response estimation method for an OFDM sounding system [C] //Proc of the Int Conf on Smart Instrumentation, Measurement and Application. Piscataway, NJ: IEEE, 2019: 1-5

[26] Gringoli F, Schulz M, Link J, et al. Free your CSI: A channel state information extraction platform for modern WiFi chipsets [C] //Proc of the 13th Int Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization. Piscataway, NJ: IEEE, 2019: 21-28



Song Yubo, born in 1977. PhD, associate professor. His main research interests include communication network security, network security protocol design, and user privacy protection.

宋宇波,1977 年生.博士,副教授.主要研究方向为通信网络安全、网络安全协议设计和用户隐私保护.



Chen Bing, born in 1999. Master. Her main research interest is communication network security.

陈　冰,1999 年生.硕士.主要研究方向为通信网络安全.

Zheng Tianyu, born in 1998. Master. His main research interest is communication network security.

郑天宇,1998 年生.硕士.主要研究方向为通信网络安全.

Chen Hongyuan, born in 1987. Master. His main research interest is communication network security.

陈宏远,1987 年生.硕士.主要研究方向为通信网络安全.

Chen Liquan, born in 1976. PhD, professor. His main research interests include mobile information security, IoT system and security, cloud computing and big data security, information hiding and digital watermarking.

陈立全,1976 年生.博士,教授.主要研究方向为移动信息安全、物联网系统与安全、云计算及大数据安全、信息隐藏与数字水印.

Hu Aiqun, born in 1964. PhD, professor. His main research interests include network and information security, mobile communication security technology.

胡爱群,1964 年生.博士,教授.主要研究方向为网络与信息安全、移动通信安全技术.