This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3186646

AuthIoT: A Transferable Wireless Authentication Scheme for IoT Devices without Input Interface

Shichen Zhang, Pedram Kheirkhah Sangdeh, Hossein Pirayesh, Huacheng Zeng, Qiben Yan, and Kai Zeng

***Abstract*—Wireless Internet-of-Things (IoT) applications have penetrated every aspect of our society and become increasingly important in smart homes, smart cities, and smart hospitals. However, many WiFi-based IoT devices (e.g., light switches, door/window open alert sensors, and Google Home) do not have input interfaces such as keypad or touchscreen due to their limits in physical size, power consumption, and/or manufacturing cost, making it inconvenient and onerous for end users to authenticate those IoT devices for wireless Internet access. In this paper, we present AuthIoT, a learning-based authentication scheme for wireless IoT devices without input interfaces. The key component of AuthIoT is a channel state information (CSI) based character classification algorithm for a WiFi access point (AP), which recognizes the passcode from an IoT device when an end user holds it in hand and writes the passcode over the air. AuthIoT has two salient features: i) it is transferable for cross-environment applications; and ii) it works in more realistic scenarios where the AP is equipped with nonlinear antenna array. We have built a prototype of AuthIoT and evaluated its performance on two testbeds: Intel 5300 WiFi card with three linear antennas and USRP N310 with four nonlinear (square-shaped) antennas. Experimental results show that AuthIoT achieves 84% and 83% recognition accuracy on the two testbeds.**

***Index Terms*—Internet of Things, IoT authentication, wireless sensing, letter recognition, deep learning**

I. Introduction

The Internet-of-Things (IoT) applications have penetrated every aspect of our society and have significantly facilitated our lives. Per Statista [1], the number of IoT devices worldwide is forecast to almost triple from 8.74 billion in 2020 to more than 25.4 billion in 2030. In real-world applications, many IoT devices rely on WiFi connections for Internet access and have no input interfaces (e.g., keypad or touchscreen) due to their limits in physical size, power consumption, and/or manufacturing cost. For example, smart home devices such as Gosund Smart WiFi power outlet [2], SYLVANIA WiFi dimmable LED light bulb [3], and AGSHOME WiFi Windows open alert sensors [4] require WiFi network access to be functional, but they have no input interfaces which end users can use to type in WiFi passcode for wireless Internet access. With the proliferation of small-sized wireless sensors in smart environments, the presence of wireless IoT devices without input interfaces will become commonplace.

S. Zhang, P. Kheirkhah Sangdeh, H. Pirayesh, H. Zeng and Q. Yan are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824.

K. Zeng is with Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA 22030.

The work of S. Zhang, P. Kheirkhah Sangdeh, H. Pirayesh, and H. Zeng was supported by the NSF under Grant No. 2100112. The work of K. Zeng was supported in part by the NSF under Grant No. 2131507, Microsoft Research Award, and the Commonwealth Cyber Initiative (CCI) and its Northern Virginia (NOVA) Node, an investment in the advancement of cyber R&D, innovation, and workforce development.

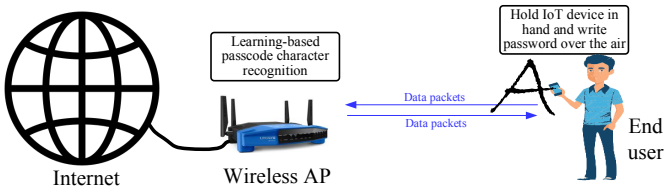


Fig. 1: A CSI-based authentication scheme for wireless IoT devices without input interfaces.

A prevalent method for authenticating WiFi-based IoT devices without input interfaces is by leveraging pre-deployed platforms such as Google Home Assistant [5] and Amazon Alex [6], which allows a WiFi router to recognize and authenticate an IoT device using a smartphone or computer. This method, however, requires end users to have a smartphone/computer with pre-installed proprietary apps such as Google Home and Amazon Alexa. It also requires Internet ready for use to gain the support of Google or Amazon cloud services. These requirements make this method inapplicable in the scenarios where a smartphone or Internet is not available and where the IoT device owners do not want to get involved in commercial cloud platforms.

In this paper, we present AuthIoT, a wireless communication authentication scheme for IoT devices without input interfaces. AuthIoT requires neither assistance from other devices nor support from an Internet-based software platform. It is a channel state information based (CSI-based) passcode recognition scheme for a WiFi communication system, as shown in Fig. 1. It consists of an access point (AP), an IoT device, and an end user. Specifically, AuthIoT works as follows: The end user holds the IoT device in hand and writes the passcode over the air; and the AP leverages recent advances in deep learning to recognize the passcode input from the IoT device based on the spatial and temporal CSI features.

A key challenge in the design of AuthIoT is to maintain its transferability across different environments. As CSI is significantly affected by the multipath effect of a wireless channel, a wireless AP tends to observe different CSI in different environments. Hence, at the wireless AP, using raw CSI for passcode recognition is not a plausible strategy because a deep neural network (DNN) trained with raw CSI in an environment does not work well in another environment (based on our experimental results). To address this challenge, AuthIoT extracts environment-independent features as the input for the training and inference of a DNN. Specifically, AuthIoT computes the angle of arrival (AoA) of the line-of-sight (LoS) signal path by leveraging recent advances in wireless localization [12], [13], [14], [15], [16], and uses the AoA (as well as normalized

TABLE I: Wireless writing and gesture recognition.

Ref.	(Tx,Rx) ant #	Nonlinear antenna array	Dataset	main approach	Learning features	computation complexity	cross-environment transferability	Reported accuracy
WriFi [7]	(2,3)	No	26 capital letters	GMM-HMMs	CSI amplitude	High	No	87%
WiReader [8]	(1,2)	No	26 capital letters	LSTM model	CSI amplitude	Medium	No	90%
LetFi[9]	(1,6)	No	26 capital letters	SOM network	CSI amplitude	Medium	No	95%
WiDraw [10]	(30,3)	No	Any	Trajectory tracking	AoA	Medium	Yes	91%
Wi-Wri [11]	(2,3)	No	26 capital letters	kNN model	CSI amplitude	High	No	82%
AuthIoT	(1,3 or 4)	Yes	48 characters	CNN-based learning	LoS AoA, CSI amplitude	Medium	Yes	84%

channel amplitude) as the input for the training and inference of DNN. Since different passcode characters tend to generate different AoA patterns, an AP is capable of recognizing the passcode characters if the DNN is well trained.

Another challenge in the design of AuthIoT is to compute the LoS AoA of received packets for an AP with a nonlinear antenna configuration. While AoA estimation of wireless packets has been studied in wireless localization (e.g., [12], [17], [13], [14], [15], [16]), most of existing techniques deal with the case where the antenna elements are equally-spaced and linearly installed. However, many WiFi routers and other APs are equipped with antenna elements in a nonlinear shape so as to save space. Existing methods such as MUSIC (MUltiple SIgnal Classification) algorithm cannot be directly used to estimate AoA for a receiving device with nonlinear antenna configuration. To address this challenge, AuthIoT extends two-dimensional MUSIC algorithm to the case where the receiver (wireless AP) is equipped with nonlinear antenna elements. Following the idea from SpotFi [12], AuthIoT jointly considers the AoA and ToF (time of flight) to enhance the AoA resolution of different signal paths.

Based on the environment-independent features (LoS AoA) as well as the normalized amplitude of CSI, AuthIoT employs a DNN to recognize the passcode when an end user continuously writes the passcode characters over the air by holding the IoT device in her hand. Once the AP detects the passcode, it will grant the network access to the IoT device; otherwise, it will wait until the correct passcode is detected or the maximum number of attempts is reached. We have built a prototype of AuthIoT and evaluated its performance on two distinct AP testbeds: i) Intel 5300 WiFi card with three linear antennas, and ii) USRP N310 with four nonlinear (square-positioned) antennas. Experimental results show that AuthIoT achieves 84% successful rate of passcode character recognition on the former testbed and 83% successful rate on the latter testbed, both for cross-environment applications.

The contributions of this paper are summarized as follows.

- AuthIoT is, to the best of our knowledge, among the first that explores environment-independent features of CSI for authenticating IoT devices without input interfaces. It is transferable to a new environment for handwriting recognition once its DNN is well trained.
- AuthIoT extends two-dimensional MUSIC algorithm for AoA estimation from linear, equally-spaced antenna configuration to nonlinear antenna configuration.
- We have built a prototype of AuthIoT and demonstrated its performance in real scenarios. Our experimental results show that it can achieve more than 83% passcode recognition accuracy in cross environments for both linear and nonlinear antenna configurations.

II. Related Work

We survey the literature in the following category.

Authenticating IoT Devices without Input Interface: As mentioned before, a mainstream authentication method for smart-home IoT devices is to leverage the platforms such as Google Home [5] and Amazon Alex [6]. This method, however, requires users to have a smartphone with pre-installed proprietary apps, to have Internet access, and to share the data with the platforms. In addition to the commercial products, research advances have been made for IoT authentication.

TouchAuth [18] harnesses induced body electric potentials (iBEPs) for IoT authentication by having users wear a wristband to touch an analog-to-digital (ADC) pin of the IoT device. It makes the ADC pin touchable by connecting devices' ADC pins to their conductive exteriors. The authentication is performed by measuring the IBEPs similarities between the wristband and the smart object. P2Auth [19] authenticates IoT devices without input interface by leveraging their inertial measurement unit. It requires users to perform unique petting operations that can be sensed by both an IoT device and a wristband device. It compares the captured data from the two devices and makes a decision for the authentication based on their similarity. SFIRE [20] is a secret-free trust establishment protocol that pairs commercial wireless devices with a hub. It requires a user to move a helping smartphone around the wireless device and measures the similarity of RSS signals for authentication. Move2Auth [21] is another authentication scheme for IoT devices without an input interface. It requires users to hold a smartphone and perform one of two hand-gestures in front of an IoT device.

In contrast to the above works, AuthIoT takes a very different approach to authenticate IoT devices without input interface. It requires neither assistance from smartphones nor hardware/software modifications on IoT devices.

CSI-based Handwriting Recognition: Our work is closely related to the research in this area. Table I presents a comparison of our work with prior work. WriFi [7] is a CSI-based handwriting system that comprises a WiFi AP, a WiFi client device, and a user writing 26 letters over the air. In this system, CSI amplitude is collected for learning-based recognition. Operations such as principal component analysis (PCA) and fast fourier transform (FFT) have been performed to extract the CSI features for hidden Markov model (HMM) training and inference. The accuracy is reported to be 86%. Similar to WriFi, Wi-Wri [11] is another CSI-based handwriting letter

recognition system. It is based on k-nearest neighbors (k-NN) model and uses dynamic time warping (DWT) to calculate the distance between CSI waveform and classified data. It reports 83% recognition accuracy for 26 letters. WiReader [8] is another work in this area. It exploits CSI from commercial WiFi devices to extract activities-related information. It employs long short-term memory (LSTM) model for recognition and adopts PCA and discrete wavelet transform (DWT) for CSI feature extraction. It reports 90% recognition accuracy for 26 letters with intelligence text correction. LetFi [9] is also a CSI-based over-the-air handwriting recognition system in WiFi networks. It employs multi-domain feature extraction method and self-organizing mapping neural networks (with SoftMax regression classifier) to recognize 26 letters. The reported recognition accuracy is 95%. WiDraw [10] is a handwritten recognition system which allows a user to write over the air. It recognizes hand movement trajectory based on the analysis of collected CSI. With the presence of 30 transmitters, it can achieve 91% word recognition accuracy and superior accuracy for hand movement patterns.

As shown in Table I, AuthIoT differs from the above works in several aspects: i) AuthIoT has a larger dataset (48 characters in AuthIoT versus 26 letters in the above-mentioned works); ii) it enables its cross-environment transferability by design; and iii) it works for WiFi AP with nonlinear antenna array.

CSI-based Gesture Recognition: In addition to handwriting recognition, many works have also been done for CSI-based gesture recognition [22], [23], [24], [25], [26], by extracting and recognizing the temporal, spatial, and Doppler features of hand movements. Generally speaking, CSI-based gesture recognition can achieve very high accuracy (over 90%), because it has a small number of dataset (e.g., 6 gestures). In contrast, AuthIoT has 48 characters in its dataset, which is much larger than the above networks. In addition, AuthIoT distinguishes itself from previous works by focusing on cross-environment transferability design.

Wireless Localization: Another research line related to our work is CSI-based wireless localization in WiFi networks [12], [14], [17], [27], [16]. Particularly, SpotFi [12] presents an accurate indoor localization scheme using commercial WiFi devices. It proposes a two-dimensional MUSIC algorithm by leveraging the information in both spectral and spatial domains to enhance the resolution of AoA estimation. It jointly estimates AoA and ToF (time of flight) of incoming WiFi signals using multiple antennas and broadband (40MHz) spectrum. The localization median accuracy is reported to 40cm using the commercial WiFi card. AuthIoT borrows the idea of AoA estimation from the above works, and extends the antenna setting from linear to nonlinear case for IoT authentication applications.

III. AuthIoT: Design Overview

A. System Setting and Operation

AuthIoT is designed for a wireless communication system as shown in Fig. 1, which comprises a wireless AP (e.g., WiFi router), an IoT device, and an end user. IoT devices do not have input interfaces such as keypads and touchscreens due to the limits in their physical size, power consumption, and/or manufacturing cost. Examples of such IoT devices include WiFi LED light bulbs [3], WiFi light switches [28], and window/door open alert sensors [29]. The wireless AP has multiple antennas for data packet reception. This is very common for WiFi routers, most of which are equipped with four or more antennas. In such a system, AuthIoT works as follows.

- **End User:** The end user first triggers wireless AP to exchange packets between itself and the IoT device at a certain rate (e.g., 200 packets/s). She then holds the IoT device in front of the wireless AP with a distance of about 2 meters and ensures that there is a LoS signal path between the IoT device and the wireless AP. After that, the end user writes each of the passcode characters over the air until the IoT device is successfully authenticated.
- **IoT Device:** The IoT device needs no hardware or software modification. It responds to the sounding packets from the wireless AP (e.g., using ACK packets) so that the wireless AP can estimate wireless channel at a desired rate.
- **Wireless AP:** The wireless AP estimates the channel between itself and the IoT device using the packets from the IoT device. It continuously runs a modified MUSIC algorithm to estimate the LoS AoA of the packets from the IoT device and feed the LoS AoA along with normalized CSI amplitude to a DNN for the recognition of each character in the passcode. It authenticates the IoT device once the passcode is detected or the maximum number of attempts is reached.

B. Challenges and Our Approach

Compared to prior CSI-based recognition work [8], [11], [7], [10], [9], AuthIoT needs to recognize a much larger set of characters, which include upper-case letters, low-case letters, numbers, and special characters. In addition, AuthIoT faces the following challenges in its design and implementation.

Cross-Environment Transferability: A challenge in the design of AuthIoT is to maintain its cross-environment transferability, so that the system can be used in any environment once its DNN has been trained. To address this challenge, AuthIoT uses environment-independent CSI features as its input for passcode recognition. Specifically, it computes the LoS AoA of the received packets from the IoT device based on the estimated CSI by leveraging recent advances in wireless localization [12], [17], [14], [16], and uses the LoS AoA as the main feature for passcode recognition. It should be noted that an end user can always hold the IoT device in front of its wireless AP to ensure the existence of LoS path between the IoT device and its AP.

Nonlinear Antenna Array at AP: Although the LoS AoA estimation techniques have been well studied for wireless localization, most of them consider the case where the receiver is equipped with linearly, equally-spaced antenna array [12], [17], [14], [16]. However, many off-the-shelf wireless APs such as WiFi routers are equipped with nonlinear antenna

array (e.g., rectangular-installed) to save space. As expected, the AoA estimation techniques proposed for a device with linear antenna array cannot be directly used for a device with nonlinear antenna array. To address this challenge, AuthIoT revisits the MUSIC algorithm and extends it for the case where the device has nonlinear antenna array. AuthIoT also borrows the idea from SpotFi [12] to jointly estimate AoA and ToF so as to improve the AoA resolution.

Indistinguishable Characters: Another challenge lies in the fact that some character pairs are hard to distinguish in their handwriting format, such as "z" and "Z", "o" and "O", "s" and "S", "v" and "V", letter "I" and number "1", etc. Sometimes, these handwritten character pairs even cannot be distinguished by a human. Unfortunately, this challenge is hard to address from a technical perspective. Therefore, AuthIoT resorts to regulation. AuthIoT asks end users to use a passcode that does not include indistinguishable pairs of characters. Excluding some characters will not compromise the passcode security as there are still sufficient characters to be used.

C. Security of AuthIoT

Essentially, AuthIoT serves as an interface for an AP to receive a passcode from an end user for authenticating a particular IoT device. It does not alter the authentication mechanism and thus has the same authentication safety as existing methods. However, due to the broadcast nature of wireless signals, AuthIoT may face the passcode leakage problem. A malicious user may overhear the signal from IoT device and attempt to infer the passcode for AP access. To address this issue, a substitution cipher [30] can be applied to the passcode at wireless AP, and the substitution rules can be updated regularly to avoid replay attacks.

IV. AoA Estimation for General Antenna Array

This section first offers a primer on the existing MUSIC algorithm for AoA estimation at a wireless device equipped with uniform linear antenna array, and then extends the MUSIC algorithm to the case where the wireless device is equipped with a general (linear or nonlinear) antenna array.

A. A Primer: MUSIC for Uniform Linear Antenna Array (MUSIC-ULAA)

System Modeling: The basic idea of AoA estimation is that different signal propagation paths are likely to have different AoAs at a receiving device. The different AoAs will introduce a corresponding phase shift across the array of antennas. For a uniform linear antenna array, once the antenna space and the phase shift are given, the AoA can be accordingly calculated. To understand AoA estimation, let us consider a receiving device with a uniform linear antenna array as shown in Fig. 2, where the number of antennas is M, and the antenna spacing is d. Assume that the number of signal propagation paths is L and let us focus on the lth path shown in the figure. Denote α_l as the complex channel attention experienced by the signal when impinging on the first antenna. Then, the complex channel attention of the signal at the second antenna

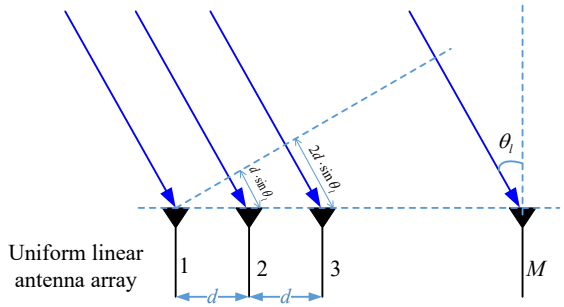


Fig. 2: Illustration of MUSIC algorithm for AoA estimates at a wireless device with uniform linear antenna array. Only one signal path with AoA θ_l is shown in the figure.

is the same except for an additional phase shift caused by the additional distance traveled by the signal. Mathematically, the additional phase shift at the mth antenna can be written as $(m-1)\cdot d\cdot \sin(\theta_l)\cdot \frac{2\pi}{\lambda}$, where λ is the wavelength of radio signal. Then, the complex channel attention at the mth antenna can be expressed as $\frac{(m-1)\cdot d\cdot 2\pi}{\lambda}\cdot \sin(\theta_l)\cdot \alpha_l$. Denote $\vec{h}_l$ as the channel coefficient vector for the lth path. Then, $\vec{h}_l = \vec{a}(\theta_l)\cdot \alpha_l$, where

$$\vec{a}(\theta_l) = \big[1 \quad e^{-j\frac{2\pi d\sin(\theta_l)}{\lambda}} \quad e^{-j\frac{4\pi d\sin(\theta_l)}{\lambda}} \quad \cdots \quad e^{-j\frac{2\pi(M-1)d\sin(\theta_l)}{\lambda}}\big]^{\mathsf{T}}. \tag{1}$$

At each antenna of the device, the observed CSI is the blend of all paths as well as noise, i.e., $\vec{H} = \sum_l \vec{h}_l = \sum_l \vec{a}(\theta_l)\alpha_l$. Then, the AoA estimation problem can be formulated as follows. Based on the N observations of CSI (i.e., $\vec{H}_n$, $n = 1, 2, \cdots N$, where $\vec{H}_n$ is the nth observation of channel vector), how to estimate θ_l, $l = 1, 2, \cdots, L$.

MUSIC Estimation: MUSIC is a subspace-based algorithm that has been widely used for AoA estimates in wireless localization. The general idea behind MUSIC method is to use all the eigenvectors that span the noise subspace to improve the performance of the Pisarenko estimator. It mainly comprises the following steps.

- Step 1: Calculate the correlation matrix of CSI observations: $\mathbf{R} = \sum_{n=1}^{N} \vec{H}_n \vec{H}_n^{\mathsf{H}}$, where $(\cdot)^{\mathsf{H}}$ is conjugate transpose operator.
- Step 2: Perform eigendecomposition of the correlation matrix: $[\mathbf{E}\ \mathbf{S}] = eig(\mathbf{R})$, where $\mathbf{E}$ is a matrix with its columns being eigenvectors and $\mathbf{S}$ is the diagonal matrix with sorted eigenvalues (in non-decreasing order).
- Step 3: Divide $\mathbf{E}$ into two sub-matrices: $\mathbf{E} = [\mathbf{E}_s\ \mathbf{E}_n]$, where $\mathbf{E}_s$ is the signal subspace and $\mathbf{E}_n$ is noise subspace.
- Step 4: Evaluate the following function for all possible θ: $p(\theta) = \frac{1}{\vec{a}(\theta)^{\mathsf{H}}\mathbf{E}_n\mathbf{E}_n^{\mathsf{H}}\vec{a}(\theta)}$, where $\vec{a}(\theta)$ is the steering direction defined in (1). The values of θ corresponding to the peaks of $p(\theta)$ are the AoAs of incoming signals.

B. MUSIC for General Antenna Array (MUSIC-GAA)

The above MUSIC algorithm assumes that the antenna array is equally spaced and linearly installed. However, in practice, most wireless APs are equipped with nonlinear antenna

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3186646

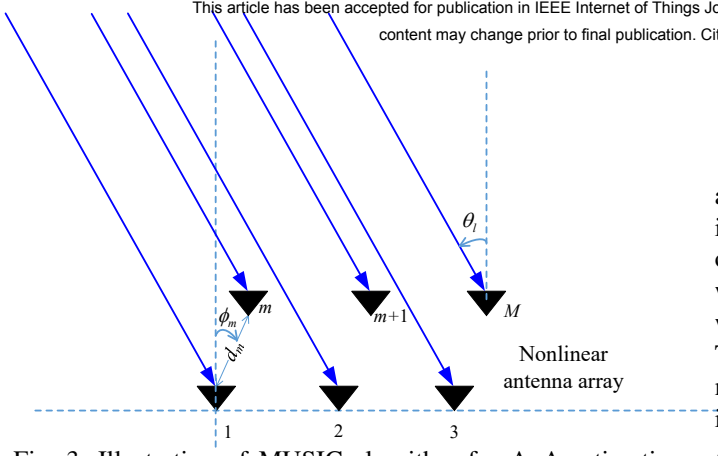


Fig. 3: Illustration of MUSIC algorithm for AoA estimation at a wireless device with nonlinear (arbitrary) antenna configuration. Only one signal path with AoA θ_l is shown here.

array. For example, many WiFi routers are equipped with four antennas which are installed in a rectangular shape to save the space. In this section, AuthIoT extends the MUSIC algorithm for a wireless device with general antenna array. In addition, it borrows the idea from SpotFi [12] to improve the AoA resolution by jointly estimating AoA and ToF of incoming signals. The rationale behind joint estimation is that, if two incoming signals are indistinguishable in the spatial domain (due to the limited number of antennas), they may be distinguishable in the time domain. Joint estimation makes it possible to distinguish two incoming signals even if they have very similar AoA.

Consider a receiving device with nonlinear antenna array as shown in Fig. 3. For notional simplicity, we adopt polar coordinate system for the antennas using the first antenna position as the origin. Denote d_m as the distance between the 1st and mth antennas and ϕ_m as their angle, as illustrated in the figure. Then, the coordinate of the mth antenna can be written as (d_m, ϕ_m). Particularly, the first antenna's coordinate is $(0, 0)$.

Recall that α_l is defined as the complex channel attention of the lth path on the first antenna. The observed channel coefficient (CSI) on the mth antenna over subcarrier k can be modeled as:

$$h_{m,k} = \sum_l \alpha_l \cdot e^{j\frac{2\pi d_m \cos(\phi_m - \theta_l)}{\lambda}} \cdot e^{-j2\pi k f_\delta \tau_l} + n_{m,k}, \quad (2)$$

where (d_m, ϕ_m) is the polar coordinate of the mth antenna, f_δ is the subcarrier spacing of OFDM modulation, $(\alpha_l, \theta_l, \tau_l)$ is the complex attention, AoA, and delay of the lth path, respectively. Lastly, $n_{m,k}$ is the CSI observation noise/error at antenna m over subcarrier k.

Collectively, the observed CSI at all antennas and over all subcarriers can be expressed as an $M \times K$ complex matrix, where M is the number of antennas and K is the number of subcarriers. Consider a four-antenna 802.11 WiFi router as an example, which has 52 valid subcarriers in OFDM modulation. The CSI matrix $\mathbf{H} \in \mathbb{C}^{4\times 52}$ can be written as follows:

$$\mathbf{H} = \overbrace{\begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} & h_{15} & h_{16} & h_{17} & h_{18} & h_{19} & \dots \\ h_{21} & h_{22} & h_{23} & h_{24} & h_{25} & h_{26} & h_{27} & h_{28} & h_{29} & \dots \\ h_{31} & h_{32} & h_{33} & h_{34} & h_{35} & h_{36} & h_{37} & h_{38} & h_{39} & \dots \\ h_{41} & h_{42} & h_{43} & h_{44} & h_{45} & h_{46} & h_{47} & h_{48} & h_{49} & \dots \end{bmatrix}}^{\text{Frequency domain over OFDM subcarriers}} \} \text{Spatial domain over antennas} \quad (3)$$

Solely using spatial degrees of freedom (DoF) provided by antennas for AoA estimate may not be an ideal approach, as it requires the number of antennas is larger than the number of paths. This requirement may not be fulfilled in a real-world indoor environment when the number of antennas on a wireless AP is limited (e.g., four antennas on a WiFi router). To improve the AoA resolution, AuthIoT expands the CSI matrix $\mathbf{H}$ for MUSIC-based AoA estimate by following the idea in [12]. Consider the CSI matrix in (3) as an example. AuthIoT can expand the CSI matrix by bonding every three columns as a new column as illustrated below:

$$\mathbf{H}_e = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} & h_{15} & h_{16} & h_{17} & \dots \\ h_{21} & h_{22} & h_{23} & h_{24} & h_{25} & h_{26} & h_{27} & \dots \\ h_{31} & h_{32} & h_{33} & h_{34} & h_{35} & h_{36} & h_{37} & \dots \\ h_{41} & h_{42} & h_{43} & h_{44} & h_{45} & h_{46} & h_{47} & \dots \\ h_{12} & h_{13} & h_{14} & h_{15} & h_{16} & h_{17} & h_{18} & \dots \\ h_{22} & h_{23} & h_{24} & h_{25} & h_{26} & h_{27} & h_{28} & \dots \\ h_{32} & h_{33} & h_{34} & h_{35} & h_{36} & h_{37} & h_{38} & \dots \\ h_{42} & h_{43} & h_{44} & h_{45} & h_{46} & h_{47} & h_{48} & \dots \\ h_{13} & h_{14} & h_{15} & h_{16} & h_{17} & h_{18} & h_{19} & \dots \\ h_{23} & h_{24} & h_{25} & h_{26} & h_{27} & h_{28} & h_{29} & \dots \\ h_{33} & h_{34} & h_{35} & h_{36} & h_{37} & h_{38} & h_{39} & \dots \\ h_{43} & h_{44} & h_{45} & h_{46} & h_{47} & h_{48} & h_{49} & \dots \end{bmatrix}. \quad (4)$$

The expanded CSI matrix is of 12 by 49 size, i.e., $\mathbf{H}_e \in \mathbb{C}^{12\times 49}$; and its correlation matrix is of 12 by 12 size, i.e., $\mathbf{H}_e\mathbf{H}_e^{\mathsf{H}} \in \mathbb{C}^{12\times 12}$. This means that, when applying MUSIC to AoA estimate, the expanded matrix renders a larger dimension for noise subspace compared to the original CSI matrix ($12-L$ versus $4-L$), thereby tending to offer a better AoA resolution.

In a general case, for CSI matrix $\mathbf{H} \in \mathbb{C}^{M\times K}$, a question is how many columns should be bonded when expanding this matrix for AoA estimate. For this question, we have the following considerations. On one hand, the number of rows of $\mathbf{H}_e$ should be maximized to improve the dimension of noise subspace; on the other hand, the expanded CSI matrix $\mathbf{H}_e$ should be a flat matrix for MUSIC calculation. Denote b as the number of bonding columns in the CSI matrix. Then, these two observations can be formulated as: $\max(Mb)$, subject to $Mg \le K - G + 1$ and $G \in \mathbb{Z}$. Hence, we have $G = \lfloor \frac{K+1}{M+1} \rfloor$. Therefore, the dimension of the expanded CSI matrix is Mg by $K - G + 1$, i.e., $\mathbf{H}_e \in \mathbb{C}^{(Mg)\times(K-G+1)}$. The jth column of $\mathbf{H}_e$ is $[H_j;\ H_{j+1};\ \cdots;\ H_{j+G-1}]$, where H_j is the jth column of $\mathbf{H}$ and $[;\cdots;]$ is vertical concatenation operator.

For the expanded CSI matrix $\mathbf{H}_e$, we would like to explore its basis for its columns. Based on (2), it can be verified that each of its columns is a linear combination of the following L basis vectors:

$$\vec{a}_l = [\underbrace{a_{11} \quad a_{21} \quad \cdots \quad a_{M1}}_{\text{column 1}} \quad \underbrace{a_{12} \quad a_{21} \quad \cdots \quad a_{M2}}_{\text{column 2}} \quad \cdots \quad \underbrace{a_{1G} \quad a_{2G} \quad \cdots \quad a_{MG}}_{\text{column G}}]^{\mathsf{T}} \quad (5)$$

for $1 \le l \le L$, where $a_{mg} = e^{j\frac{2\pi d_m \cos(\phi_m - \theta_l)}{\lambda}} \cdot e^{-j2\pi g f_\delta \tau_l}$ with $1 \le m \le M$ and $1 \le g \le G$.

Based on the expanded CSI matrix $\mathbf{H}_e$ and its column basis, the two-dimensional MUSIC algorithm is summarized as follows.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3186646

TABLE II: Simulation parameters of MUSIC-GAA.

parameter	value	parameter	value
carrier frequency	5 GHz	# of paths	5
bandwidth	40 MHz	path 1: $(\alpha_1, \theta_1, \tau_1)$	$(1.00e^{j1.26}, 15^o, 5ns)$
FFT size	64	path 2: $(\alpha_2, \theta_2, \tau_2)$	$(.40e^{j0.64}, -71^o, 21ns)$
# of valid subcarrier	52	path 3: $(\alpha_3, \theta_3, \tau_3)$	$(.20e^{-j1.86}, 81^o, 38ns)$
# of antennas	4	path 4: $(\alpha_4, \theta_4, \tau_4)$	$(.15e^{j1.64}, -15^o, 65ns)$
antenna configuration	Vertex of 6cm×6cm square	path 5: $(\alpha_5, \theta_5, \tau_5)$	$(.10e^{-j1.51}, 31^o, 89ns)$

Step 1: Measure the CSI matrix $\mathbf{H}$ at M antennas over K subcarriers. Construct the expanded CSI matrix $\mathbf{H}_e$ by letting its jth column be $[H_j;\ H_{j+1};\ \cdots;\ H_{j+G-1}]$, where H_j is the jth column of $\mathbf{H}$, $[;\cdots;]$ is vertical concatenation operator, and $G = \lfloor \frac{K+1}{M+1} \rfloor$.

Step 2: Calculate the correlation matrix of CSI observations: $\mathbf{R} = \mathbf{H}_e\mathbf{H}_e^{\mathsf{H}}$, where $(\cdot)^{\mathsf{H}}$ is conjugate transpose operator.

Step 3: Perform eigendecomposition of the correlation matrix: $[\mathbf{E}\ \mathbf{S}] = eig(\mathbf{R})$, where $\mathbf{E}$ is a matrix with its columns being eigenvectors and $\mathbf{S}$ is the diagonal matrix with sorted eigenvalues (in non-decreasing order).

Step 4: Divide $\mathbf{E}$ into two sub-matrices: $\mathbf{E} = [\mathbf{E}_s\ \mathbf{E}_n]$, where $\mathbf{E}_s$ is the signal subspace and $\mathbf{E}_n$ is noise subspace.

Step 5: Evaluate the following function for all possible θ and τ:

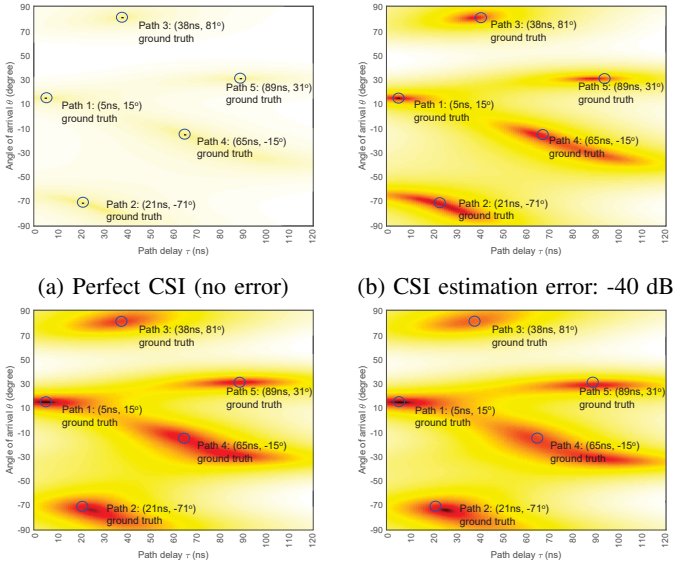
$$p(\theta, \tau) = \frac{1}{\vec{a}(\theta,\tau)^{\mathsf{H}}\mathbf{E}_n\mathbf{E}_n^{\mathsf{H}}\vec{a}(\theta,\tau)}. \quad (6)$$

Based on (5), the steering vector $\vec{a}(\theta, \tau)$ is defined as follows:

$$\vec{a}(\theta,\tau) = \big[\underbrace{a_{11}\ \ a_{21}\ \cdots\ a_{M1}}_{\text{column 1}}\ \ \underbrace{a_{12}\ \ a_{21}\ \cdots\ a_{M2}}_{\text{column 2}} \quad (7)$$
$$\cdots\ \ \underbrace{a_{1G}\ \ a_{2G}\ \cdots\ a_{MG}}_{\text{column G}}\big]^{\mathsf{T}},$$

where $a_{mg} = e^{j\frac{2\pi d_m \cos(\phi_m - \theta)}{\lambda}} \cdot e^{-j2\pi g f_\delta \tau}$ for $1 \le m \le M$ and $1 \le g \le G$. The values of (θ, τ) corresponding to the peaks of $p(\theta, \tau)$ are regarded as a path with AoA of θ and delay of τ.

An Example: We use an example to illustrate the performance of MUSIC-GAA. We consider a wireless AP and an IoT device and attempt to estimate the AoA of signal paths at the wireless AP. Table II lists the parameters that we use for simulation. Particularly, the antennas on the AP are not linear installed; instead, they are installed at the vertex of a 6cm×6cm square. This antenna configuration is more realistic compared to a uniform linear antenna array. In this case, the number of paths is greater than the number of antennas. Fig. 4 shows our simulation results when the CSI bears different levels of error. Specifically, Fig. 4a depicts the result when the AP has perfect CSI. In this figure, the small circles mark the ground truth, while the black dots in the circles are the results of MUSIC-GAA. The results reveal that MUSIC-GAA finds the exact AoAs and delays of the five paths. Figs. 4b-d depict the results when the CSI at the AP has -40 dB, -30 dB, and -20 dB error. It can be seen that the heatmap becomes increasingly blurry when the CSI bears larger error. This indicates that accurate CSI is crucial. Fortunately, AuthIoT has accurate CSI for MUSIC-GAA as the IoT device is physically close to the AP with a LoS path.



(a) Perfect CSI (no error) (b) CSI estimation error: -40 dB

(c) CSI estimation error: -30 dB (d) CSI estimation error: -20 dB

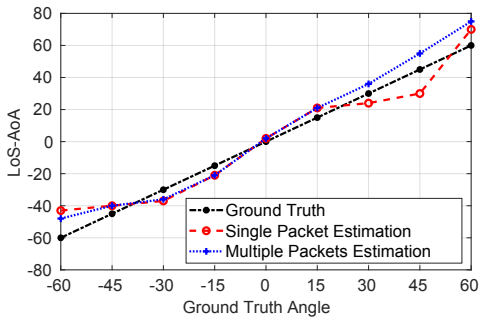
Fig. 4: Performance of MUSIC-GAA algorithm.

Another observation from Figs. 4b–d is that the hot spots appear to be horizontally stretched, rendering better accuracy for AoA estimate than for delay estimate. This is because AuthIoT only requires AoA of LoS signal path and does not need the delay information. This phenomenon stems from the CSI expansion operation (see (4) for example), where each column of the expanded CSI matrix contains the CSI from all antennas (but the CSI from a subset of subcarriers).

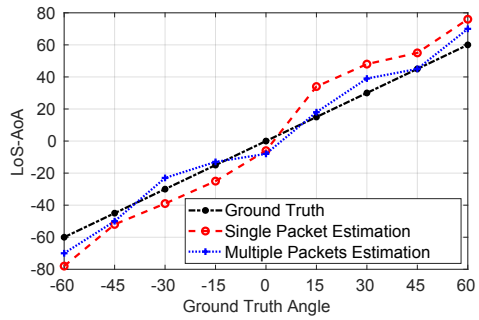
C. MUSIC-GAA for AuthIoT

Using MUSIC-GAA for AuthIoT to estimate the LoS AoA faces the following two challenges. The first challenge is the very small delay difference of multiple paths indoor environments, especially in a small room with many objects. For example, if the distance difference of two paths is 1m, their delay difference is 3.3ns. To achieve this delay resolution (3.3ns), it requires 300MHz bandwidth. Such a large signal bandwidth is not affordable for most wireless systems. 5GHz WiFi offers 40MHz bandwidth, which is insufficient to distinguish two paths whose distance difference is less than 1m. The second challenge is the CSI quantization error. For example, Atheros WiFi NIC [31] offers 10-bit CSI quantization, rendering a quantization error of $10\log_{10}(1/2^{10}) = -30$ dB; Intel 5300 WiFi NIC [32] offers 8-bit quantization for CSI, and its quantization error is $10\log_{10}(1/2^8) = -24$ dB. As shown

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3186646



(a) Linear antenna array.



(b) Nonlinear antenna array.

Fig. 5: Experimental results of MUSIC-GAA in two cases: (a) Intel 5300 card with three linear equal-spaced antennas; (b) USRP N310 with four antennas placed at the vertex of 6cm×6cm square.

in Fig. 4, the CSI error degrades the performance of MUSIC-GAA.

AuthIoT addresses these two challenges as follows. First, it asks users to keep the IoT device close to the AP (∼2m) so that there is a strong LoS path between the two devices. It also asks users to handwrite the passcode over the air at a large scale (i.e., spanning a 75cm×75cm area for each passcode character), so that the AoA change of writing a passcode character is significant. These requirements will be specified on the manual for end users. Second, it combines multiple consecutive packets to improve the LoS AoA estimation through k-means clustering [33]. Details will be given in §V-B.

We have evaluated the performance of MUSIC-GAA for AuthIoT via experiments on two cases: i) the AP is an Intel 5300 card with three linear equal-spaced antennas; and ii) the AP is a USRP N310 with four antennas placed at the vertex of 6cm×6cm square. Both testbeds use WiFi signal for data packet transmission, and the packet rate is 1000 per second. It means that the AP can obtain 1000 CSI instances per second. The distance between the AP and the IoT device is 2m, with the presence of LoS path. We conducted the measurement campaign in a two-story apartment with ordinary furniture. Fig. 5 shows our experimental results. It can be seen that the estimated LoS AoA increases/decreases as the ground-truth LoS AoA increases/decreases. This observation is consistent for both testbeds. This indicates that the LoS AoA tends to manifest a unique pattern based on the movement of IoT devices.

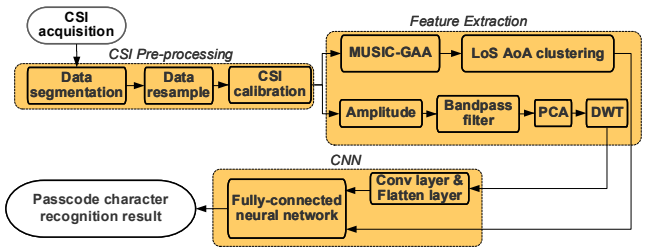


Fig. 6: Diagram of CSI-based passcode character recognition.

V. Learning-based Passcode Recognition

A passcode is composed of several characters (English alphabets, numbers, and some special characters). AuthIoT recognizes each individual character based on its generated CSI. Fig. 6 depicts the high-level system diagram of AuthIoT's passcode recognition. As shown in the diagram, AuthIoT uses both LoS AoA and normalized amplitude of CSI as the features for CNN-based character recognition. The reason is that our experiments show, compared to solely using LoS AoA as a feature, adding normalized CSI amplitude as input can considerably improve the recognition accuracy (by 5% on average in our observations). In what follows, we explain each module in Fig. 6.

A. CSI Segmentation, Resampling, and Compensation

CSI Segmentation: When a user continuously writes passcode characters in the air, the AP pings the IoT device at a certain rate (e.g., 200 ping packets per second), so that it can frequently estimate the CSI based on the ACK packets from the IoT device. In practice, an end user may take different amounts of time to write different characters, and different users may take different amounts of time to write the same character. Therefore, it is necessary to separate the collected CSI data in the time domain for each written character. To facilitate the CSI segmentation and improve its accuracy, AuthIoT asks end users to pause (holding IoT device still) one second before they begin to write a character. AuthIoT leverages the pause between two neighboring characters for CSI segmentation. In addition, AuthIoT asks end users to hold IoT device still for two seconds before they start to write passcode and after they complete passcode writing. Since a still IoT device generates unique CSI features, AuthIoT leverages such features to determine the time period of passcode writing.

Fig. 7 shows an example of AuthIoT's CSI segmentation, which comprises the following steps.

Step 1: Calculate the following metric: $g(i) = \angle(h_{m,k}(i)h_{n,k}(i)^*) \cdot |h_{m,k}(i)|$, where $h_{m,k}(i)$ is the channel coefficient from antenna m, subcarrier k, and packet i. In our design, m and n are the two antennas that offer strongest CSI, and $k = 1$. Fig. 7a shows an instance of phase difference of two channels, i.e., $\angle(h_{m,k}(i)h_{n,k}(i)^*)$. Fig. 7b shows an instance of channel amplitude, i.e., $|h_{m,k}(i)|$. Fig. 7c shows an instance of $g(i)$.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3186646

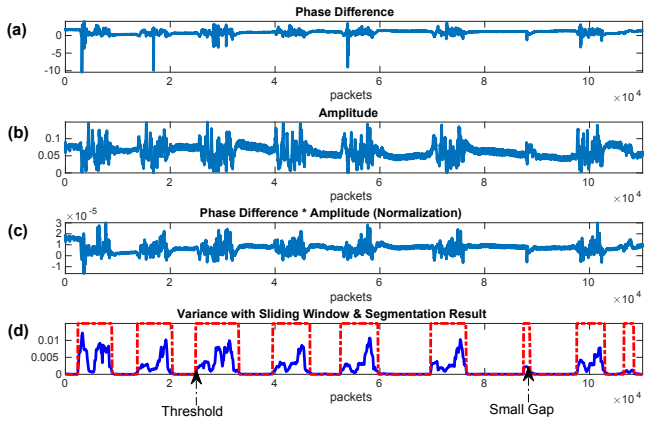


Fig. 7: An example illustrating CSI segmentation.

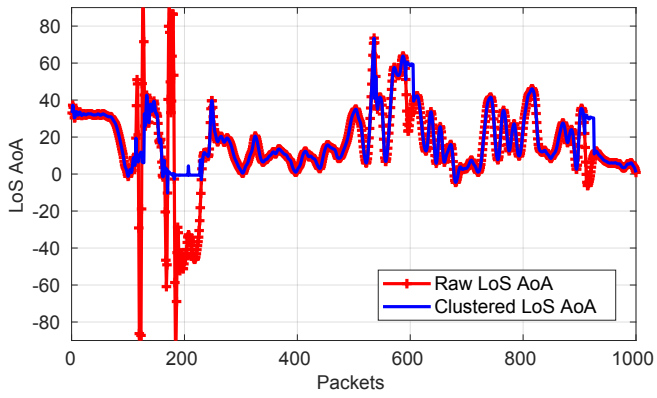


Fig. 8: Removal of abnormal LoS AoA samples through filtering.

Step 2: Calculate the window-slided variance as follows: $v(i) = \frac{1}{W}\sum_{j=i}^{i+W-1}|g(j) - \bar{g}|^2$, where $\bar{g} = \frac{1}{W}\sum_{j=i}^{i+W-1} g(j)$. Fig. 7d shows an instance of $v(i)$.

Step 3: Compare $v(i)$ with a threshold T_v, where $T_v = 0.03 \times avg\{v(i)\}$. The CSI segment corresponding to $v(i) \geq T_v$ is considered for an individual character. Fig. 7d illustrates the windows corresponding to the segments of CSI to be used for character recognition.

Step 4: Check the segmentation length for each letter. If the time duration of a CSI segment is shorter than 1 second or longer than 4 seconds. AuthIoT discards this CSI segment.

CSI Resampling: After CSI segmentation, different CSI segments may have different numbers of CSI samples. The purpose of resampling is to make sure that the number of CSI samples in each CSI segment is identical. Doing so is likely to ease the training and inference of CNN. AuthIoT resamples each CSI segment using linear interpretation and/or decimation on the real and imaginary parts of CSI samples.

CSI Compensation The CSI data need to be calibrated before feeding to the MUSIC-GAA. Since the receiver and transmitter are not synchronized, the CSI data from a WiFi receiver may suffer from Sampling Time Offset (STO) and Sampling Frequency Offset (SFO). To compensate STO and SFO, a popular method is performing linear regression over multiple consecutive CSI instances in both time and frequency domains [12]. The linear fit of the unwrapped CSI phase for i_{th} packet can be expressed as

$$\tau_{s,i} = arg\min_{\alpha} \sum_{m=1}^{M}\sum_{n=1}^{N}(\phi_i(m,n) + 2\pi f_\delta(n-1)\alpha + \beta) \quad (8)$$

The $\tau_{s,i}$ is the STO for i_{th} packet. The f_δ is the frequency spacing between subcarriers. And the $\phi_i(m,n)$ is the wrapped phase at m_{th} antenna and n_{th} subcarrier. After estimating $\tau_{s,i}$ based on (8), the compensation is performed by adding $2\pi f_\delta(n-1)\tau_{s,i}$ to subcarrier n, $n = 1, 2, \cdots, N$. The same compensation applies to the CSI from each antenna.

B. Feature Extraction

1) LoS AoA Feature Extraction: AuthIoT uses MUSIC-GAA to estimate the AoA-delay profile of the signal paths based on the CSI samples. One observation from our experiments is that the AoA corresponding to the largest profile value is always associated with the minimum delay. This makes sense as there always exists a strong LoS path between the IoT device and the AP. Based on this observation, AuthIoT chooses the AoA corresponding to the largest value as LoS AoA.

As shown in Fig. 8, the LoS AoA computed from CSI is noisy due to the imperfection of hardware (e.g., 8-bit quantization) and interference caused by environment changes (e.g., body movement). Sometimes the LoS AoA jumps over 20 degrees for consecutive 10 packets. Obviously, such a big jump is abnormal. To reduce the adverse effect of this phenomenon, AuthIoT employs a clustering algorithm for the elimination of unexpected AoA values. The rationale behind this algorithm is that the AoA should not change over 20 degrees over 10 packets (10 ms). The clustering algorithm works as follows.

Step 1: Slide a window of size 10 to move across the LoS AoA sample sequence using the step size of 5. In each window, the k-means clustering algorithm [33] is employed to divide the 10 LoS AoA samples into 2 groups.

Step 2: Calculate the average values of the samples in the two groups. If the difference is larger than 20 degree and the number of samples in one group is less than 3, then the group of smaller size is regarded as abnormal.

Step 3: Replace every sample in the abnormal group with the average value of the larger group.

2) Amplitude Feature Extraction: In addition to LoS AoA, AuthIoT uses CSI amplitude as another feature for CNN-based recognition. The raw CSI amplitude is noisy. To enhance the input data quality, AuthIoT employs a Butterworth band-pass filter with frequency band 5Hz–20Hz to eliminate the undesired frequency components and reduce the noise for the CSI amplitude. This is because human's writing movement is in this frequency range [8]. Fig. 9 shows an example of the filtering operation.

In indoor environments, wireless channels over neighboring subcarriers are very similar [7]. Hence, AuthIoT applies Principle Component Analysis (PCA) to a group of adjacent

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3186646

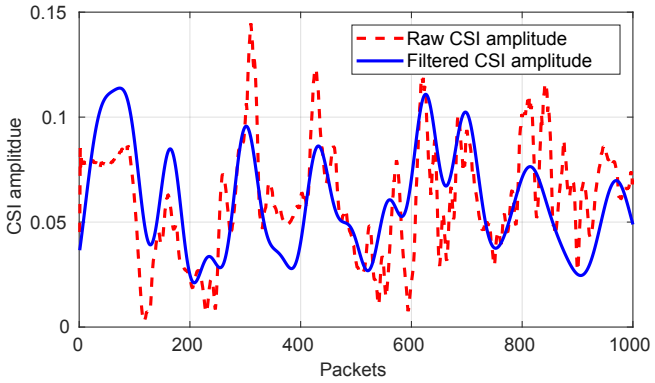


Fig. 9: CSI amplitude before and after bandpass filtering.

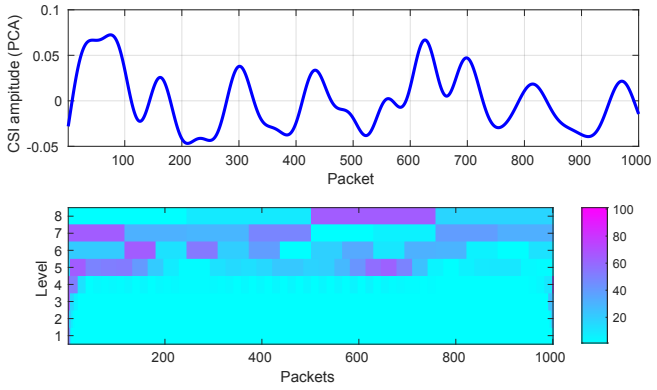


Fig. 11: Illustration of DWT operation on CSI amplitude.

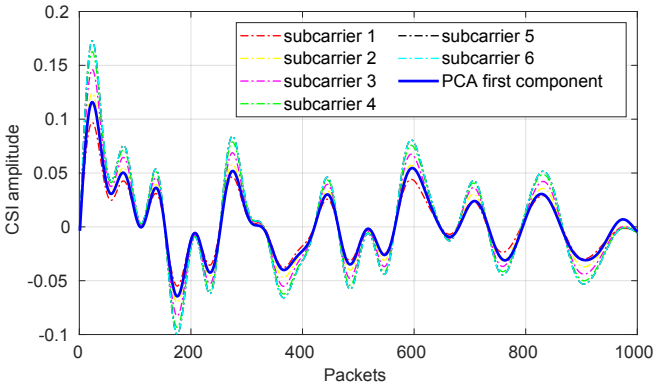


Fig. 10: Illustration of PCA operation on CSI amplitude.

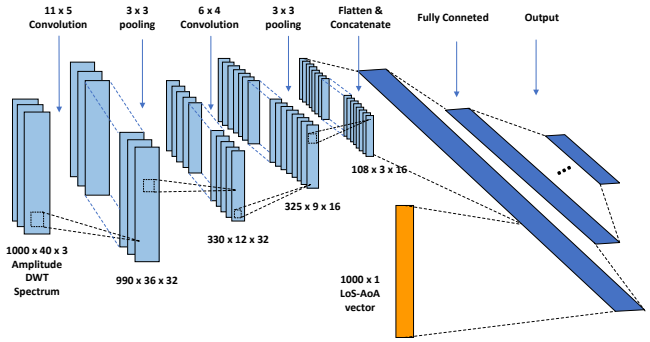


Fig. 12: CNN Structure.

subcarriers for data compression. Specifically, AuthIoT groups every 6 subcarriers and applies PCA to each group. The first component of PCA results is kept as the amplitude features, while other components are discarded. Fig. 10 shows an example of this operation. As it can be seen, the adjacent 6 subcarriers have similar channel amplitude, and the first component of PCA results maintains the main shape of the channels.

Writing a character over the air mainly comprises a series of strokes. The action of each stroke is the key feature for the CSI-based character recognition. To capture the action of each stroke, AuthIoT performs Discrete Wavelet Transform (DWT) on the CSI amplitude after PCA operation, as shown in Fig. 6. Similar to WiReader [8], it performs 8-level discrete wavelet transform on the CSI amplitude samples using `symlet` as the basis function. Fig. 11 shows an example of DWT operation on the CSI amplitude, where Fig. 11(a) shows the CSI amplitude from PCA and Fig. 11(b) shows the DWT results. The DWT results are then sent to the CNN for training and inference.

C. CNN Settings and Training

1) CNN Settings: Fig. 12 shows the structure of CNN, which is composed of convolution layers, flatten layers and fully-connected layers. Since the CSI amplitude matrix is of high dimension $(1000 \times 40 \times 3)$, AuthIoT employs convolution operations to extract its high-level features and reduce its dimension. Specifically, AuthIoT treats the amplitude DWT spectrum (1000×40) as an image and each of the three antennas as an image channel, similar to the process of RGB channels in colorful image recognition. Two convolution layers are used to compress the amplitude DWT spectrum. The first convolution layer involves 32 kernels of 11×5 size, and the second layer has 16 kernels of 6×4 size. The step size of both kernels is one. The purpose of the convolution layers is to extract the features from amplitude DWT spectrum based on its spatial relationship. It employs kernels moving across the feature matrix and outputs the convolution result with ReLU function. To further reduce the data dimension, AuthIoT employs an averaging pooling layers with a size of 3×3 for each of the convolution layers. The pooling layers down-sample the amplitude matrix, thereby reducing the computational complexity. The output of the second pooling layer is flattened for vectorization. AuthIoT then concatenates the resultant amplitude features with the AoA features, and feeds the concatenated data vector to a fully-connected $128 \times 64 \times 32$ neural network. SoftMax activation function is used for the output layer to calculate the probability of each possible passcode character.

2) CNN Training and Inference: As stated before, some character pairs are not distinguishable in their handwriting format, such as "z" and "Z", "c" and "C", "o" and "O", "s" and "S", "v" and "V", letter "I" and number "1", etc. Unfortunately, this challenge is hard to address from a technical perspective. Therefore, AuthIoT excludes the subset of indistinguishable characters. Table III lists the 48 characters that can be used for passcode in AuthIoT.

TABLE III: Passwords Characters

Capital Letters	A-Z
Lower-case letters	a,b,d,e,f,g,h,q,r,t
Numbers	3-9
Special Characters	#,$,%,+,=

To train the CNN model, CSI data are collected from different locations and diverse users (details given in §VI). The batch size in our training process is set to 100, and the number of epochs is set to 25. A batch normalization layer is added to the neural network after the activation function. We observed that it could improve the convergence speed in the training process, especially when the CSI is not stable due to the change of environment. In addition, a dropout layer is added after the second (64 neurons) and third (32 neurons) layers to avoid overfitting [34]. It can make the network less sensitive to specific neurons, and in turn make the network better generation. The dropout rate is set to 0.2 for each layer by randomly setting the output to zero. The CNN uses cross-entropy as the loss function and employs Adam optimization algorithm to update the weights.

After the CNN is trained, the system is then used for online passcode character recognition in different environments. The CNN model will eventually yield the possibility of the input being each character. The character with the highest probability is regarded as the character being written by the end user.

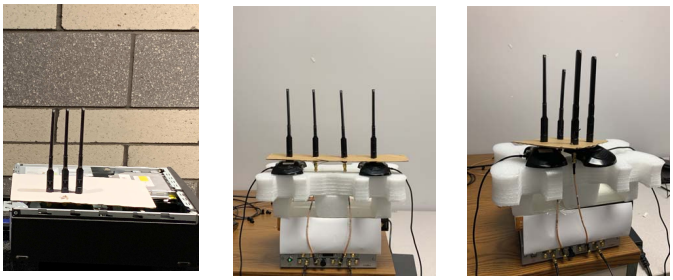
VI. Experimental Evaluation

A. Implementation and Experimental Settings

Intel 5300 Testbed: This testbed is implemented using Dell XPS 8940 Desktop with the Intel WiFi NIC 5300 and a Redmi Note 9 Pro cellphone. The desktop serves as AP working in hotspot mode, and the cellphone emulates an IoT device. The desktop is installed with the Ubuntu 14.04 operating system with 802.11 Linux CSI tool [32], which is used to acquire the CSI from the WiFi card. The carrier frequency is 5GHz, and the bandwidth is 40MHz. The packet rate is 600 packets per second. Intel WiFi NIC 5300 is equipped with three antennas, which are linearly placed with equal spacing. The antenna spacing is half wavelength (3cm). Fig. 13(a) shows the linear antenna setting of this testbed.

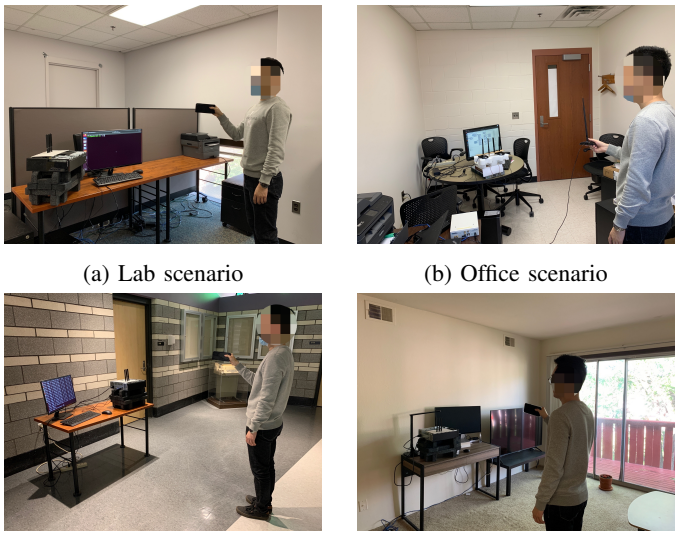
USRP Testbed: This testbed consists of a USRP N310 and a USRP N210. USRP N310 has four antennas. It serves as the AP. USRP N210 has one antenna. It emulates an IoT device by sending data packets to USRP N310. The carrier frequency is 2.4GHz, and the bandwidth is 20MHz. The packet rate is 1000 per second. This testbed has two antenna settings: linear antenna array as shown in Fig. 13b and nonlinear antenna array as shown in Fig. 13c. For the linear case, the antenna spacing is 6.25cm. For the nonlinear case, the four antennas are positioned at the vertex of a 6cm×6cm square.

Experimental Settings: Four scenarios are considered for the evaluation of AuthIoT: lab, office, hallway, and home, as shown in Fig. 14. The AP was placed on a table of 70cm height, and the IoT device was held by the participants. The participants were asked to face the AP and keep an approximate 2m distance. We placed the two testbeds in these four scenarios and collected data to evaluate the performance.



(a) (b) (c)

Fig. 13: AP antenna settings: (a) Intel 5300 testbed with linear antenna array; (b) USRP testbed with linear antenna array; (c) USRP testbed with nonlinear antenna array.



(a) Lab scenario (b) Office scenario

(c) Hallway scenario (d) Home scenario

Fig. 14: Experimental settings.

The training data were collected solely from lab, while the evaluation (inference) was performed in four scenarios (lab, office, hallway, and home). The training data were collected from five different participants, while the evaluation was conducted over nine participants (i.e., those five participants for training plus four new participants). In the training phase, each participant was asked to write the 48 characters in Table III, and each character was repeated 12 times. In total, 576 data samples were collected from each participant in the lab scenario for the training purpose.

In the test (inference) phase, each of the nine participants was asked to hold the IoT device and write 500 characters at his/her will at each scenario. The collected data samples were fed into the system for evaluation purpose.

B. Experimental Results from Intel 5300 Testbed

Intel 5300 is a commercial off-the-shelf WiFi NIC that is widely used for computers and routers. Evaluating AuthIoT on this testbed reveals its performance in real-world WiFi networks.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3186646

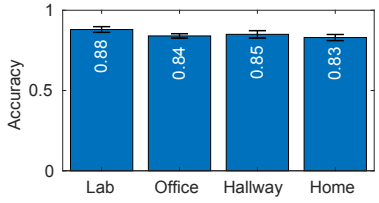


Fig. 15: Recognition accuracy of AuthIoT on Intel 5300 testbed.

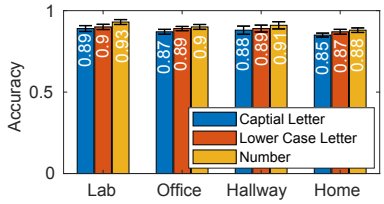


Fig. 16: Recognition accuracy breakdown of AuthIoT on Intel 5300 testbed.

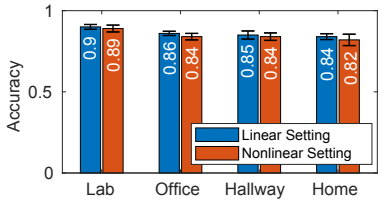


Fig. 17: Recognition accuracy of AuthIoT on USRP testbed.

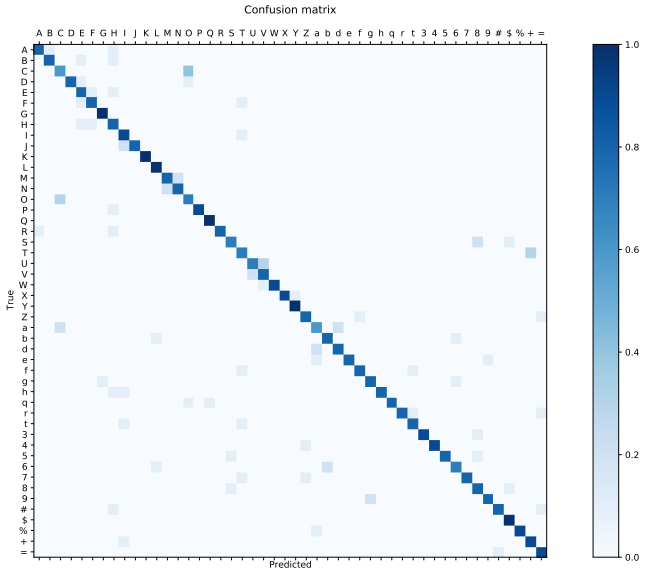


Fig. 18: Confusion matrix for Intel 5300 testbed (with linear antenna array)

Overall Accuracy: Fig. 15 presents the overall recognition accuracy on this testbed. Literally, AuthIoT reaches 88% recognition accuracy with a standard deviation of 0.018 over the nine participants in the lab scenario; it reaches 85% recognition accuracy with a standard deviation of 0.023 in the hallway scenario; it reaches 84% recognition accuracy with a standard deviation of 0.014 in the office scenario; and it reaches 83% recognition accuracy with a standard deviation of 0.019 in the home scenario. It can be seen that AuthIoT performs best in the lab scenario. This is not surprising, because AuthIoT's CNN model was trained by the dataset collected from the lab scenario.

Fig. 18 shows the confusion matrix of passcode character recognition. It can be seen that the accuracy is above 85% for most characters. The majority of errors occur due to the ambiguity of the characters sharing similar hand gestures. For example, AuthIoT is more likely to be confused by letters 'C' and 'O'; it is also hard to distinguish letters 'M' and 'N'.

Accuracy of Individual Category: To obtain more details, we examine the performance of AuthIoT over three subsets of passcode characters: 26 upper-case letters, 10 lower-case letters, and 10 numbers. Fig. 16 shows our test results. It can be seen that the recognition accuracy in all scenarios are beyond 85% for the three subsets of characters.

C. Experimental Results from USRP Testbed

We further evaluate the performance of AuthIoT on the USRP testbed with linear and nonlinear antenna arrays.

Linear Antenna Array: Fig. 17 presents the recognition accuracy on the USRP testbed when it is equipped with four linearly equal-spaced antennas. It can be seen that the recognition accuracy in the lab scenario is better than other scenarios. This is because AuthIoT's CNN model was trained by the dataset collected from the lab scenario. It also can be seen that the recognition accuracy on the USRP testbed is slightly higher than that on Intel 5300 testbed. This can be attributed to the fact that the USRP testbed has one more antenna than the Intel 5300 testbed.

We examine the recognition accuracy for each individual participants. Fig. 19 shows our experimental results. The results show that the recognition accuracy is within the range of 81% to 88% for the nine participants. This indicates that AuthIoT is robust against the variation of end users.

Nonlinear Antenna Array: Fig. 17 also presents the recognition accuracy when the USRP testbed is equipped with four nonlinear antennas. It can been seen that the two cases (linear antenna array and nonlinear antenna array) have very similar recognition accuracy, with a difference less than 2%. The performance similarity can be traced down to the accuracy of LoS AoA estimation as shown in Fig. 5. Since the LoS AoA estimation in the two antenna settings has similar accuracy, it is not surprising that the recognition in the two antenna settings has similar accuracy.

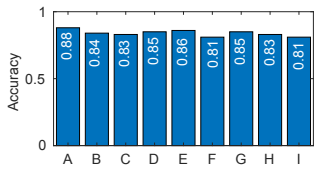


Fig. 19: Recognition accuracy of each individual participant on USRP testbed.

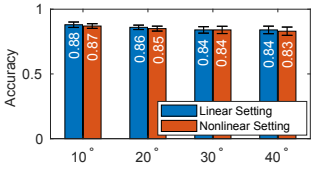


Fig. 20: Recognition accuracy for user at different angles to the AP.

D. Robustness of AuthIoT

To evaluate the robustness of AuthIoT, we examine its performance when the user is located at different distances and from different directions.

Different Distances: We change the distance between AP and IoT device to examine the performance of AuthIoT. We consider four distances: 2.0m, 2.5m, 3.0m, and 3.5m. We

TABLE IV: Recognition accuracy of AuthIoT when the distance between AP and IoT device changes.

	Lab		Office		Hallway		Home	
distance	Linear	Non-linear	Linear	Non-linear	Linear	Non-linear	Linear	Non-linear
2.0m	89%	89%	86%	84%	85%	84%	84%	82%
2.5m	87%	86%	85%	84%	84%	84%	83%	82%
3.0m	85%	85%	84%	83%	84%	83%	82%	81%
3.5m	85%	84%	83%	82%	83%	83%	82%	81%

conduct experiments in four scenarios: lab, office, hallway, and home. Table IV presents our experimental results. It can be seen that, in each scenario, AuthIoT has a consistent performance when the distance between AP and IoT device varies from 2.0m to 3.5m. For all cases, the recognition accuracy of AuthIoT is within the range from 81% to 89%, regardless of the experimental scenario, the antenna pattern, and the distance between AP and IoT device. This indicates the robustness of AuthIoT.

Different Directions: To evaluate its robustness to user's facing direction, we let the user keeps the same distance to the AP but moves around with different facing angles ranging from 10 degree to 40 degree. As we can observe from Fig. 20, the recognition accuracy of AuthIoT slightly degrades when the angle between the user and AP increases from 0 to 40 degree. This is because the training data are collected at the 0 degree location. However, it can be observed that the accuracy for both linear and nonlinear settings are always above 83%.

Discussions: The overall recognition accuracy of 83% is not perfect but within an acceptable range. In practice, there are some ways to further improve AuthIoT's Quality of Experience (QoE) for end users. For example, an end user can consider using an all-numbers passcode. AuthIoT offers a superior performance when the passcode is all numbers. Meanwhile, an all-number passcode is sufficiently strong in practice. Moreover, a prompt-notification mechanism can be added into AuthIoT to improve the QoE of end users. In essence, AuthIoT is a learning-based classification algorithm. The output of AuthIoT includes not only the corresponding character but also its recognition probability (i.e., the recognition confidence). When AuthIoT has a low confidence for a character recognition, it immediately asks end user to rewrite the previous character. Doing so will offer a better QoE for end users.

VII. CONCLUSION

In this paper, we studied the communication authentication problem for wireless IoT devices without an input interface. We presented AuthIoT to authenticate such IoT devices in WiFi networks by leveraging the unique CSI pattern generated by the movement of IoT devices. AuthIoT exploits environment-independent CSI features for learning-based character recognition, and therefore is transferable for cross-environment applications. AuthIoT also extends its applications for the case where a WiFi AP is equipped with a nonlinear-installed antenna array by generalizing existing AoA estimation methods. We have built a prototype of AuthIoT and evaluated its performance on the testbeds with linear and nonlinear antenna arrays. Our experimental results confirm that AuthIoT is transferable for cross-environment applications, and show that AuthIoT achieves at least 83% recognition accuracy.

REFERENCES

[1] Statista, "Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030." https://www.statista.com/topics/2637/internet-of-things/, Accessed:13-June-2021.

[2] "Gosund WiFi smart switch." https://www.gosund.com, Accessed:11-June-2021.

[3] "Sylvania WiFi dimmable led light bulb." https://consumer.sylvania.com, Accessed:11-June-2021.

[4] "Agshome solutions." https://www.agshomesolutions.com, Accessed:11-June-2021.

[5] "Google home assistant." https://assistant.google.com, Accessed:11-June-2021.

[6] "Amazon Alex." https://developer.amazon.com/en-US/alexa, Accessed:11-June-2021.

[7] Z. Fu, J. Xu, Z. Zhu, A. X. Liu, and X. Sun, "Writing in the air with WiFi signals for virtual reality devices," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. 473–484, 2019.

[8] Z. Guo, F. Xiao, B. Sheng, H. Fei, and S. Yu, "WiReader: Adaptive air handwriting recognition based on commercial WiFi signal," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10483–10494, 2020.

[9] L. Zhang, J. Wang, Q. Gao, X. Li, M. Pan, and Y. Fang, "Letfi: Letter recognition in the air using CSI," in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2018.

[10] L. Sun, S. Sen, D. Koutsonikolas, and K.-H. Kim, "WiDraw: Enabling hands-free drawing in the air on commodity WiFi devices," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pp. 77–89, 2015.

[11] X. Cao, B. Chen, and Y. Zhao, "Wi-Wri: Fine-grained writing recognition using Wi-Fi signals," in *2016 IEEE Trustcom/BigDataSE/ISPA*, pp. 1366–1373, 2016.

[12] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using wifi," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pp. 269–282, 2015.

[13] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single WiFi access point," in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, (Santa Clara, CA), pp. 165–178, USENIX Association, Mar. 2016.

[14] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, (Lombard, IL), pp. 71–84, USENIX Association, Apr. 2013.

[15] K. J. Jie Xiong, Karthikeyan Sundaresan, "Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, MobiCom '15, (New York, NY, USA), p. 537–549, Association for Computing Machinery, 2015.

[16] T.-C. Tai, K. C.-J. Lin, and Y.-C. Tseng, "Toward reliable localization by unequal AoA tracking," in *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '19, (New York, NY, USA), p. 444–456, Association for Computing Machinery, 2019.

[17] Y. Xie, J. Xiong, M. Li, and K. Jamieson, "Md-track: Leveraging multi-dimensionality for passive indoor Wi-Fi tracking," in *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom '19, (New York, NY, USA), Association for Computing Machinery, 2019.

[18] Z. Yan, Q. Song, R. Tan, Y. Li, and A. W. K. Kong, "Towards touch-to-access device authentication using induced body electric potentials," in *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom '19, (New York, NY, USA), Association for Computing Machinery, 2019.

[19] X. Li, F. Yan, F. Zuo, Q. Zeng, and L. Luo, "Touch well before use: Intuitive and secure authentication for iot devices," in *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom '19, (New York, NY, USA), Association for Computing Machinery, 2019.

[20] N. Ghose, L. Lazos, and M. Li, "SFIRE: Secret-free-in-band trust establishment for COTS wireless devices," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pp. 1529–1537, 2018.

[21] J. Zhang, Z. Wang, Z. Yang, and Q. Zhang, "Proximity based IoT device authentication," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, 2017.

[22] Y. Zheng, Y. Zhang, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Zero-effort cross-domain gesture recognition with Wi-Fi," in *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 313–325, 2019.

[23] C. Li, M. Liu, and Z. Cao, "WiHF: Enable user identified gesture recognition with wifi," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 586–595, IEEE, 2020.

[24] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "Signfi: Sign language recognition using WiFi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, pp. 1–21, 2018.

[25] R. H. Venkatnarayan, G. Page, and M. Shahzad, "Multi-user gesture recognition using WiFi," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 401–413, 2018.

[26] C. Wu, F. Zhang, Y. Fan, and K. R. Liu, "RF-based inertial measurement," in *Proceedings of the ACM Special Interest Group on Data Communication*, pp. 117–129, 2019.

[27] E. Soltanaghaei, A. Kalyanaraman, and K. Whitehouse, "Multipath triangulation: Decimeter-level WiFi localization and orientation with a single unaided receiver," in *Proceedings of the 16th annual international conference on mobile systems, applications, and services*, pp. 376–388, 2018.

[28] "Decora smart - smart switches." https://www.leviton.com/, Accessed:11-June-2021.

[29] "Simplisafe." https://simplisafe.com, Accessed:11-June-2021.

[30] Wikipedia, "Substitution cipher." https://en.wikipedia.org/wiki/Substitution_cipher, Accessed:13-June-2021.

[31] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity WiFi," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, MobiCom '15, (New York, NY, USA), p. 53–64, ACM, 2015.

[32] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *SIGCOMM Comput. Commun. Rev.*, vol. 41, p. 53, Jan. 2011.

[33] S. Lloyd, "Least squares quantization in PCM," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.