Learning-Based Radio Fingerprinting for RFID Secure Authentication Scheme

Jiaqi Xu*, Xingya Zhao*, Arjun Bakshi, Kannan Srinivasan
The Ohio State University
{xu.1629, zhao.2053, bakshi.11}@osu.edu, kannan@cse.ohio-state.edu

***Abstract*—In recent years, many low-cost identification systems have been proposed using RFID tags. Due to their limited computational capability, passive RFID tags cannot provide secure links with cryptographic algorithms, and are vulnerable to cloning and replaying attacks. RF fingerprinting techniques based on distinctive hardware imperfections of devices have been explored for RFID systems. However, since RFID tags send their payloads by reflecting the carrier wave signal from readers, the received tag response signal will contain reader-specific fingerprints which can hinder the authentication accuracy using RF fingerprints. The rich dynamics of the wireless channel also present significant challenges for RF fingerprinting. To address these issues, we present FILES, a learning-based RF fingerprinting scheme for RFID systems. It is built upon the top of the EPC C1G2 protocol and works with any COTS tags. FILES analyzes the channel and reader information from the received carrier wave signal and decouples it from the tag response signal. It then extracts features from the preprocessed signal and classifies them with a neural network. We build a prototype of our system and conduct extensive experiments. The results show that our system can achieve an accuracy of up to 99% across readers, across tag locations, and across environments with 50 COTS RFID tags.**

***Index Terms*—Radio frequency identification (RFID), Radio frequency fingerprinting**

I. Introduction

In recent years, radio frequency identification (RFID) has gained increasing attention and has been extensively used to build low-cost monitoring and identification systems in supply chains, identity documents, and sensing systems [1]–[4]. Typically, an RFID system contains at least one tag reader and multiple passive RFID tags. RFID readers transmit a carrier wave (CW) modulated by query commands, and tags would reflect the CW signal and encode their payloads in the reflected signal. This paradigm reduces the power consumption of commodity off-the-shelf (COTS) tags and works well for low-cost identification purposes. However, the limited computational capability of tags also makes them unable to support security protocols based on cryptographic algorithms, and thus vulnerable to cloning and replaying attacks.

Radio frequency (RF) fingerprinting [5]–[8] has provided a new opportunity for low-cost and efficient secure schemes using RFID tags. The intuition is to extract features from RF signals caused by hardware imperfections in devices, such as frequency and sampling offset, harmonic distortions, and phase noise. These distortions are often collectively referred to as the *fingerprint* of a device. RF fingerprinting has been applied to secure RFID systems. In [9], the authors utilize modulation-shape and spectral features of backscatter signals and identify high frequency RFID tags with traces collected in a controlled environment. In [10], the authors exploit the time interval error and the average baseband power as fingerprints for ultra-high frequency (UHF) RFID tags and demonstrate their effectiveness with traces of high sampling rate collected with different reader-tag distances and orientations. In [11], the authors extract the covariance-based distribution and power spectrum density as fingerprints for UHF RFID tag classification and evaluate the proposed system under different deployments and environments. A more recent work [12] introduces the convolutional neural network (CNN) to the tag fingerprinting problem. With data augmentation and federated learning, the trained CNN model can classify UHF RFID tag traces from different deployments with high accuracy.

Though features and methods proposed in the above-mentioned works are all well-evaluated, we notice that only one reader is used to collect traces in each work, and no attention has been paid to how RFID readers would affect the fingerprints of tags. Due to the backscattering nature of RFID tags, the received signals are affected by channel conditions, the hardware imperfections of tags, and the hardware imperfections of readers. The fingerprinting methods in [10]–[12] are proved effective across different channel conditions, but their result fingerprints can still be a mixture of features of tags and readers. The accuracy might decrease if fingerprints are used to authenticate traces collected by other readers. Instead of taking the decrease in performance or building a separate fingerprint dataset for every reader, we would prefer the tag fingerprints to be more independent of the readers.

Our goal is to design a low-cost authentication scheme for RFID-based systems. The new scheme should perform well with different readers and under different channel conditions, require no hardware modification, and produce results quickly enough for real-time processing. To achieve this goal, we present FILES (FIngerprint LEarning System), a learning-based RF fingerprinting authentication scheme for RFID systems. The intuition of FILES is that the received CW signal contains information about the reader and channel conditions. FILES analyzes the received CW signal and cancels it out from the tag response signal. After this cancellation, the residual

*These authors contributed equally to this work.

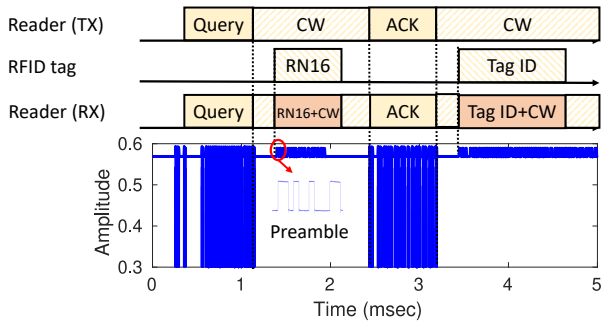

Fig. 1. RFID reader-tag communication

signal should only contain the reflected signal with tag features in it. The preprocessed signal is sent to a feature selection algorithm to reduce its size and then a lightweight neural network for classification. FILES is built upon the EPC C1G2 protocol [13] and is compatible with commercial RFID tags. Evaluation results show that FILES can achieve an accuracy of up to 97% across readers, up to 99% across tag locations, and up to 96% across environments.

The main contributions of our work are as follows.

- To the best of our knowledge, FILES is the first system that addresses the effect of readers in RF fingerprinting for RFID tags. We study how readers affect the received signal and propose a method to remove their effects.
- We propose an effective and efficient learning-based RF fingerprinting method for RFID systems. It is compatible with COTS RFID tags and can be used as an additional secure layer in any RFID system.
- The evaluation results prove that FILES is efficient and accurate in various experiment settings. It can also classify traces collected in unseen environments or using unseen readers with high accuracy.

II. Background and Challenges

In this section, we introduce the RFID communication protocol and the challenges in RF fingerprinting for RFID tags.

A. A primer on RFID Communication

UHF RFID systems operate between 902 and 928 MHz following the EPCglobal Class 1 Gen 2 (EPC C1G2) protocol [13]. Fig. 1 illustrates the RFID communication process in one time slot. The reader initializes the time slot by sending a query command and then starts transmitting the CW signal to power the tags. After receiving the query command, the tags would use the CW signal to backscatter and transmit an RN16 packet in randomly chosen time slots. Tags transmit data by reflecting or absorbing the CW signal. Each RN16 packet includes a preamble and a 16-bit random number sequence. After decoding the first collision-free RN16 packet, the reader responds with an ACK to the selected tag. This tag then transmits the EPC packet that contains its identity information.

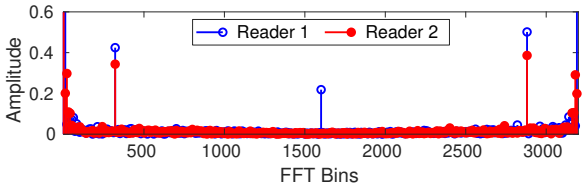

Fig. 2. The spectrum of clean received CW signal

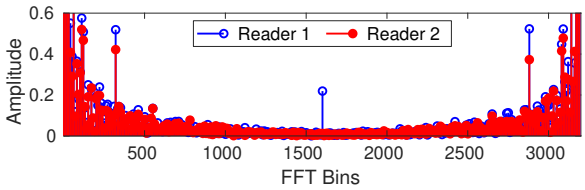

Fig. 3. The spectrum of tag response signal. The differences are dominated by the differences of received CW signal.

B. Challenges in Fingerprinting RFID Tags

Both channel conditions and readers can affect the RF fingerprints of RFID tags. In [10]–[12], the authors have discussed the effect of channel conditions on the performances of fingerprinting systems. Therefore, we will focus on the effect of hardware imperfections of readers.

We record the traces with two readers, extract 3200 samples as examples from the received CW signal and the tag response signal, and plot their spectrums as in fig. 2 and 3. When collecting these traces, we fix the locations of the tag, antennas, and surrounding objects in the environment to minimize changes in channel conditions.

The received CW signal is affected by the environment channel and hardware imperfections of the reader, but it does not contain any information about tags since they are not triggered at this time. The tag response signal is the superposition of the CW signal and the signal reflected by the tags. From fig. 2, we can see that the non-DC components of received CW signals from two readers are significantly different, especially at certain frequencies. Considering that the two example traces are collected in a controlled environment, the difference is mostly introduced by readers' hardware imperfections. A similar spectrum difference pattern is also observed in fig. 3, which means the distortions introduced by readers cannot be neglected in tag response signals. Such distortions caused by readers can affect RF fingerprinting methods that take spectral characteristics [9]–[11] or raw signals [12] as input. Their performances could downgrade if the fingerprints are extracted with traces from one reader and they are used to authenticate traces from other readers.

III. Design

The core idea of FILES is to decouple the received CW signals and tag-reader channels from tag response signals, then use a supervised learning approach to classify the tags. FILES is built on the EPC C1G2 protocol. It does not require any hardware modification to tags and can be added as a secure layer to any RFID system.

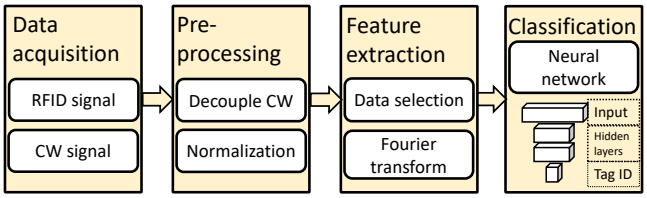

Fig. 4. Architecture of FILES.

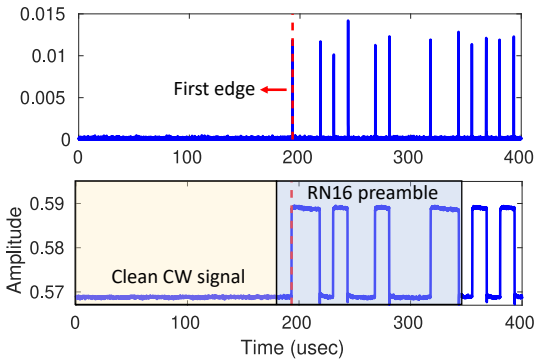

Fig. 5. An example of data acquisition. The upper figure plots the absolute derivative of each sample. The below figure shows the signal amplitude.

Fig. 4 illustrates FILES's system architecture. After acquiring signals from the receiver, our first step is to extract the RN16 preambles. Next, we extract the signal reflected by tags by decoupling the received CW signal through optimization and normalizing the signal to cancel the effect of dynamic wireless channels in the pre-processing step. After that, we extract features by applying Fourier transformation to selected portions of the signal. The last step is to use a fully-connected neural network to classify the tags for authentication purposes. We will present the details of each step in this section.

A. Data Acquisition

As shown in fig. 1, signals from readers contain multiple packets in every tag-reader communication. To avoid data-dependent bias, we use signals of the RN16 preamble for fingerprinting, which is fixed for every tag in every communication. To extract RN16 preambles, we first locate the RN16 packets with knowledge of the orders and lengths of packets as defined in the protocol, as well as the different amplitude variations in reader packets and tag packets. After detecting the RN16, we need to find the fine-grained starting point of the first edge to align the signals. As demonstrated in fig. 5, we calculate the absolute amplitude difference of adjacent samples in the previously-located RN16 packet, and choose the first point greater than a certain threshold as the first edge. We include a fixed number of samples before the first edge to make sure the tag features in the first edge are preserved. We also obtain the clean received CW signals by retrieving a fixed number of samples before the RN16 preambles. The clean received CW signals will be used to initialize the optimization problem in pre-processing.

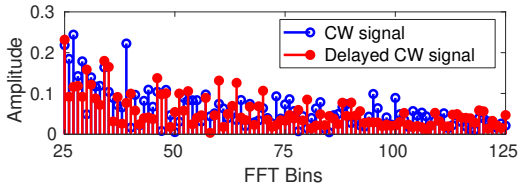

Fig. 6. Spectrum of the two adjacent sample sequences in the CW signal

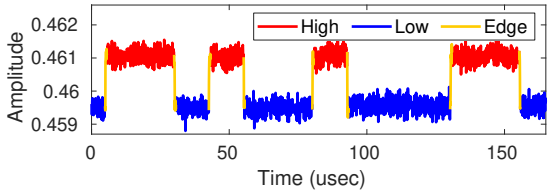

Fig. 7. An example RN16 preamble. The samples can be categorized into High, Low and Edge points based on amplitude.

B. Pre-processing

In this section, we will first introduce how we decouple the CW signals from RN16 preambles as an optimization problem, which includes discussions about defining the objective function, initial values for variables, and constraints. After that, we will discuss how the remaining signals are normalized to reduce the effects of tag-reader channels.

1) Decoupling CW: Tag response signals, including the RN16 preambles that we will use, are superpositions of the received CW signals and signals reflected by the tags. To decouple the CW signals, we need to predict the received CW signals and subtract them from the tag response signals. One naive solution is to analyze the spectrum of clean CW signals before RN16 preambles and estimate what they will be like in the coming samples. However, we have observed that the spectrum of received CW signals is not stable over time. As an example, we extract two adjacent sequences of samples from the same trace and plot part of their spectrums in fig. 6. We can see that the changes over time can not be neglected, and they can bring much error to fingerprinting results if we use the clean received CW signals before RN16 to estimate received CW signals in RN16 preambles.

To reduce the effect of time-varying received CW signals, we use selected samples within the RN16 preamble itself to predict received CW signals in the rest of the samples. We classify the samples into High, Edge, and Low points, as shown in fig. 7. The High and Low points are samples when the tag reflects or absorbs the signal, respectively. The Edge points are samples in the transition between High and Low points. We notice that the Low points in tag response signals can be used as clean received CW signals since the tags are not reflecting any signal during these samples. We can use them to estimate CW signals for the High and Edge points.

Since the Low points we have are discontinuous in the time domain, we cannot directly calculate the spectrum of received CW signals with them and apply it to other samples. We transform it into an optimization problem where we know some samples of the CW signal in the time domain, and we

want to know what its spectrum is most likely to be. We formulate the objective function as follows.

Suppose that we have N samples of the RN16 preamble and the received CW signal in it is $\{y_0, y_1, ..., y_{N-1}\}$. The spectrum of the received CW signal contains N bins. The amplitude and phase of bin i are denoted as A_i and θ_i, respectively, where i is between 0 and $N-1$. In this way, we have

$$y_n = \frac{1}{N}\sum_{i=0}^{N-1} A_i e^{\theta_i} e^{2\pi j n \frac{i}{N}}, n \in [0, N-1] \tag{1}$$

Assume K of the N samples in the RN16 preamble are Low points and their indices in the sample sequence are $\{l_0, l_1, ..., l_{K-1}\} \subset [0, N-1]$. Let Y_L denote the concatenation of observed Low points in the received RN16 preamble, $\tilde{y}_k$ denote the k-th Low point calculated as in (1) using A_n and θ_n, $n = 0, ..., N-1$, and $\widetilde{Y}_k$ denote the sequence of calculated Low points. We have

$$\begin{aligned} Y_L &= \begin{bmatrix} y_{l_0} & y_{l_1} & ... & y_{l_{K-1}} \end{bmatrix} \\ \widetilde{Y}_L &= \begin{bmatrix} \tilde{y}_{l_0} & \tilde{y}_{l_1} & ... & \tilde{y}_{l_{K-1}} \end{bmatrix} \end{aligned} \tag{2}$$

The goal of this optimization problem is to find the spectrum coefficients that minimize the difference between the observed and estimated values of Low points, i.e.,

$$\{A_n, \theta_n\}_{n=0}^{N-1} = arg\min_{A_n, \theta_n} (\left\| Y_L - \widetilde{Y}_L \right\|^2 + \lambda \left\| \{A_n\}_{n=0}^{N-1} \right\|) \tag{3}$$

where the second item in the objective function is the L1-norm regularization and $\lambda > 0$ is the regulation parameter. We add this item to avoid the overfitting problem and will discuss the choice of λ in sec. V-B1.

This optimization problem is prone to converge to local minimums because its objective function is non-convex. To solve this problem, we propose a method to find proper initial values for the optimization variables and also add some constraints based on our observations of the traces.

Initialization: We use the clean received CW signal before RN16 packets to generate initial values for optimization variables. To reduce the effect of noise, we select M groups of N samples, calculate the FFT results, correct the delay, and then take their averages as initial values for the optimization variables. Since the spectrum of the received CW signal changes over time, we would like to use samples as close to the RN16 packet as possible. Our solution is to utilize $N+M-1$ samples $\{y_{cw,0}, y_{cw,1}, ..., y_{cw,N+M-2}\}$ right before the RN16 preamble, and get the M groups of samples with a sliding window of size N as in the following equation, where each row contains N continuous samples.

$$\begin{bmatrix} Y_{cw,0} \\ Y_{cw,1} \\ \vdots \\ Y_{cw,M-1} \end{bmatrix} = \begin{bmatrix} y_{cw,0} & y_{cw,1} & \cdots & y_{cw,N-1} \\ y_{cw,1} & y_{cw,2} & \cdots & y_{cw,N} \\ \vdots & \vdots & \ddots & \vdots \\ y_{cw,M-1} & y_{cw,M} & \cdots & y_{cw,N+M-2} \end{bmatrix} \tag{4}$$

After applying FFT, we can get an $M \times N$ matrix D_{cw}, where

$$d_{cw}(p,q) = \sum_{i=p}^{p+N-1} y_{cw,i} e^{-2\pi j \frac{i-p}{N} q} \tag{5}$$

Our next step is to correct the delay between each row and the initialized points. For i-th row, the delay is $N+M-i-1$ samples. Hence, after correction, we have D'_{cw} where

$$d'_{cw}(p,q) = \sum_{i=p}^{p+N-1} y_{cw,i} e^{-2\pi j \frac{i-p}{N} q} e^{2\pi j \frac{N+M-i-1}{N} q} \tag{6}$$

The initialized values $\{A_{init,n}, \theta_{init,n}\}_{n=0}^{N-1}$ can be obtained by calculating the average across the rows in D'_{cw}:

$$\begin{aligned} A_{init,n} &= \frac{1}{M}\sum_{p=0}^{M-1} \|d'_{cw}(p,n)\| \\ \theta_{init,n} &= \frac{1}{M}\sum_{p=0}^{M-1} \angle d'_{cw}(p,n) \end{aligned} \tag{7}$$

Constraints: To reduce the convergence time as well as avoid the local minimums, we add two constraints to the optimization problem.

Constraint 1: The spectrum of the received CW signal varies within a certain range in the same trace. This constraint is based on our observations of the traces. We take a clean received CW signal, use the above mentioned method to calculate the initial values of optimization variables with the first $N+M-1$ samples, and compare them with the ground truth values calculated with samples after them. Fig. 8 plots the distribution of the absolute difference between the initial values and ground truth of 5 example bands when the delay is over 3000 samples, which is approximately the number of RN16 preamble samples in our experiment setting. The results indicate that the amplitude and phase are not constant over time, but their variations are within a certain range around the initial values, especially for the amplitude. Hence, we add the boundary constraints for $\{A_n, \theta_n\}_{n=0}^{N-1}$:

$$\begin{aligned} A_{init,n} - \gamma_{1,n} \leqslant & A_n \leqslant A_{init,n} + \gamma_{2,n} \\ \theta_{init,n} - \beta_{1,n} \leqslant & \theta_n \leqslant \theta_{init,n} + \beta_{2,n} \end{aligned} \tag{8}$$

where $\gamma_{1,n}$, $\gamma_{2,n}$, $\beta_{1,n}$, $\beta_{2,n}$ define the boundaries of the search ranges of amplitude and phase. For each reader, we set the boundaries based on the evaluation of one received CW signal trace. For some FFT bins with very low SNR, the phase offsets can be unpredictable, and we set their search ranges as $[-\pi, \pi]$.

Constraint 2: The noise level of High Points after subtracting the CW signal should not be higher than the noise level of original High points. After decoupling the CW signal, the residual signal should only contain the signal reflected by the tag. Since Low points should be closed to zero, and the shapes of edges are hard to predict due to different electromagnetic properties, we only consider the noise level of High points. After subtracting the CW signal, the High points will contain only the signal reflected from the tag. Compared

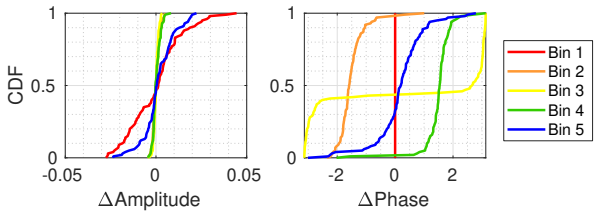

Fig. 8. CDF of amplitude and phase differences between the initial values and ground truth in 5 example FFT bins

to the received CW signal, the reflected signal has a much lower power level, thus its noise floor should be no larger than the original noise floor.

2) Normalization: Until now, the full optimization problem has been defined and we can use it to estimate the spectrum variables of the received CW signal. We recover the received CW signal within the RN16 preamble by applying IFFT to the optimization results $\{A_n, \theta_n\}_{n=0}^{N-1}$. And we use the recovered CW signal and the received RN16 preamble to calculate the signal reflected by the tag. The received CW signal can be described as

$$Y_{CW} = H_{reader} H_{TxRx} X_{CW} + N \tag{9}$$

where H_{reader} represents the effect of readers' hardware imperfections on the signal, H_{TxRx} is the channel between the transmitting and receiving antennas of the reader, X_{CW} is the CW signal, and N is the noise. The tag response signal we obtain on the receiver side can be expressed as

$$Y_{Tag} = H_{reader}(H_{TxRx} X_{CW} + H_{TxTagRx} X_{CW} X_{Tag}) + N \tag{10}$$

where $H_{TxTagRx}$ is the channel from transmitting to receiving antennas reflected by the tag, X_{Tag} is the signal of tags modulated by on-off keying, and other parameters are defined as in previous equations.

In our implementation, we use a software-defined radio as the RFID reader and set its sampling rate to 20 MHz, which means signals traversing different paths will arrive within one sampling duration if the path length differences are within 15 meters. Since the communication range of RFID tags is 3-10 meters, we can view wireless channels in the RFID system as one-tap channels. This is also true for commercial readers, since their sampling rates are mostly less than 20 MHz.

Notice that we focus on the shape of the tag signal X_{Tag}, but not the absolute value of the waveform affected by channel conditions. Thus, with the noise neglected, we can approximate the shape of X_{Tag} by

$$\widetilde{X}_{Tag} = \frac{Y_{Tag} - Y_{CW}}{Y_{CW}} \propto X_{Tag} \tag{11}$$

where Y_{Tag} and Y_{CW} are defined in (9) and (10). We further normalize $\widetilde{X}_{Tag}$ with the median of values of High points and will use the normalized $\widetilde{X}_{Tag}$ in the following procedures.

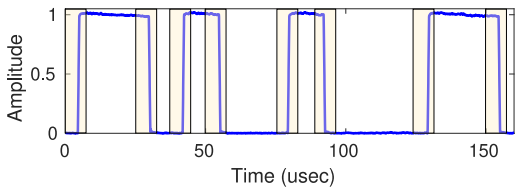

Fig. 9. An example RN16 preamble signal after pre-processing. We use the highlighted samples to extract features.

C. Feature Extraction

After obtaining the normalized tag signal, our next step is to extract features from it. Multiple features have been used to fingerprint RFID tags, such as time interval error, covariance-based distribution, and power spectrum density [10], [11]. However, relying on only one or the combination of a few features might limit the maximum number of distinguishable tags. Furthermore, analyzing a large number of traces and manually picking features from them can also cost much time and effort. In FILES, we utilize the spectrum of information-rich samples around the edges and leverage neural networks to automatically generate a feature space that is sufficient to accurately distinguish a large number of tags.

Instead of feeding the entire processed RN16 preamble signal to the neural network, we only use the edge portions that contain more distinctive transient features of the tags [14]. This helps to speed up the training process and reduce the size of neural networks. Fig. 9 illustrates an RN16 preamble waveform after pre-processing. As can be seen from the figure, after decoupling the CW signal, most Low points are very close to zero. They are not supposed to contain any tag information since the tag is silent during these samples. Moreover, some distinctive properties, such as frequency offset, can be extracted using part of the signals and do not require the whole signal. Hence, we remove some redundant Low and High points that do not contain much information, and only select the samples around the edges, as the highlighted sections in fig. 9. After selecting these samples, we reassemble the sample pieces into a sequence and calculate the transient features in the frequency domain.

Notice that this approach will change the frequency components of each tag. However, it keeps distinctive features in tag signals and will not affect the classification accuracy.

D. Classification

FILES trains neural networks to classify RFID fingerprints, which does not depend on computational-expensive feature selection algorithms. We choose fully connected neural networks for the classification task. This choice is because of the limited number of tuning parameters and its low training time, compared to other architectures such as convolutional neural networks (CNN) and recurrent neural networks.

The neural network model takes the transient features extracted in previous steps as input and produces the predicted tag ID as output. To make the data type compatible with the machine learning libraries we use in implementation, the

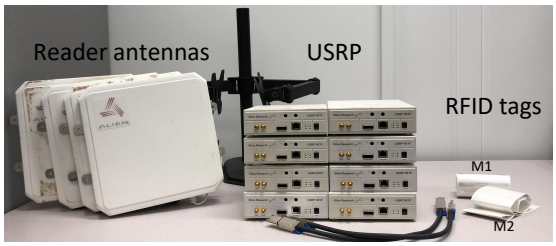

Fig. 10. Equipment used in implementation

feature arrays of complex numbers are transformed into arrays of real numbers by extracting and reorganizing their real and imaginary parts. We use the ELU activation function and the Adam optimizer for the network, and define its structure to contain 5 layers and 200 neurons per layer. Intuitively speaking, earlier layers in the network will learn a feature space that will be best suited to the classification task.

For authentication purposes, we compare the tag ID produced by the neural network with the claimed ID decoded from the EPC packet. If the two IDs match, FILES will accept this trace as authentic. Otherwise, it will suggest to the RFID application that the trace might not be from the claimed tag.

IV. Implementation

We build a prototype of FILES using N210 USRP software-defined radios as readers to authenticate commercial UHF RFID tags. They are connected to a desktop with the Intel i7-7700 processor to run the tag reader program from [15]. Each USRP radio is equipped with an SBX daughterboard and connected to two Alien ALR-8696 RFID antennas, one for transmitting and another for receiving. The RFID tags are of two models, Alien Squiggle ALN-9740 and Impinj H47. We will refer to them as M1 and M2 tags respectively in the rest of this paper. The reader and the tags work at 910 MHz. They both use FM0 encoding and the RN16 preambles contain 8 edges. Note that FILES can also work with the Miller encoding supported in the EPC C1G2 protocol, since they are all modulated with on-off keying. Fig. 10 shows the equipment we used in the implementation.

To receive signals with the 20 MHz sampling rate, we use an additional USRP radio as the *monitor*. We use monitors instead of increasing the sampling rate in the reader program because a large number of samples will get missed due to the overflow problem when the reader's sampling rate is set above 5 MHz in our settings. A monitor has the same hardware setting as a reader. But instead of sending queries and communicating with tags, it only records the signal at the higher sampling rate and is connected to another desktop. The reader and monitor are synchronized through a MIMO cable.

With a sampling rate of 20 MHz, an RN16 preamble contains around 3000 samples. We take $N = 3200$ samples in data acquisition to ensure that the transient features of the first and last edges are included. When solving the optimization problem in pre-processing, we apply a specialized interior-point method that uses the preconditioned conjugate gradients algorithm to compute the search direction [16]. The neural network for classification is fully connected. It contains 5 layers and has 200 neurons in each layer. We will discuss the choices of other parameters in sec. V.

TABLE I
Experiment settings of datasets

Dataset	#Tags	Model	#Tag Loc.	#Reader	Env.
1	50	M1,M2	3	2	Lab
2	20	M1,M2	10	1	Lab
3	20	M1,M2	3	4	Lab
4	20	M1,M2	3	1	Hall

V. Evaluation

In this section, we will first introduce the datasets used for the evaluation. After that, we will present microbenchmark test results, system-level test results, and security test results. In the microbenchmark test, we focus on the procedures within FILES and discuss how values of significant parameters will affect the system performance. In the system-level test, we evaluate our system under different settings and compare it with other RF fingerprinting methods for RFID systems. In the security test, we realize a replay attack to validate the robustness of FILES against counterfeiting.

A. Datasets

In the rest of this paper, we redefine the term *reader* as a combination of the USRP reader and monitor as described in sec. IV. A *different reader* means that both radios are different. In Table 1, we summarize the experiment settings of datasets we collected over two weeks. The tag-reader distances of different tag locations vary from 30 cm to 200 cm, and their tag-reader orientations vary from -90° to 90°. Each trace contains 120 successful reads. Datasets 1-3 are collected in a multipath-rich indoor lab environment. Dataset 4 is collected in a hall with fewer multipaths compared to the lab.

B. Microbenchmark test

1) CW signal estimation: In sec. III-B, we propose an algorithm to decouple the CW signal from the tag response signal. To evaluate its performance, we select 1000 sample sequences of length $N = 3200$ from the clean received CW signals with different tag locations for each reader. For each sample sequence, we assume 1700 samples are the Low points, use them to estimate the other 1500 samples (assumed High and Edge points), and compare the estimation with the ground truth. Fig. 11 illustrates the average mean square error (MSE) with different values for the regulation coefficient λ in the optimization objective function. We observe that when λ is larger than 0.01, the optimization algorithm fails to find a feasible solution in most cases. When λ is too small, the optimization problem is likely to converge to local minimums. We use $\lambda = 0.005$ in the following evaluations.

Fig. 12 shows the average processing time of one sample sequence with different λ. It takes around 16 msec to resolve

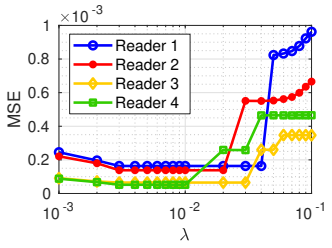

Fig. 11. MSE for estimating CW signal with different values for the regulation coefficient λ

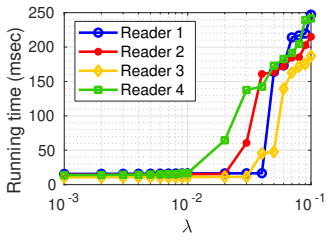

Fig. 12. Running time for optimizing the signal of one time slot. The average time is less than 16 msec.

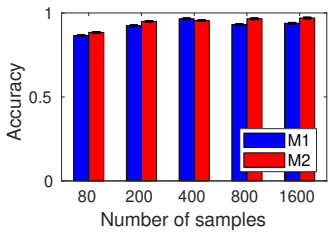

Fig. 13. Authentication accuracy with different numbers of selected samples

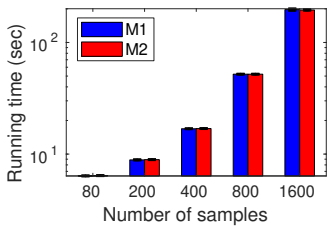

Fig. 14. Model training time with different numbers of selected samples

a sequence when $\lambda = 0.005$. Considering that one time slot takes around 10 msec and the reader usually needs several time slots to interrogate one tag, the computational overhead of optimization can be tolerated in RFID systems.

2) Classification accuracy and feature size: In sec. III-C, we propose an algorithm to extract fingerprints with selected samples around the edges. In this part, we will discuss how the number of selected samples would affect the classification accuracy. We take 80000 RN16 preambles from traces in Dataset 1 and use 5-fold cross-validation to evaluate the classification accuracy. The evaluation result is shown in fig. 13. When the number of selected samples is less than 800, the accuracy increases with the sample number since more tag features are captured. However, when the selected sample number increases to 1000, the accuracy decreases slightly. This is because the spectral features become sparse when too many samples are used. Some bands do not contain tag information but introduce noise, which causes a decrease in accuracy.

Fig. 14 shows the average training times of neural network models with different numbers of selected samples. We can see that the average training time increases with the number of selected samples, which is also the input size of the neural network. In the following evaluations, we select 400 samples in total around the edges, which gives high accuracy and takes a relatively short training time.

C. System-level test

In system-level tests, we compare our system with the following RF fingerprinting works for RFID tags. Similar to FILES, these systems aim to build efficient authentication schemes for COTS tags without adding new infrastructure or introducing high overhead.

- *CNN* [12]: This system trains a CNN model to classify tags with the full RN16 signal as input. We use the open-source code published by its authors. For a fair comparison, we disable its federated learning feature.
- GenePrint [11]: We implement GenePrint with a C4.5 classifier that explores the similarity between response pulses. We use the covariance-based distribution and power spectrum density as fingerprints and use 80 bins for the distributions.
- ∂_{TIE} [9], [10]: This method extracts the time interval errors of the symbol duration as features. We use the 3-nearest neighbor classifier to classify them.
- *Spectral* [9]: We extract the features as described in [9] and classify the fingerprints with a support vector machine (SVM) classifier.

Overall Performance In this part, we evaluate the overall performance of all systems with traces in Dataset 1. Each trace contains samples of 120 successful reads of one tag. In every trace, we randomly select 80% of the reads as the training dataset and use the remaining 20% for testing. The classification accuracy results are presented in table II. For the ∂_{TIE} and *Spectral* methods, we notice a large gap between our results and results in corresponding papers. We believe that this is because the original systems use oscilloscopes with a 1 GHz sampling rate, whereas our traces are collected with a 20 MHz sampling rate. Due to hardware limitations, we could not collect traces with the 1 GHz sampling rate to replicate the experiments. Thus we directly list their results in the table.

Comparing the results of the 20 MHz sampling rate, we can see that *CNN* has the best performance for both tag models. We can conclude that neural networks can be a very powerful tool for classifying the traces, and using more samples (each RN16 preamble has around 12000 samples in our experiment setting) for fingerprinting helps extract more features. We observe that the accuracy of ∂_{TIE} is very low with the 20 MHz sampling rate. This happens because the differences in time interval error can be very small among tags of the same model and are difficult to capture with a low sampling rate. We also observe that the accuracy of $Spectral$ is closed to random guesses with a low sampling rate, which can be explained by differences in sampling rates and the various locations and readers of the traces. In [9], the authors have also noticed a decrease in accuracy from 100% to 37.2% when traces collected under different conditions instead of a controlled environment are used. Compared with other systems, FILES has the second-highest accuracy of 87.92% and 89.2% for M1 and M2 tags, respectively. Though not as accurate as *CNN*, FILES achieves slightly lower accuracy with 97% fewer samples and uses a lightweight neural network structure that runs faster than the CNN model during both training and testing.

Performance across tag locations In this part, we evaluate the systems when training and testing traces are collected with different tag locations. The received signal reflected by tags contains channel information, where the tag-reader channel plays an important role. An ideal fingerprinting method should focus on the tag features and be robust across tag locations. For this evaluation, we use Dataset 2 which contains traces

TABLE II
CLASSIFICATION ACCURACY WITH DATASET 1

	Accuracy (20 MHz) M1	Accuracy (20 MHz) M2	Accuracy (1 GHz) ALN9540
FILES	0.8792	0.8920	-
CNN	0.9650	0.9495	-
GenePrint	0.6680	0.7023	-
∂_{TIE}	0.0878	0.1124	0.714
Spectral	0.0546	0.0673	0.996

† We use the result in [9], [17] as benchmark of the 1 GHz sampling rate where the traces are collected at same location.

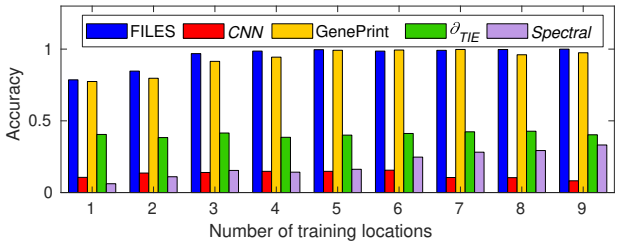

Fig. 15. Accuracy with different numbers of tag locations in the training dataset. Traces of the remaining tag locations are used as the testing dataset.

collected for 20 tags at 10 different tag locations. We randomly choose traces collected from 1-9 locations to train the models and use traces of the remaining locations for testing. Fig. 15 shows the average accuracy with different numbers of training locations. FILES can classify tags at unseen locations with an accuracy of 78.58% when trained with traces from only one tag location. When traces of more tag locations are used for training, both FILES and GenePrint can achieve high accuracies of over 99%. The accuracies of *CNN* are close to random guesses. We think this is because the CNN model learns the combined features of channels, readers, and tags, but is incapable to separate the tag features from the combined features. When the channel condition changes, there will be no good match for the new combined features in the training dataset and this leads to the performance downgrades.

Performance across readers In this part, we investigate the performance when we use different readers for training and testing. In this evaluation, we use Dataset 3 which contains traces of four readers. We train the models with traces of combinations of Readers 1-3 and use traces of Reader 4 to test the models. Fig. 16 illustrates the classification accuracy with different training datasets. From the results, we can first conclude that the hardware imperfections of readers have a significant effect on RF fingerprints of RFID tags. This can be seen from the decrease in accuracy of all systems. The results also prove that FILES is capable to reduce the effect of readers on tag fingerprints. With traces collect from one reader, FILES can classify traces from an unseen reader with an accuracy of up to 86.55%. When traces of three readers are used for training, it can classify traces from an unseen reader with an accuracy of 97.00%.

Performance across environments In this part, we investigate the performance of FILES across different multipath environments. We randomly choose 70% data from Dataset

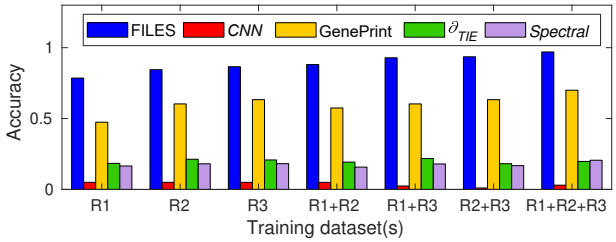

Fig. 16. Accuracy when the training and testing datasets are collected from different readers. Traces of reader 4 are used as the testing dataset.

TABLE III
CLASSIFICATION ACCURACY ACROSS ENVIRONMENTS

Testing Env.	Lab (30% traces) M1	Lab (30% traces) M2	Hall M1	Hall M2
FILES	0.9890	0.9996	0.9462	0.9682
CNN	0.9863	0.9745	0.5182	0.5074
GenePrint	0.9200	0.9576	0.8796	0.9163

† The models are trained with 70% of lab traces.

1 to train a model and test it with the remaining data from Dataset 1 and traces from Dataset 4 separately. Table III shows the results. We can see that all systems perform well when testing traces are collected in the same environment as the training traces. However, when tested with traces collected in the unseen hall environment, the accuracy of *CNN* decreases to around 50%. The performances of FILES and GenePrint are also affected by the change of environments, but the accuracy decreases less than *CNN*.

D. Security Test

We implement a replay attacker using a USRP N210 radio and two VERT900 vertical antennas. The antennas are placed 1 cm away from the tags. The attacker first eavesdrops on the data when the tag responds and later transmits the recorded signal. We apply the replay attack to 10 tags of each model and collect 120 successful reads in each trace. The traces are fed into a model trained with Dataset 1 for authentication. The false acceptance rates (FAR) of the tags are shown in fig. 17.

The results indicate that FILES can effectively defend replay attacks with an average error rate of 9.6%. Notice that we cannot defend against replay attacks if the attacker can transmit the identical signal as RFID tags. If the attacker can obtain its impulse in advance, it can cancel its signature by applying self-interference cancellation when it replays the signal. To the best of our knowledge, existing RFID security works based on physical layer solutions can not effectively prevent such an attacker as well. However, such an attacker is very difficult to implement. It requires additional computational capability and high samplings rate to realize digital and analog interference cancellation and recover the signal [11].

VI. RELATED WORK

RF fingerprinting There has been a growing interest in RF fingerprinting for devices in communication systems. The intuition is to utilize the instinctive properties caused by the

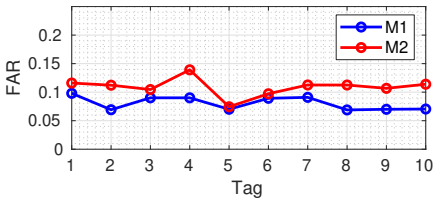

Fig. 17. False acceptance rates when deploying replay attack

various hardware imperfections from the received signal to uniquely identify the devices. In [5], the authors leverage frequency offsets and constellation errors to identify 802.11b network interface cards. SecureArray [6] uses the angle-of-arrival as signatures to identify Wi-Fi clients. Some more recent works introduce machine learning tools to classify the fingerprints. Ref. [7] identifies nominally identical software-defined radios using a CNN architecture with unprocessed I/Q samples. The authors of [8] apply an optimized finite impulse response filter to strengthen the fingerprints and utilize the CNN learning model to identify Wi-Fi and ADS-B devices.

Physical-layer authentication schemes for RFID systems Besides RF fingerprinting, researchers have also investigated other secure mechanisms for RFID systems to defend against impersonation attacks. Hu-Fu [18] and Butterfly [19] both require an additional tag for authentication purposes. Hu-Fu takes advantage of the inductive coupling of closely placed tags. It can distinguish the replayed signals since the attacker device does not couple with the retained tag like authentic tags. Butterfly uses the difference in signals of two tags to reduce the effect of environments on RF fingerprints. In [20], the authors take the electrical energy stored in the resistor-capacitor circuit in tag chips as fingerprints. Comparing with FILES and existing RF fingerprinting works, they require additional hardware [18], [19] or introduce large overheads in the communication process to collect fingerprints [20].

VII. Conclusion

In this paper, we propose FILES, a learning-based RF fingerprinting authentication scheme for RFID systems. FILES analyzes the received CW signal that contains information about channels and hardware features of readers, cancels it out from the tag response signal, and uses a lightweight neural network to classify features extracted from the processed tag signal. The evaluation results show that FILES can achieve high accuracy with different readers, tag locations, and environments. FILES is designed based on the EPC C1G2 protocol and compatible with COTS tags. It can be used as an additional secure layer to existing RFID systems.

Acknowledgment

We would like to thank the reviewers for their thoughtful comments and constructive suggestions. This work is partly supported by NSF CNS-2007581 and ECCS-2128567 awards.

References

[1] A. Sarac, N. Absi, and S. Dauzère-Pérès, "A literature review on the impact of RFID technologies on supply chain management," *International Journal of Production Economics*, vol. 128, no. 1, pp. 77–95, 2010.

[2] K. Koscher, A. Juels, V. Brajkovic, and T. Kohno, "EPC RFID tag security weaknesses and defenses: passport cards, enhanced drivers licenses, and beyond," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, 2009, pp. 33–42.

[3] S. Pradhan, E. Chai, K. Sundaresan, L. Qiu, M. A. Khojastepour, and S. Rangarajan, "Rio: A pervasive RFID-based touch gesture interface," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, 2017, pp. 261–274.

[4] J. Xu, W. Sun, and K. Srinivasan, "Embracing collisions to increase fidelity of sensing systems with COTS tags," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 2, pp. 1–20, 2021.

[5] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, 2008, pp. 116–127.

[6] J. Xiong and K. Jamieson, "Securearray: Improving WiFi security with fine-grained physical-layer information," in *Proceedings of the 19th Annual International Conference on Mobile Computing and Networking*, 2013, pp. 441–452.

[7] S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury, "Deep learning convolutional neural networks for radio identification," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 146–152, 2018.

[8] F. Restuccia, S. D'Oro, A. Al-Shawabka, M. Belgiovine, L. Angioloni, S. Ioannidis, K. Chowdhury, and T. Melodia, "Deepradioid: Real-time channel-resilient optimization of deep learning-based radio fingerprinting algorithms," in *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2019, pp. 51–60.

[9] B. Danev, T. S. Heydt-Benjamin, and S. Capkun, "Physical-layer identification of RFID devices." in *USENIX Security Symposium*, 2009, pp. 199–214.

[10] D. Zanetti, B. Danev, and S. Čapkun, "Physical-layer identification of UHF RFID tags," in *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking*, 2010, pp. 353–364.

[11] J. Han, C. Qian, P. Yang, D. Ma, Z. Jiang, W. Xi, and J. Zhao, "Geneprint: Generic and accurate physical-layer identification for UHF RFID tags," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 846–858, 2015.

[12] M. Piva, G. Maselli, and F. Restuccia, "The tags are alright: robust large-scale RFID clone detection through federated data-augmented radio fingerprinting," in *Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 2021, pp. 41–50.

[13] EPCglobal, "EPC UHF Gen2 air interface protocol." [Online]. Available: www.gs1.org/standards/rfid/uhf-air-interface-protocol

[14] A. I. Sunny, G. Y. Tian, J. Zhang, and M. Pal, "Low frequency (LF) RFID sensors and selective transient feature extraction for corrosion characterisation," *Sensors and Actuators A: Physical*, vol. 241, pp. 34–43, 2016.

[15] N. Kargas, F. Mavromatis, and A. Bletsas, "Fully-coherent reader with commodity SDR for Gen2 FM0 and computational RFID," *IEEE Wireless Communications Letters*, vol. 4, no. 6, pp. 617–620, 2015.

[16] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l_1 -regularized least squares," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–617, 2007.

[17] B. Danev, D. Zanetti, and S. Capkun, "On physical-layer identification of wireless devices," *ACM Computing Surveys (CSUR)*, vol. 45, no. 1, pp. 1–29, 2012.

[18] G. Wang, H. Cai, C. Qian, J. Han, X. Li, H. Ding, and J. Zhao, "Towards replay-resilient RFID authentication," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 385–399.

[19] J. Han, C. Qian, Y. Yang, G. Wang, H. Ding, X. Li, and K. Ren, "Butterfly: Environment-independent physical-layer authentication for passive RFID," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 4, pp. 1–21, 2018.

[20] X. Chen, J. Liu, X. Wang, H. Liu, D. Jiang, and L. Chen, "Eingerprint: Robust energy-related fingerprinting for passive RFID tags," in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2020, pp. 1101–1113.