Rhythmic RFID Authentication

Jiawei Li, *Student Member, IEEE*, Chuyu Wang, Ang Li, Dianqi Han, *Member, IEEE*, Yan Zhang, *Member, IEEE*, Jinhang Zuo, Rui Zhang, *Member, IEEE*, Lei Xie, *Member, IEEE*, and Yanchao Zhang, *Fellow, IEEE*

***Abstract*— Passive RFID technology is widely used in user authentication and access control. We propose RF-Rhythm, a secure and usable two-factor RFID authentication system with strong resilience to lost/stolen/cloned RFID cards. In RF-Rhythm, each legitimate user performs a sequence of taps on his/her RFID card according to a self-chosen secret melody. Such rhythmic taps can induce phase changes in the backscattered signals, which the RFID reader can detect to recover the user's tapping rhythm. In addition to verifying the RFID card's identification information as usual, the backend server compares the extracted tapping rhythm with what it acquires in the user enrollment phase. The user passes authentication checks if and only if both verifications succeed. We also propose a novel phase-hopping protocol in which the RFID reader emits Continuous Wave (CW) with random phases for extracting the user's secret tapping rhythm. Our protocol can prevent a capable adversary from extracting and then replaying a legitimate tapping rhythm from sniffed RFID signals. Comprehensive user experiments confirm the high security and usability of RF-Rhythm with false-positive and false-negative rates close to zero.**

***Index Terms*— RFID security, authentication.**

I. Introduction

PASSIVE (battery-less) RFID technology has been widely used in user authentication and access control. An RFID authentication system comprises a backend server, RFID readers, and RFID cards which refer to identification cards with embedded RFID tags. For convenience only, we use the terms RFID cards and tags interchangeably hereafter whenever no confusion arises. An RFID reader sends wireless signals to interrogate a nearby RFID card, which returns its tag ID by backscattering the reader's signals. The reader then forwards the tag ID to the backend server. If a matching ID can be found in the database, the user passes authentication and is permitted to access protected electronic or physical resources such as entering a gated area.

Lost/stolen/cloned RFID cards pose the most critical threat to RFID authentication systems. In particular, RFID cards are often of small size and can be easily lost or stolen; they can also be cloned with many cheap off-the-shelf tools. More specifically, most commodity RFID cards do not support cryptographic operations, so the RFID reader-card communications are in plaintext and vulnerable to eavesdropping with cheap tools online. The adversary can then easily exploit the sniffed card information to make a clone. Since RFID cards are not password-protected, the adversary can use a lost/stolen/cloned RFID card—referred to as an **adversarial RFID card** henceforth—to pass authentication and impersonate the legitimate user to get illegal access to a gated area or sensitive physical/electronic resources protected by RFID-based access control. An effective countermeasure can be two-factor authentication which requires the RFID user to present the second piece of identification information. One such solution requires the RFID user to additionally input a PIN code on a keypad [1]. This solution not only diminishes the convenience of contactless RFID authentication but also requires a nontrivial infrastructure update to existing RFID systems. Another plausible solution is exploring commercial mobile 2FA solutions such as Duo Mobile [2], which require the RFID user to manually acknowledge an authentication request on his/her enrolled smartphone/smartwatch. This solution needs the RFID user to own and always carry a smartphone with good network connectivity, which may not be feasible in practice.

We propose RF-Rhythm, a secure and usable two-factor RFID authentication system with strong resilience to adversarial RFID cards. In RF-Rhythm, each legitimate user performs a sequence of taps on his/her RFID card according to a self-chosen secret melody. Such rhythmic taps can induce phase changes in the backscattered RFID signals, which the RFID reader can detect to recover the user's rhythm. In addition to verifying the card ID as usual, the backend server compares the recovered rhythm with what it acquires in the

Manuscript received 25 July 2021; revised 1 July 2022; accepted 21 August 2022; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor A. Khreishah. This work was supported in part by the U.S. National Science Foundation (CAREER) under Grant CNS-1651954, Grant CNS-1824355, Grant CNS-1933047, Grant CNS-1933069, and Grant CNS-2055751; in part by the National Natural Science Foundation of China under Grant 61902175, Grant 61872174, Grant 61832008, and Grant 61872173; and in part by the Jiangsu Natural Science Foundation under Grant BK20190293. *(Corresponding author: Yanchao Zhang.)*

Jiawei Li, Ang Li, and Yanchao Zhang are with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: jwli@asu.edu; anglee@asu.edu; yczhang@asu.edu).

Chuyu Wang and Lei Xie are with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China (e-mail: chuyu@nju.edu.cn; lxie@nju.edu.cn).

Dianqi Han is with the Computer Science and Engineering Department, University of Texas at Arlington, Arlington, TX 76019 USA (e-mail: dianqi.han@uta.edu).

Yan Zhang is with the Electrical and Computer Engineering Department, The University of Akron, Akron, OH 44325 USA (e-mail: yzhang1@uakron.edu).

Jinhang Zuo is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: jzuo@andrew.cmu.edu).

Rui Zhang is with the Computer and Information Sciences Department, University of Delaware, Newark, DE 19716 USA (e-mail: ruizhang@udel.edu).

Digital Object Identifier 10.1109/TNET.2022.3204204

user enrollment phase. The user passes authentication only if both verifications succeed.

The security, usability, and feasibility of RF-Rhythm lie in many aspects. First, a user can easily select a secret song segment which is familiar to him/herself but very difficult for others to guess. Second, different users may interpret the same song segment in various ways, resulting in diverse rhythmic tap patterns on the card. This means that even if the adversary knows the secret song segment, it may still have great difficulty performing the correct tapping rhythm on the RFID card. Third, RF-Rhythm is naturally resilient to traditional replay and relay attacks on RFID authentication systems. Fourth, the phase information of backscattered signals is readily available on commodity RFID readers, so RF-Rhythm only needs a minor software update to an existing RFID authentication system. Finally, RF-Rhythm applies to commodity RFID cards and does not need the user to carry any other device.

The design of RF-Rhythm faces two critical challenges.

1. **Rhythm detection and classification: how to detect and verify the tapping rhythm from noisy RFID signals?** Rhythmic taps are performed on the RFID card and have to be indirectly extracted from noisy backscattered signals. We explore various signal processing techniques to process noisy raw phase data for extracting a reliable tapping rhythm. We also use machine learning techniques to train a classifier the backend server uses to validate an extracted tapping rhythm.
2. **Rhythm anti-eavesdropping, i.e., how to prevent the adversary from acquiring the user's tapping rhythm from sniffed RFID signals?** The adversary can easily eavesdrop on the open RFID channel and then behave in the same way as the RFID reader to decode the user's tapping rhythm from sniffed RFID signals. It can then repeat the rhythmic taps on adversarial RFID card to attempt impersonating the legitimate user. We tackle this challenge by a novel phase-hopping protocol in which the RFID reader emits Continuous Wave (CW) with random phases for extracting the user's tapping rhythm. Since the adversary does not know the phase-hopping sequence, it can no longer extract the correct tapping rhythm from sniffed RFID signals.

We evaluate the security and usability of RF-Rhythm by comprehensive experiments on Impinj RFID readers, commodity passive tags, and USRP devices. Our experiments involve 19 volunteers from two countries and explore three representative machine learning techniques, including Support Vector Machine (SVM), Neural Networks (NN), and Convolutional Neural Networks (CNN). We show that RF-Rhythm is highly secure with false-positive and false-negative rates close to zero. In addition, we demonstrate the high resilience of RF-Rhythm to brute force, visual eavesdropping, and RF eavesdropping attacks. We also confirm the high usability of RF-Rhythm by a user survey.

The rest of this paper is organized as follows. Section II gives some necessary background about RFID systems. Section III describes the adversary model. Section IV provides an overview of RF-Rhythm. Section V details the design of RF-Rhythm. Section VI presents the phase-hopping protocol for anti-eavesdropping. Section VII reports the experimental evaluation of RF-Rhythm. Section VIII briefs the related work.

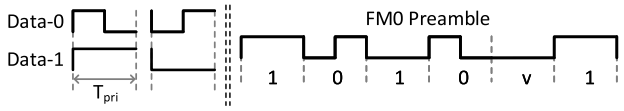


Fig. 1. FM0 baseband symbols and preamble.

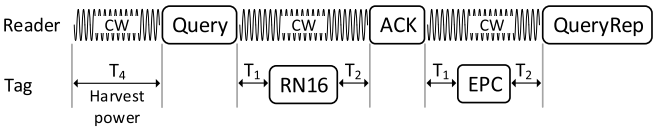


Fig. 2. The basic EPC Gen-2 query protocol with a single RFID card.

II. Basics of Passive UHF RFID Systems

Passive RFID systems can be classified into low-frequency, high-frequency, and ultra-high-frequency (UHF) types. We focus on UHF systems which are dominating the RFID market. The extension of RF-Rhythm to low-frequency and high-frequency RFID systems are left as future work. In this section, we introduce some necessary background about passive UHF RFID systems to help illustrate the subsequent RF-Rhythm design. A typical RFID system consists of a backend server, readers, and RFID cards. The RFID reader sends both modulated commands and continuous wave (CW). The RFID card sends back its data by exploring the energy harvested from the reader's signals to switch its input impedance between two states and thus modulate the backscattered signal. EPC Gen 2 [3] is the most popular UHF RFID standard and assumed hereafter.

RFID cards encode the backscattered data using either FM0 baseband or miller modulation. We only consider FM0 encoding in this paper, but our work can easily extend to miller modulation. Fig. 1 shows the basic FM0 symbols. FM0 inverts the baseband phase at every symbol boundary with an additional mid-symbol phase inversion for each data-0. The duration of an FM0 symbol is denoted by $T_{\text{pri}} = 1/\text{BLF}$, where BLF represents the backscatter link frequency ranging from 40 kHz to 6400 kHz [3]. To ease our presentation, we assume BLF equal to 40 kHz, corresponding to $T_{\text{pri}} = 25$ μs.

Fig. 2 shows the basic query protocol in EPC Gen-2 [3].

1) The reader emits CW of length T_4 for the RFID card to harvest and store energy.
2) The reader sends a Query command followed by CW of length $T_1 + T_2 + T_{\text{RN16}}$. During this CW period, the card backscatters an RN16 message comprising a 6-bit preamble, a 16-bit random number, and one dummy bit.
3) The reader sends an ACK followed by CW of length $T_1 + T_2 + T_{\text{EPC}}$. During this CW period, the card backscatters its EPC (Electronic Product Code).
4) The reader sends QueryRep to end this session.

EPC Gen-2 [3] gives recommendations for the above timing parameters. Let RTcal represent the duration of Interrogator-to-Tag calibration symbol, which is specified in the reader

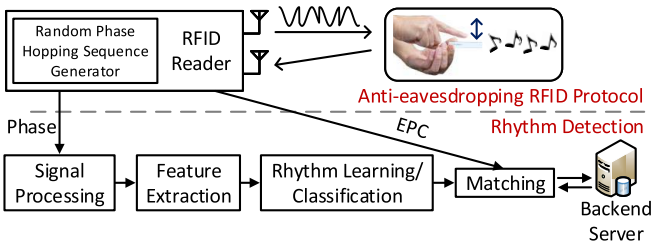


Fig. 3. The RF-Rhythm system flowchart.

configuration and set to $\mathrm{RTcal} = 72\ \mu$s in our implementation. Also let FrT be the frequency tolerance of FM0 baseband signals, which equals 4% for BLF = 40 KHz. We have $T_4 = 2\mathrm{RTcal} = 144\ \mu$s and $75\ \mu\text{s} \leq T_2 \leq 500\ \mu$s. In addition, the maximum, minimum, and nominal values of T_1 are 262 μs, 238 μs, and 250 μs, respectively.

III. Adversary Model

We assume an adversary $\mathcal{A}$ who attempts to use an adversarial RFID card to pass authentication checks and thus impersonate the legitimate user. Since the adversarial RFID card has identical information to that of the corresponding legitimate card, $\mathcal{A}$ can succeed in a traditional RFID authentication system if no additional countermeasure is adopted. $\mathcal{A}$ knows how RF-Rhythm works and can perform rhythmic taps on the RFID card with fingers or even a fully programmable robotic arm. We assume that $\mathcal{A}$ does not know the legitimate user's secret song segment and can try the following attack strategies.

- **Brute force**: $\mathcal{A}$ performs totally random rhythmic taps on the RFID card.
- **Visual eavesdropping**: $\mathcal{A}$ observes the legitimate user's tapping behavior, e.g., by shoulder surfing or a spy camera, and then tries to emulate it.
- **RF eavesdropping**: $\mathcal{A}$ sniffs all the PHY communication traces between the RFID reader and card to attempt recovering and performing the legitimate user's rhythmic taps.

IV. System Overview

RF-Rhythm consists of an enrollment phase and a verification phase, and its major modules are depicted in Fig. 3,

During the enrollment phase, the legitimate user first selects an arbitrary song segment familiar to him/herself. Then the user performs rhythmic taps on his/her RFID card in accordance with his/her own interpretation of the chosen song segment, e.g., by singing it silently. The user's tapping rhythm is referred to as his/her secret rhythm hereafter.

The security of RF-Rhythm relies on the secrecy of the chosen song segment and also the user's likely unique tapping rhythm for it. In particular, since there are numerous song segments available, the adversary can hardly guess the selected song segment of a target user; an advanced user such as a musician can even self-compose the song segment. In addition, people may have very subjective mental interpretations about the same song segment, resulting in very different tapping rhythms.

The backend server handles the enrollment request as follows. First, it acquires the EPC of the user's RFID card through the reader with the protocol in Fig. 2. Second, it instructs the user to perform rhythmic taps on the RFID card, which would induce phase changes in the backscattered signals received by the reader. Third, the server invokes a *Signal Processing* module to extract reliable phase data from noisy backscattered signals. Fourth, it uses a *Feature Extraction* module to obtain a feature vector that characterizes the use's tapping rhythm. Finally, it asks the user to repeat the rhythmic taps multiple times and then feeds all the resulting feature vectors into a *Rhythm Learning* module to train a high-quality binary rhythm classifier for this user.

In the verification phase, the backend server first explores the RFID card for its EPC with the protocol in Fig. 2. If the EPC is found in the database, the server instructs the reader to execute multiple rounds of the protocol again in Fig. 2. RF-Rhythm is highly usable in the sense that the RFID user just needs to perform his/her secret tapping rhythm multiple times without the need to know when the server starts to extract it in both the enrollment and verification phases. The server invokes the same Signal Processing and Feature Extraction modules to extract a candidate tapping rhythm in each round, which is then tested with the trained rhythm classifier associated with the EPC acquired before. The authentication process terminates until when the server either detects a valid tapping rhythm or fails to detect one after a threshold number of rounds. The RFID card and corresponding user are considered authentic in the former case and fake in the latter.

RF-Rhythm features a novel anti-eavesdropping protocol employed by the RFID reader to emit CW with random phases for extracting the user's secret tapping rhythm in both enrollment and verification phases. Our protocol can prevent a capable adversary from recovering and then replaying the legitimate user's secret rhythm from sniffed RFID signals.

V. RF-Rhythm Design Details

In this section, we illustrate the details of RF-Rhythm.

A. Feasibility Study: Tap Detection

The backscattered signal's phase information is readily available on commercial RFID readers such as Impinj R420 [4]. According to [5], it can be expressed as $\phi = (\frac{4\pi df}{c} + \phi_{\text{reader}} + \phi_{\text{card}}) \mod 2\pi$, where $2d$ is the round-trip propagation distance between the reader and card, f is the CW frequency, c is the speed of light, ϕ_{reader} denotes the phase rotation due to the reader's transmit and receive circuits, and ϕ_{card} represents the phase rotation caused by the RFID card's reflection characteristics.

Finger taps on the RFID card can change its circuit impedance [6] and also signal propagation, leading to some additional phase rotation denoted by ϕ_{tap}. So we modify the phase expression above to

$$\phi = \left(\frac{4\pi df}{c} + \phi_{\text{reader}} + \phi_{\text{card}}\right) + \phi_{\text{tap}} \mod 2\pi. \quad (1)$$

To better understand the effect of finger taps, we perform a simple experiment using an Impinj R420 reader and a

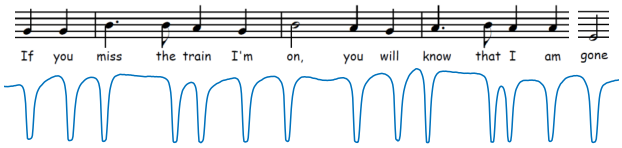


Fig. 4. Absolute phase changes induced by rhythmic taps.

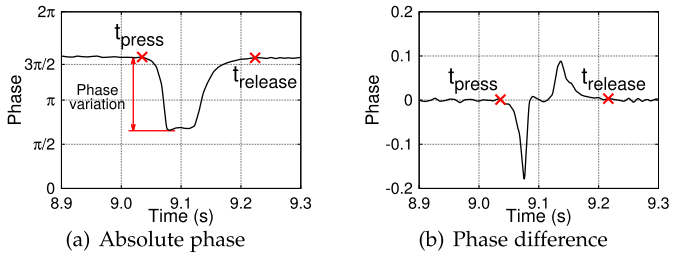


Fig. 5. Absolute and differential phase changes caused by a single tap.

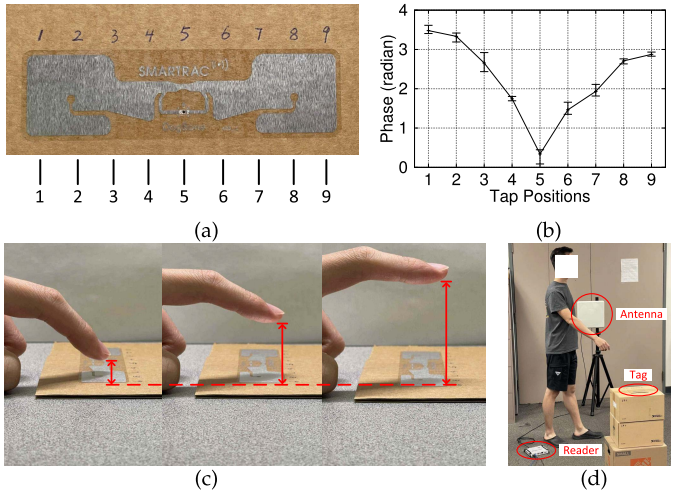


Fig. 6. (a) SMARTRAC R6 DogBone tag labeled with 9 positions; (b) box plot of phase variations of different tapping positions; (c) 3 tapping-pressure levels: low, medium, high (from left to right); (d) experiment setup for measuring the impacts of ambient environment changes.

SMARTRAC R6 DogBone tag Fig. 6(a). Fig. 4 shows the phase changes induced by rhythmic finger taps on the edge of the RFID card in accordance with the shown song segment. We also show the phase change associated with a single tap in Fig. 5. A tap event can be decomposed into a press stage and a release stage. So we use $[t_{\text{press}}, t_{\text{release}}]$ to represent a tap event in the time domain, where t_{press} and t_{release} denote the time that the phase (difference) starts to change and return to the baseline value, respectively. Fig. 5(a) and Fig. 5(b) depict the absolute phase values and the difference between adjacent phase values, respectively.

These results demonstrate the feasibility of exploring phase changes for tap detection under the simplest scenario. However, the phase changes induced by finger taps are also affected by tapping positions and pressure, finger-skin humidity, and ambient environment changes. We further conduct the following experiments to validate that finger taps can induce consistent and stable phase changes even when these factors are considered. To quantify the tap-induced phase changes, we define the phase variation as the phase difference between the baseline and the bottom of the pit with an example shown in Fig. 5(a).

1) Impact of Tapping Positions: We first measure the phase changes induced by finger taps on different tag positions. The SMARTRAC R6 DogBone tag is attached to a flat cardboard as shown in Fig. 6(a). We evenly divide the tag into 9 zones labeled on the cardboard.

The same volunteer uses his index finger to tap the tag 10 times at each position with similar tapping pressure. Fig. 6(b) is a box plot of phase variations, which illustrates the maximum, mean and minimum phase variations at each position. The results indicate that the tap event can induce obvious phase changes over all positions except position 5. When touching the metallic antenna, the impedance of the antenna is changed due to the coupling effect [6], which further causes the phase change of the backscatter signal. However, when the volunteer taps position 5, the average phase variation is only 0.47 rad and much less obvious than the second smallest value 1.46 rad at position 6, as the silicon RFID chip is not affected by the coupling effect.

2) Impact of Tapping Pressure: The actual tapping pressure is hard to quantify without special equipment. To simplify the experiment, we let the volunteer raise his index finger to 3 different height to tap the tag. A longer distance gives the finger more time to accelerate so that the tapping can have higher pressure over the tag. We associate 3 different heights with 3 tapping-pressure levels: low, medium, and high. The volunteer taps the tag at position 8 for 10 times for each pressure level. The average phase variations shown in Table I indicates that the phase changes are not affected by different tapping-pressure levels.

3) Impact of Finger-Skin Humidity: We simulate three different finger-skin humidity levels by applying different volume of water to the volunteer's finger tip. The volunteer taps position 8 for 10 times with 3 humidity levels. The results in Table I indicates that the wet finger can decrease the average phase variation by about 0.25 rad, but the phase changes are still obvious under all three humidity levels.

4) Impact of Ambient Environment Changes: We design the following experiment to measure the phase change caused by ambient environment changes. The experiment setup is shown in the Fig. 6(d). The tag is placed on the top of 3 cardboard boxes. We let the volunteer freely move between the antenna and the tag and do whatever he wants to induce as many ambient environment changes as possible. We let the volunteer perform a random movement for 10 s and repeat it 5 times. We compare the *Variance* and *Range* (i.e., maximum - minimum) of the backscattered signal phase due to such ambient environment changes with those caused by the volunteer's finger taps on position 8 in Table I. As we can see, ambient environment changes do not induce any rapid and huge phase changes in contrast to finger taps.

From the above experiment results, we can conclude that finger-tapping pressure, finger-skin humidity, and ambient environment changes have negligible impact on tap detection based on the backscattered signal's phase changes. As for

TABLE I
AVERAGE PHASE VARIATIONS UNDER DIFFERENT CASES

Tapping Pressure	Low	Medium	High
Average phase variations	2.5464	2.7059	2.6200
Finger Skin Humidity	Dry	Damp	Wet
Average phase variations	2.2518	1.9941	1.9880
	Variance	Range	
Ambient Environment Changes	0.0014	0.2147	
Tapping position 8	0.6059	2.8286	

the tapping position, as long as the users avoids tapping the chip position of the tag per some usage guideline, finger taps dominate the phase changes in the backscattered signals. These results demonstrate that the phase-change feature is reliable and robust for finger-tap detection.

B. Data Processing

We represent the reader's phase report at time t_i by $[\phi_i, f_i, t_i]$, where f_i denotes the CW frequency at t_i. According to Eq. (1), we have

$$\phi_i = \left(\frac{4\pi d f_i}{c} + \phi_{\text{reader}} + \phi_{\text{card}}\right) + \phi_{\text{tap},i} \mod 2\pi \,, \quad (2)$$

where $\phi_{\text{tap},i}$ denotes the phase shift during the ith tap. The interval $t_{i+1} - t_i$ $(i \geq 0)$ is about 4ms on the Impinj R420 reader. We temporarily assume that f_i is constant and process the raw phase data to extract more useful information for further rhythm extraction as follows.

1) Phase Difference and Unwrapping: We use the phase difference instead of the absolute phase to eliminate the approximately constant $\frac{4\pi d f_i}{c} + \phi_{\text{reader}} + \phi_{\text{card}}$ during adjacent tap events. In addition, the raw phase data are wrapped within $[0, 2\pi]$, so it is critical to perform phase unwrapping to eliminate ambiguity. Our experiments reveal that although the phase change induced by tap events are sharp, it is always bounded by π. According to this finding, the unwrapped phase difference is calculated by

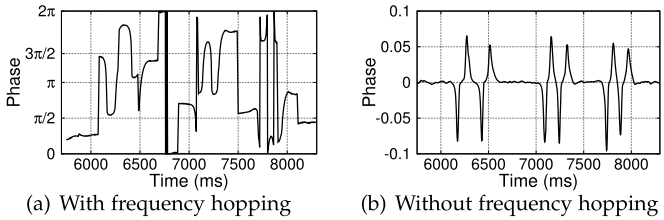
$$\begin{aligned}\Delta\phi_i &= \phi_{\text{tap},i} - \phi_{\text{tap},i-1} \\ &= \begin{cases} \phi_i - \phi_{i-1}, & |\phi_i - \phi_{i-1}| \leq \eta \\ \phi_i - \phi_{i-1} + 2\pi, & \phi_i - \phi_{i-1} < -\eta \\ \phi_i - \phi_{i-1} - 2\pi, & \phi_i - \phi_{i-1} > \eta \end{cases} \end{aligned} \quad (3)$$

Here η is an empirical value set to 3.5 in this paper.

2) Normalization: Since the sampling rate of the RFID reader is not consistent, so we further derive the time-normalized phase difference as

$$\overline{\Delta\phi_i} = \frac{\Delta\phi_i}{\Delta t_i} = \frac{\Delta\phi_i}{t_i - t_{i-1}} \,. \quad (4)$$

3) Interpolation and Filtering: We further use a linear interpolation with a factor of 4 and a 15-point average value filter to smooth the data and also mitigate the noise. We denote the final smoothed data by $\Phi = [\overline{\Delta\phi_1}, \overline{\Delta\phi_2}, \ldots, \overline{\Delta\phi_N}]$, where N denotes the total number of data points.



(a) With frequency hopping (b) Without frequency hopping

Fig. 7. Mitigating frequency hopping in phase data.

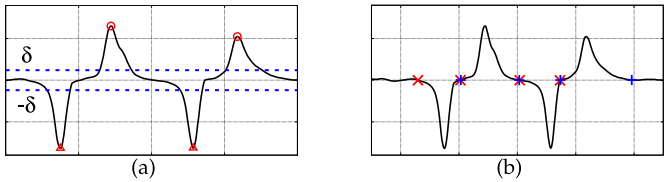


Fig. 8. An example of threshold-based tap event detection. (a) Find the local minimum and maximum points which are labeled by red triangles and circles, respectively; (b) find the $t_{\text{press,start}}, t_{\text{press,end}}$ labeled by red crosses and the $t_{\text{release,start}}, t_{\text{release,end}}$ labeled by blue pluses.

C. Mitigating Frequency Hopping

We intend RF-Rhythm to be a universal solution worldwide and thus must deal with frequency hopping mandated in many regions. For example, FCC requires that all RFID readers used in the US apply frequency hopping across 50 channels ranging from 902 MHz to 928 MHz with the dwell time on each interval no larger than 0.4 s. According to Eq. (2), such frequency hopping naturally leads to phase discontinuity in Fig. 7(a).

To see the effect of frequency hopping more clearly, assume that frequency hopping occurs at t_i $(i \geq 2)$. In the Impinj R420 reader, the frequency-hopping interval is 200 ms, while the phase-report interval is about 4 ms. So there is no frequency hopping at t_{i-2}, t_{i-1}, and t_{i+1}, i.e., $f_{i-2} = f_{i-1} \neq f_i = f_{i+1}$. The phase difference in Eq. (3) is in effect

$$\Delta\phi_i = \phi_{\text{tap},i} - \phi_{\text{tap},i-1} + \left(\frac{4\pi d f_i}{c} - \frac{4\pi d f_{i-1}}{c}\right).$$

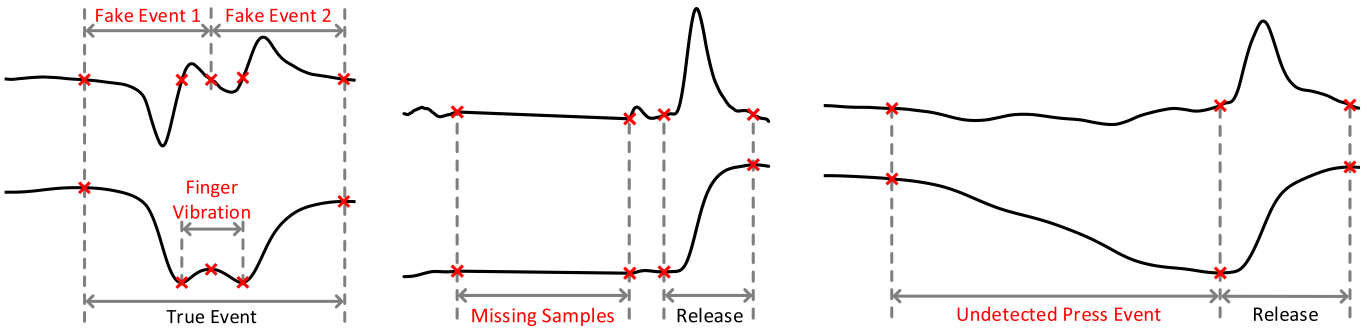
Since d is unknown and hard to estimate in practice, we cannot do a simple calibration by subtracting the term in the parenthesis from $\Delta\phi_i$. Instead, we compute the time-normalized phase difference for t_i as

$$\overline{\Delta\phi_i} = (\overline{\Delta\phi_{i+1}} + \overline{\Delta\phi_{i-1}})\frac{t_i - t_{i-1}}{t_{i+1} - t_{i-1}} \quad (5)$$

Fig. 7(b) plots the output of the Data Processing module corresponding to Fig. 7(a) after we adopt the above technique.

D. Feature Extraction

Since a tapping rhythm consists of individual taps and tap-durations, we first seek to extract individual tap events from the processed phase data $\Phi = [\overline{\Delta\phi_1}, \overline{\Delta\phi_2}, \ldots, \overline{\Delta\phi_N}]$. Recall that each tap event can be represented by $[t_{\text{press}}, t_{\text{release}}]$. We draw three observations from Fig. 5(b) obtained from preliminary experiments. First, the start and end of a tap event correspond to the phase difference beginning to deviate from and return



(a) Case 1: Finger vibration. (b) Case 2: Missing samples. (c) Case 3: Slow press or release.

Fig. 9. Three cases in which the threshold-based detector does not work. In all three figures, the top curves represent the phase difference $\overline{\Delta\phi_i}$, and the bottom curves are the absolute phase ϕ_i.

to the zero baseline, respectively. Second, the phase difference first decreases from and then returns to the zero baseline when the user finger goes from just touching to fully press on the RFID card, leading to a local minimum. Finally, the phase difference first increases from and then returns to the zero baseline when the user finger goes from decreasing the pressure on to completely leaving the RFID card, resulting in a local maximum. The later two observations are both because the card impedance gradually change with the finger pressure on the card during a tap event.

Armed with these observations, we use the following empirical process.

1) Find all the local maximums above δ and minimums below $-\delta$ in Φ, which are the triangle marks in Fig. 8(a). The threshold δ can be obtained empirically through experiments.
2) Pair each local minimum with the immediate local maximum (if any) such that there are no other local minimums or maximums in between. We require the user's tapping rhythm to be sufficiently long such that $M \gg 2$ local minimum-maximum pairs can be located in Φ, each associated with a unique tap event.
3) Find the starting and end points—$t_{\text{press,start}}$, $t_{\text{press,end}}$, $t_{\text{release,start}}$, $t_{\text{release,end}}$—of the press and release events, which are the red crosses and blue pluses shown in Fig. 8(b), respectively. $t_{\text{press,start}}$ and $t_{\text{press,end}}$ denote the first data points smaller than the zero baseline before and after the local minimum, respectively, while $t_{\text{release,start}}$ and $t_{\text{release,end}}$ are the first data points larger than the zero baseline before and after the local maximum, respectively. Furthermore, $t_{\text{press,start}}$ and $t_{\text{release,end}}$ are selected to represent the tap event, which are simplified to be t_{press} and t_{release} hereafter.

Finally, we obtain an M-tap event sequence as

$$\mathbb{V} = \begin{bmatrix} t_{\text{press},1} & t_{\text{press},2} & \dots & t_{\text{press},M} \\ t_{\text{release},1} & t_{\text{release},2} & \dots & t_{\text{release},M} \end{bmatrix}, \tag{6}$$

from which we can derive a feature vector $\mathbb{F} = [F_1, \dots, F_{M-1}]$, where $F_i = t_{\text{press},i+1} - t_{\text{release},i}$.

The aforementioned threshold-based empirical process can extract most tap events in the dataset, but there are some exceptions which may cause false or missed detections

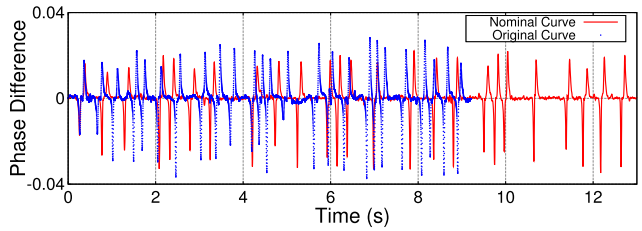
1) Case 1: Finger Vibration: Even a tiny vibration of the user's finger tip can induce a large phase change in the backscattered signals, which may be similar to that associated with a real tap event and thus make the threshold-based detector output a fake tap event. Fig. 9(a) shows an example. When the user presses the RFID card, the phase of the backscattered signal decreases. The user's finger stays on the tag for a while after touching the tag, during which the phase is stable. Then the user releases his/her finger, so the phase returns to the previous value. When comparing Fig. 9(a) with Fig. 5(a) which corresponds to a standard tap event, we can see a bump at the bottom in Fig. 9(a) in contrast to the flat bottom in Fig. 5(a). That bump is caused by the finger vibration. The threshold-based detector detects two continuous tap events as the top curve in Fig. 9(a). This kind of false detections cannot be eliminated by simply raising the threshold, as the phase changes induced by different users' finger movement may vary a lot.

E. Case 2: Missing Samples

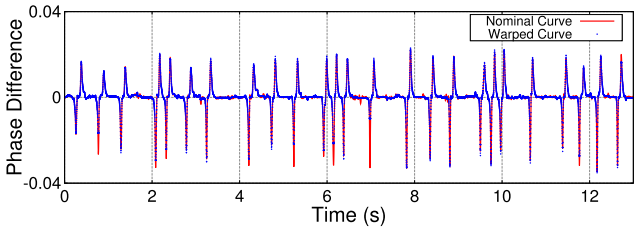
The RFID reader cannot maintain a consistent query rate. In particular, the reader may occasionally stop querying the tag for a few hundred milliseconds, during which some backscattered signals relating to true tap events are lost. The incomplete signals may cause missed tap-event detections. As exemplified in Fig. 9(b), only a release event is detected, while the samples ahead of the release event are missed. This case can be easily detected because all the phase samples have corresponding timestamps. More specifically, if the difference of timestamps between two samples are much larger than the sampling period, there are most likely missed samples between them.

F. Case 3: Slow Press/Release

Some users may press or release their fingers too slowly. If we compare Fig. 9(c) with Fig. 5(a), the phase caused by a slow press in Fig. 9(a) decreases much more slowly. Therefore, the corresponding phase difference $\overline{\Delta\phi_i}$ stays low for a long while and does not exceed the threshold. We have the similar observation for a slow release. Slow finger presses and releases both lead to missed tap-event detections.



(a) Nominal curve and original curve.



(b) Nominal curve and warped curve.

Fig. 10. Eliminate rate variations via DTW.

To solve the above issues, an absolute phase-based correction mechanism is developed to mitigate the drawbacks of the threshold-based detector. In Section V-C, we use the phase difference instead of the absolute phase to mitigate the phase discontinuity issues caused by frequency hopping. The absolute phase ϕ_i can be easily reversed from $\overline{\Delta\phi_i}$ by summing up. The correction process is as follows.

1) To prevent the fake events from been detected, the detector checks the amount of the absolute phase change during finger press or release. If $|\phi_{t_{\text{press,start}}} - \phi_{t_{\text{press,end}}}| < \delta_I$ or $|\phi_{t_{\text{release,start}}} - \phi_{t_{\text{release,end}}}| < \delta_I$, the detected press (release) events are considered fake events and then ignored. The threshold δ_I can be obtained empirically through experiments.
2) Since cases 2 and 3 have the similar phase difference curves and both lead to missed tap-event detections, we use the same strategy to correct the detection results. If there is any unpaired press or release event that satisfy $|\phi_{t_{\text{press,start}}} - \phi_{t_{\text{press,end}}}| \geq \delta_I$ or $|\phi_{t_{\text{release,start}}} - \phi_{t_{\text{release,end}}}| \geq \delta_I$, missed detections are considered happening. We empirically make up the missed tap event by $t_{\text{press, start}} = t_{\text{release, start}} - T_{\text{release}}$ or $t_{\text{release, end}} = t_{\text{press, end}} - T_{\text{press}}$, where T_{release} and T_{press} denote the duration of the detected but unpaired release or press event, respectively.

G. Rhythm Classification

The backend server builds a rhythm classifier during the enrollment phase. To do so, it instructs the user to perform rhythmic taps in accordance with his/her secret song segment multiple times. The resulting phase-difference vectors may vary due to slight tapping-rate variations. So we apply Dynamic Time Warping (DTW) [7] to align all the phase-difference vectors to that of the first acquired tapping rhythm. Fig. 10(a) shows two examples for the same rhythm performed by the same volunteer with different tapping rates. We use DTW to warp the original curve to the nominal curve. The results in Fig. 10(b) demonstrate the efficacy of DTW on handling the tapping-rate variations. Then we obtain a feature vector from each aligned phase-difference vector and pad zeros in the end (if needed) to make all the feature vectors have the same length. Finally, we use the resulting feature vectors to train a rhythm classifier based an any established machine learning technique. We compare the performance of one-vs-all linear Support Vector Machine (SVM), Neural Networks (NN), and Convolutional Neural Networks (CNN) in Section VII. During each authentication session, the server explores the same processes to extract a tapping rhythm and then test it with the rhythm classifier.

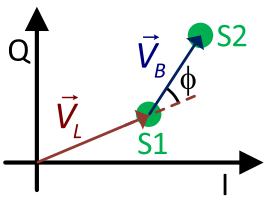


Fig. 11. Complex demodulated signals received by the reader.

VI. Anti-Eavesdropping Via Phase Hopping

In this section, we present a phase-hopping technique to prevent a capable adversary from acquiring the legitimate tapping rhythm from sniffed RFID signals. Below we first illustrate the rhythm-eavesdropping attack, followed by the motivation for using phase hopping as a defense. Then we detail the protocol design and analyze its security.

A. Rhythm-Eavesdropping Attack

We first explain the principle with which the RFID reader extracts the signals backscattered by the RFID card. As shown in Fig. 1, there are two possible voltage levels in FM0 symbols. The card only backscatters when transmitting high-voltage pulses. Consider the query protocol in Fig. 2. The symbols received by the reader between its two consecutive commands (e.g., Query and ACK) can be classified into two states (S1 and S2). The symbols in S1 contain only constant CW, while those in S2 are the superposition of CW and backscattered signals. For simplicity, we represent the symbols in S1 and S2 by two single points in the complex I-Q plane in Fig. 11, corresponding to vector $\vec{V_L}$ and $\vec{V_B}$, respectively. The phase of backscattered signals can be derived as [8]

$$\phi = \arccos\left(\frac{\vec{V_B} \cdot \vec{V_L}}{\left|\vec{V_B}\right| \left|\vec{V_B}\right|}\right). \tag{7}$$

The phase reports from the reader correspond to the samples of ϕ above. As said, the phase-sampling frequency in the Impinj R420 reader is about 4 ms.

To launch the rhythm-eavesdropping attack, the adversary can just passively sniff the reader-card communications with its own RFID reader or a software-defined radio. After classifying sniffed symbols into S1 and S2, it uses the same process above to extract ϕ. Next, it explores the workflow in Section V to acquire the legitimate tapping rhythm. Finally, it can carefully study the tapping rhythm and reproduce it

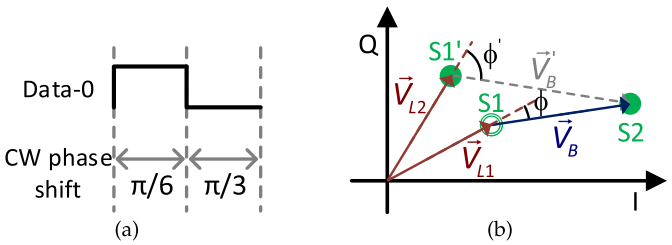


Fig. 12. Illustration of reader-phase hopping. (a) CW phase shift for data-0 symbol. (b) Constellation diagram of data-0 symbol with phase hopping.

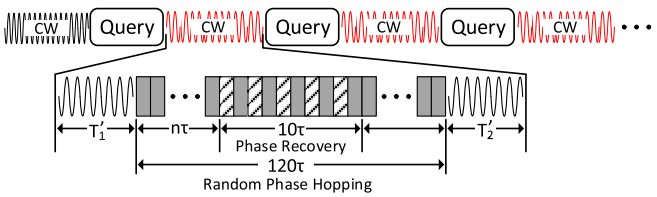


Fig. 13. Timing diagram of phase hopping.

by hand or even through a programmable robotic arm on the lost/stolen/cloned RFID card. Since this attack directly exploits physical-layer RFID signals, it cannot be thwarted by encrypting protocol messages at the application layer.

B. Phase Hopping to Mitigate Rhythm Eavesdropping

We propose to let the RFID reader emit CW with random phases to counteract the rhythm-eavesdropping attack. The objective is to prevent the adversary from obtaining matching symbols in states S1 and S2, so it cannot derive the correct phases of backscattered signals as in Fig. 11.

Fig. 12 explains the intuition of our defense. Assume that the RFID card is backscattering a data-0 symbol. As said above, the card only backscatters the high-voltage part. As shown in Fig. 12(a), we let the reader set the CW phases to $\pi/6$ and $\pi/3$ during backscattering and non-backscattering, respectively. The adversary again tries to cluster sniffed symbols into states S1 and S2. Due to phase hopping, the S1 symbols that correspond to non-backscattering has a phase offset of $\pi/3$, labeled by S1$'$ in Fig. 12(b). The true S1 symbol matching the S2 symbol, however, should have a phase offset of $\pi/6$, labeled by S1 in Fig. 12(b). Since the adversary does not know the true CW phase during backscattering, it can only use the symbols in S1$'$ and S2 to derive a wrong phase ϕ'. But the reader knows the true CW phase or S1 symbol and can thus derive the correct phase ϕ.

C. Protocol Design

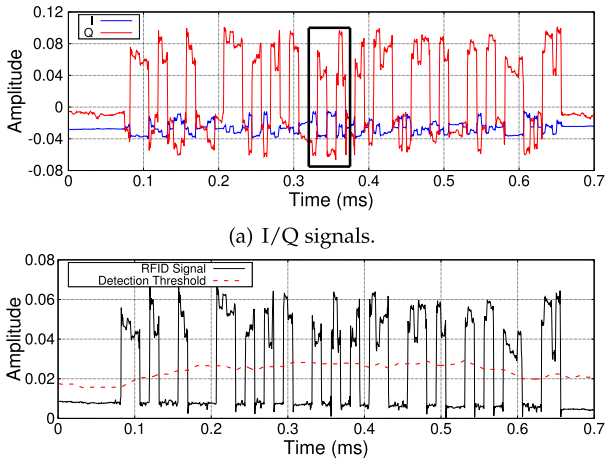
It is very challenging to properly implement the phase-hopping idea above. In particular, our example in Fig. 12 assumes perfect reader-tag synchronization such that the reader knows exactly when backscattering occurs and thus when to change the CW phase. This assumption is impossible to hold in practice. Therefore, the adversary may still be able to obtain matching symbols in S1 and S2 to derive the correct phase and eventually the legitimate tapping rhythm. A tempting solution is using a very short hopping interval, which nevertheless may negatively affect the reader's capability to recover the correct phase and thus the tapping rhythm. It is thus critical to determine the optimal phase-hopping interval to strike a balance between attack resilience and system correctness.

We illustrate our phase-hopping protocol with a simplified version of the query protocol in Fig. 2. Assume that the backend server acquires and validates the card's EPC with the protocol in Fig. 2. It then instructs the RFID reader to initiate additional query rounds to acquire the user's tapping rhythm. Each query round consists of a Query message followed by a CW period of length $T_1 + T_2 + T_{\text{RN16}}$, where T_1 and T_2 are random variables mentioned in Section II. In the original RFID protocol, the CW phase is constant. Our goal now is to determine when phase hopping should start/stop and how often it should be in each CW period.

The begin and end of phase hopping depend on T_1. According to Section II, T_1 is in $[238\ \mu\text{s}, 262\ \mu\text{s}]$ with the nominal value equal to $250\ \mu$s. We also measure the actual distribution of T_1 over 5,639 card replies. Since 98.92% of T_1 are between $244\ \mu$s to $247\ \mu$s, it is safe to conclude that if the phase-hopping duration covers $[244\ \mu\text{s}, 247\ \mu\text{s} + T_{\text{RN16}}]$, almost all the backscattered signals associated with RN16 can be covered.

The next challenge is to determine the hopping interval τ, which should be as short as possible for high attack resilience. The minimum τ is hardware-specific and empirically set to $\tau = \frac{T_{\text{pri}}}{5} = 5\ \mu$s in our USRP implementation, where $T_{\text{pri}} = 1/\text{BLF} = 25\ \mu$s denotes the FM0 symbol duration introduced in Section II. Ideally speaking, each CW phase value leads to a unique pair of S1 and S2 symbols as shown in Fig. 11. In practice, we can only obtain two clusters of symbols associated with S1 and S2, respectively, which are referred to the S1 and S2 clusters for convenience. The RFID reader needs to obtain the matching S1 and S2 clusters for at least one random CW phase to recover the correct phase for the backscattered RN16. Our experiments reveal that strictly sticking to τ would induce too many randomly distributed symbols in the I-Q plane, which make it very difficult for the reader to do proper symbol clustering.

We tackle the above issue by introducing a short *phase-discovery* period lasting γ that must satisfy two requirements. First, it starts from a random hopping interval hard to predict by the adversary. Second, it covers at least one phase inversion in the FM0 symbols of the RN16 message. An RN16 message comprises a 6-bit preamble, a 16-bit random number, and one dummy bit. According to FM0 encoding in Fig. 1, there is a phase inversion at every symbol boundary and also one in the middle of each data-0 symbol, but the FM0 preamble contains a phase-inversion violation at the fifth symbol labeled "v". So the longest time that the RFID card does not invert the signal phase is $1.5\ T_{\text{pri}}$. Since the reader does not know when backscattering (i.e., the RN16 transmission) starts, we set $\gamma = 2T_{\text{pri}} = 10\tau$ to satisfy both requirements above. The phase-discovery period obviously consists of 10 hopping intervals. In addition, the reader uses the same CW phase in the



(a) I/Q signals.

(b) Amplitude of I/Q signals.

Fig. 14. (a) The I/Q signals received by the RFID reader, which contains the RN16 response from the RFID tag; (b) The amplitude ($\sqrt{I^2+Q^2}$) of the I/Q signals in Fig. 14(a) and the detection threshold used for detecting the positive and negative edges.

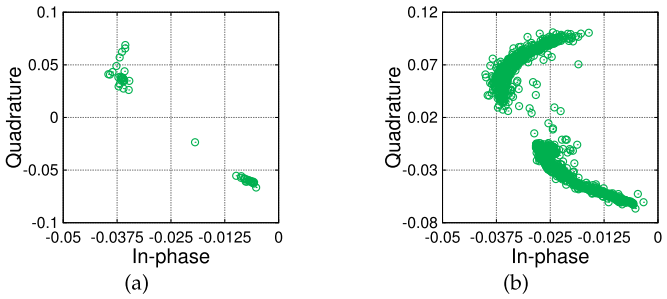


Fig. 15. (a) Extracted samples for phase recovery; (b) The adversary's sniffed symbols for Fig. 14(a).

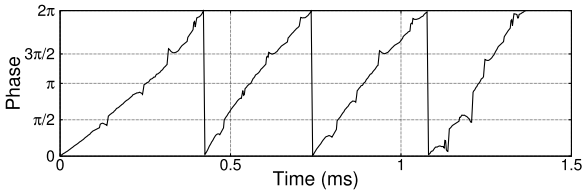


Fig. 16. Phase recovered by the reader.

odd-numbered hopping intervals and performs random phase hopping in the rest intervals of the phase-discovery period.

Now we explain the protocol details with the timing diagram in Fig. 13. After sending the Query message, the RFID reader starts the phase-hopping duration at T_1' which is divided into short hopping intervals of $\tau = 5$ μs long. We require the phase-hopping duration to at least cover the range $[244\ \mu\text{s}, 247\ \mu\text{s} + T_{\text{RN16}}]$, where $T_{\text{RN16}} = 575$ μs [3]. So we set $T_1' = 240$ μs and the phase-hopping duration to 600 μs long which corresponds to 120 hopping intervals. For each rhythm-query round, the reader determines 24 CW phase values

$$\Theta = [\theta_{\text{init}}, \theta_{\text{init}} + 1, \theta_{\text{init}} + 2, \ldots, \theta_{\text{init}} + 23], \tag{8}$$

where θ_{init} is a random integer in $[0, 360)$. Assume that the phase-recovery period starts at $T_1' + n\tau$, where $n \in [0, 110]$ is randomly chosen by the reader because the phase-hopping duration lasts 120 hopping intervals. In addition, the reader randomly selects $\theta_{\text{reserve}} \in \Theta$ and uses it for the five odd-numbered hopping intervals (represented by lined blocks) in the phase-recovery period. Finally, the reader performs random phase hopping across the remaining 23 phase values in the rest 115 hopping intervals (represented by gray blocks) such that each phase value in Θ (including θ_{reserve}) is used exactly five times in each rhythm-query round.

Fig. 14(a) gives an example for the efficacy of our protocol, which is based on our prototyping implementation on a USRP 2954R device. The phase-hopping duration is from 0.1 ms to 0.7 ms, and the reader's received signals in the phase-recovery period are enclosed by the black rectangle. Since the reader knows exactly when the phase-recovery period starts, it can precisely locate the symbols associated with the constant phase θ_{reserve}. As shown in Fig. 15(a), the reader can easily cluster these symbols into states S1 and S2 whereby to extract the correct phase of backscattered signals. To highlight the correctness of our protocol, we show complete phase plots obtained by the reader in Fig. 16 with our phase-hopping protocol, which match well with those on a traditional RFID reader without phase hopping [4]. We also demonstrate the decoding correctness of the proposed protocol using Fig. 14(b) which plots the amplitude of the I/Q signals in Fig. 14(a). The red dash line is the threshold used for detecting the positive and negative edges of the backscatter signal. The detection threshold is the average amplitude with window length equal to 500, according to Nikos' implementation [9]. Although the amplitude is slightly distorted due to the I/Q imbalance, the positive and negative edges are still perfectly separated by the detection threshold, and the decoding procedure is not affected by phase hopping. The proposed random phase-hopping protocol only shifts the phase of the continuous wave. Since phase shifting theoretically does not change the signal amplitude, the amplitude-based decoder is not affected. In the experiment, we also observe that the RFID reader with random phase hopping has the similar reading rate to the original reader. In contrast, the adversary does not know when the phase-recover period starts. So it has to exploit all the sniffed symbols for phase recovery, which is almost impossible as shown in Fig. 15(b).

D. Resilience to Advanced Eavesdropping

The proposed phase-hopping protocol can thwart basic eavesdropping attacks in which the adversary has only one sniffer that overhears the superposition of the backscattered signal and CW with random phase hopping. Now we analyze its resilience to advanced eavesdropping attacks in which the adversary has an additional sniffer at distance d_1 from the reader and d_2 from the card. The adversary can also vary d_1 and d_2 arbitrarily. Theoretically speaking, the second sniffer also receives the superposition of the backscattered signal and CW with random phase hopping. Assume that the adversary can make d_2 large enough such that the backscattered signal is attenuated too much to detect, while keeping d_1 sufficiently

small such that the CW signal is still strong enough. The signal overheard by the second sniffer thus corresponds to CW alone. The adversary can then derive the phase-hopping sequence and correlate it with the signals obtained by the first sniffer to recover the phase information of backscattered signals.

To analyze the feasibility of the advanced eavesdropping attack above, we assume the free-space path loss (FSPL) model for RFID signal propagation, FSPL $= (\frac{4\pi d}{\lambda})^2$, where d is the distance between antennas, and λ is the CW wavelength. Assume that the RFID card is at distance d_0 from the reader. The power of the reader's signal at the card is $P_{\text{card}} = P_t G_t (\frac{\lambda}{4\pi d_0})^2$, where P_t is the reader's transmission power, d_0 is the distance between reader, and G_t is the reader's antenna gain. According to [10], the EIRP (Equivalent Isotropically Radiated Power) of passive RFID cards is

$$\text{EIRP}_{\text{card}} = P_{\text{reader}} \frac{4\pi\sigma}{\lambda^2} = P_t G_t \frac{\sigma}{4\pi d_0^2}\ , \tag{9}$$

where σ denotes the tag's radar cross section (RCS) [10]. σ mainly depends on the impedance of card antenna and chip and depicts the backscattered power strength tag.

The second sniffer receives the superposition of CW and the backscattered signal. The signal strength for CW can be expressed by

$$P_{\text{CW},d_1} = \frac{P_t G_t G_r}{\text{FSPL}_{\text{reader}}} = P_t G_t G_r (\frac{\lambda}{4\pi d_1})^2\ ,$$

where G_r denotes the second sniffer's antenna gain. Similarly, the signal strength for the backscattered signal can be expressed by

$$P_{\text{BS},d_2} = \frac{\text{EIRP}_{\text{card}} \text{G}_\text{r}}{\text{FSPL}_{\text{reader}}} = P_t G_t G_r \frac{\sigma\lambda^2}{(4\pi)^3 d_0^2 d_2^2}.$$

Let τ_{rx} and τ_{dec} denote the minimum signal strengths that the sniffer can detect and decode RFID signals, respectively. The advanced eavesdropping attack works if and only if $P_{\text{CW},d_1} \geq \tau_{\text{dec}}$ and $P_{\text{BS},d_2} \leq \tau_{\text{rx}}$ can simultaneously hold. It is equivalent for the adversary to find d_1 and d_2 that satisfy

$$d_1 \leq \sqrt{\frac{P_t G_t G_r}{\tau_{\text{dec}}} (\frac{\lambda}{4\pi})^2}$$

and

$$d_2 \geq \sqrt{\frac{P_t G_t G_r}{\tau_{\text{rx}}} \frac{\sigma\lambda^2}{(4\pi)^3 d_0^2}}\ .$$

The above requirement corresponds a *vulnerable region* outside the circle centered at the card with radius d_2 and inside the circle centered at the reader with radius d_1. In Section VII, we experimentally show that the vulnerable region can be very difficult or infeasible to find in practice.

VII. Evaluation

A. Experimental Setup

We use two Impinj R420 readers (GX21M and USA2M1 models) with Laird S9028 antenna. GX21M does not use frequency hopping, while USA2M1 does. The data from USA2M1 are calibrated with the method in Section V-C and then combined with the data from GX21M. We use three types of RFID tags, including SMARTRAC R6 DogBone, Impinj E51, and Alien 9640. In addition, we prototype the phase-hopping protocol on a USRP 2954R and also used an R&S FSVR7 real-time spectrum analyzer for signal analysis.

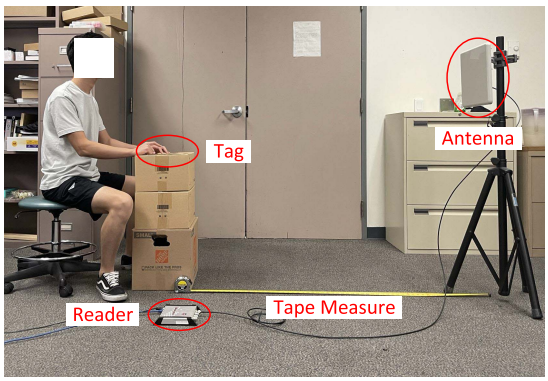


Fig. 17. Experiment setup.

We compare the classification performance of SVM, NN, and CNN. The comparison is based on the SVM toolbox in Matlab and the NN and CNN implementations in PyTorch. We use a fully connected NN with one hidden layer and 256 perceptions. In addition, the CNN we use has two 1D convolutional layers and a kernel size of 2. All the training and classification procedures are performed on a Ubuntu desktop with i7-8700k CPU and 16 GB RAM.

We recruite 19 volunteers from China and US who are either undergraduate or graduate students. The data-collection setup is shown in Fig. 17. Each volunteer taps a random RFID tag 40 times according to his/her self-chosen rhythm. The volunteers are asked not to tap the chip position of the tag as described in the previous section. Most chosen rhythms last 6 s to 12 s with the average and variance equal to 9.61 s and 5.86 s, respectively. The RFID reader-tag distance is always about 40 inches. We collect 760 tapping rhythm samples in total.

B. Resilience to Brute-Force Attacks

We first evaluate the performance of RF-Rhythm under the brute-force attack. For this evaluation, we randomly choose K rhythm samples from each volunteer to train one-vs-all classifiers. The remaining rhythm samples are treated as the testing set. We use a modified K-fold cross validation method to eliminate the potential dataset bias. K denotes the number of training samples and can be used to represent the duration of user enrollment phase. A smaller K means that a legitimate user can input his/her tapping rhythm fewer times in the enrollment phase, leading to shorter enrollment time and higher usability, and vice versa. When the classifier of each volunteer is tested against the data samples of all the other 19 volunteers, it amounts to launching a brute-forth attack on RF-Rhythm.

Fig. 18 shows the testing accuracy (ACC), false positive rate (FPR) and false negative rate (FNR) of SVM, NN and CNN classifiers. Overall, RF-Rhythm can admit legitimate users and reject random impostors with overwhelming probability under all three classifiers. The ACCs of SVM, NN and

TABLE II
CLASSIFICATION ACCURACY FOR ENROLLMENT-AUTHENTICATION LOCATION VARIATIONS

	SVM				NN				CNN			
	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4
E1	1.0	0.925	0.8	0.9	1.0	0.925	0.8	0.9	1.0	0.975	0.9	0.95
E2	1.0	1.0	1.0	0.925	1.0	1.0	1.0	0.95	1.0	1.0	1.0	0.95
E3	0.95	1.0	0.92	0.925	1.0	0.95	1.0	0.95	1.0	1.0	1.0	0.95
E4	0.95	1.0	1.0	0.92	1.0	1.0	0.95	0.95	1.0	1.0	1.0	1.0

CNN are 95.91%, 95.04%, 95.40% at $K = 4$, and 98.39%, 98.21%, 99.11% at $K = 20$, which are similar. The FPRs for three classifiers are 0.24%, 0.38%, and 0.35%, which means all three models are resilient to the brute-force attack, and SVM shows the better overall performance. Especially, the performance of CNN fluctuates when K increases, which contradicts the intuition that its performance should improve. This is because the training dataset is not large enough to feed the neural network.

Since the same user may perform enrollment and authentication at a different distance from the RFID reader, we also evaluate the impact of this distance factor. In this experiment, we place an RFID card at 20, 40, 80, and 120 inches from the RFID reader and let a random volunteer input his tapping rhythm 40 times at each testing location. Then we train a classifier for the volunteer at each location by using his rhythm samples collected there and the samples of all the other 18 volunteers as the training data. Finally, we test each obtained classifier against the volunteer's rhythm samples collected at the same and different locations. Table II shows the classification accuracy for this evaluation, where E1&T1, E2&T2, E3&T3, and E4&T4 denote the enrollment and testing locations at 20, 40, 80, and 120 inches, respectively. If the enrollment and testing locations are the same, we randomly divide the volunteer's samples at that location into 2 parts for training and testing, respectively; otherwise, all the 40 samples are used for training in each enrollment location. The results represent the average of 10 runs. It is clear that RF-Rhythm is robust to enrollment-authentication location variations.

C. Robustness to Elapsed Time

Now we evaluate the robustness of RF-Rhythm to the elapsed time after the user enrolls. For this experiment, we recruit 10 new volunteers from US. Each volunteer taps a random tag 40 times during the enrollment phase. We combine the rhythms collected during the enrollment phase and all the previous collected rhythms as the training dataset. We let each volunteer come back after 1 hour, 1 day, 3 days, and 10 days, and recollect his/her rhythms 5 times each time as the testing dataset. The training and testing processes for each classifier are repeated 10 times. The average accuracy and recall rates are shown in Table III. All three classifiers perform well over 4 subsets of testing data, especially the CNN because of the increased training dataset size. The results demonstrate the robustness of RF-Rhythm over a long period, in which the user can easily reproduce his/her rhythmic taps according to a self-selected favorite song segment.

TABLE III
ACCURACY AND RECALL RATES FOR DIFFERENT ELAPSED-TIME SETTINGS

	1 hour	1 day	3 days	10 days
SVM	0.998/1.00	1.00/1.00	0.995/0.920	0.998/0.940
NN	0.990/0.948	0.996/0.980	0.991/0.950	0.996/0.976
CNN	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00

TABLE IV
REJECTION RATE (%) FOR VISUAL EAVESDROPPERS

	SVM	NN	CNN
one observation, one try	94.28	98.58	95.72
arbitrary observations, 4 tries	92.14	91.78	92.86

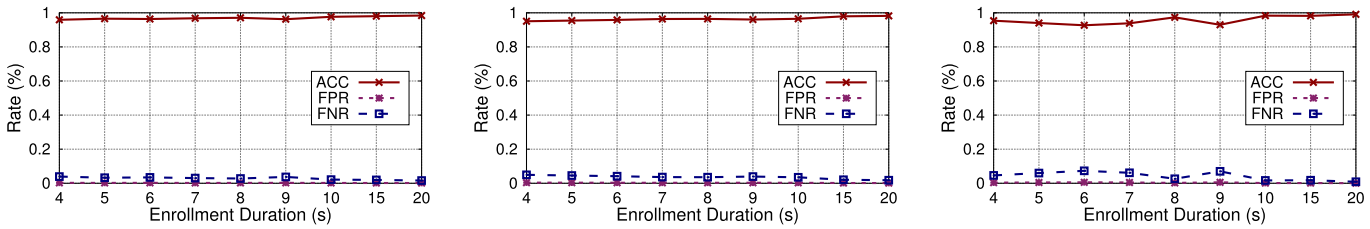
D. Resilience to Visual Eavesdropping

We also evaluate the resilience of RF-Rhythm to visual eavesdropping. In this evaluation, we use a high-definition smartphone to video-record each volunteer's entire rhythm-tapping process as shown in Fig. 19. Then we recruit five volunteers that act as attackers to watch all the 19 videos and then emulate the tapping rhythms they observe. We consider two scenarios. First, each attacker has a one-time watching of each video and then tries to perform the observed rhythm once. This scenario emulates the shoulder-surfing attack. Second, each attacker can watch each video as many times as they want and then performs each perceived rhythm four times. This scenario emulates the video-taping attack via a spy camera. We totally collect 475 attack samples. Then we build a classifier for each of the 19 volunteers with all the aforementioned 760 rhythm samples as the training data. Finally, we test each attack sample with the corresponding volunteer's classifer.

Table IV shows the rejection rate for visual eavesdroppers, which represents the average of 10 runs. We can see that RF-Rhythm has strong resilience to visual eavesdroppers under all three classification methods. In addition, a visual eavesdropper can intuitively achieve a higher success rate with more observations and authentication attempts. RF-Rhythm can rate-limit unsuccessful authentication attempts to provide a stronger defense.

E. Resilience to Basic Rhythm Eavesdropping

Next we examine the efficacy of our phase-hopping protocol to a rhythm eavesdropper with a single sniffer. As shown in Fig. 15(b), the adversary can roughly cluster sniffed symbols into states S1 and S2, respectively. But it cannot precisely find the matching S1 and S2 symbols of the same CW phase. We assume that the adversary is very powerful and knows how our phase-hopping protocol works. Since the CW phase in each query round takes random values in $\Theta = [\theta_{\text{init}}, \theta_{\text{init}} + 1, \theta_{\text{init}} + 2, \ldots, \theta_{\text{init}} + 23]$, we assume that the adversary can estimate a candidate phase vector Θ' from sniffed S1 symbols. Due to noise, interference, and processing errors, Θ' may overlap but is usually much larger than Θ. The symbols in Θ' can be much fewer than sniffed S1 symbols. Then the adversary picks an arbitrary sniffed S2 symbol,



(a) Performance of SVM (b) Performance of NN (c) Performance of CNN

Fig. 18. The accuracy (ACC), false positive rate (FPR) and false negative rate (FNR) of SVM, NN and CNN.

Fig. 19. Recorded rhythm tapping process.

denoted by s_2, and uses each S1 symbol in Θ' as a candidate matching symbol for s_2 to derive a candidate phase of the backscattered RN16. The probability of a correct guess is simply $1/|\Theta'|$. Each rhythm-query round is about 2.179 ms long, and the average tapping-rhythm duration is 9.61 s in our experiments. So we need about 4,410 rounds to cover and detect an average tapping rhythm. The probability that the adversary can recover the correct tapping rhythm from sniffed signals can be estimated by $\tilde{P} = (1/|\Theta'|)^n$. For example, if $|\Theta'| = 24|48|72$, the adversary can succeed with negligible probability. Therefore, our phase-hopping protocol is highly effective against the basic rhythm-eavesdropping attack.

F. Resilience to Advanced Rhythm Eavesdropping

We also evaluate the resilience of RF-Rhythm to advanced rhythm-eavesdropping attacks in which the adversary has two sniffers at strategic locations. In Section VI-D, we identify a theoretical vulnerable region in which this attack can succeed. In this section, we show that the vulnerable region may not be easily found by an adversary with reasonable equipment.

In this evaluation, we assume that the adversary places his second sniffer d_1 from the RFID reader and d_2 from the RFID card. For simplicity, we assume that the reader, tag, and sniffer are on the straight line. This is a reasonable assumption because most commonly used RFID antennas are directional with a relatively focused and narrow radio wave beam. We implement a EPC Gen2 RFID reader prototype [9] on an NI USRP 2954R and assume that the adversary has a similar sniffer device. We also use an R&S FSVR7 real-time spectrum analyzer for signal measurements. Recall that τ_{rx} and τ_{dec} denote the minimum signal strengths that the sniffer can detect and decode RFID signals, respectively. According to our measurements, $\tau_{\text{rx}} = -81.21$ dB m and $\tau_{\text{dec}} = -55.98$ dB m.

To emulate the attack, we vary the RFID card-reader distance d_0 from 10 to 40, 80, and 120 inches. For each d_0 value, we measure the CW signal strength P_{CW,d_1} and the backscattered signal strength P_{BS,d_2} at $d_2 = 40$ inches from the RFID card, which also corresponds to $d_1 = d_0 + 40$ inches. This location is regarded as the sniffer's initial location. The results are shown in Table V. Since we assume the reader-card-sniffer line topology, P_{CW,d_1} and P_{BS,d_2} are attenuated by the same amount when d_2 and equivalently d_1 increase. According to our analysis in Section VI-D, the advanced

TABLE V

POWER MEASUREMENTS FOR ADVANCED RHYTHM EAVESDROPPING

d_0/inch	P_{CW,d_1}/dB m	P_{BS,d_2}/dB m	$P_{\text{CW},d_1} - P_{\text{BS},d_2}$/dB m
10	-3.30	-27.91	24.61
40	-7.98	-27.00	20.78
80	-10.15	-24.02	14.53
120	-14.52	-26.43	12.29

eavesdropping attack succeeds if and only if $P_{\text{CW},d_1} \geq \tau_{\text{dec}}$ and $P_{\text{BS},d_2} \leq \tau_{\text{rx}}$ can simultaneously hold. This requires $P_{\text{CW},d_1} - P_{\text{BS},d_2} \geq \tau_{\text{dec}} - \tau_{\text{rx}} = 25.23$ dB m per our measurements. This requirement cannot be satisfied according to Table V, so the advanced eavesdropping attack would fail.

It is possible that a more capable adversary with advanced equipment can successfully overhear the legitimate user's tapping rhythm. Instead of being a perfect solution, RF-Rhythm, however, just aims to enhance the security of a traditional RFID authentication system that is naturally vulnerable to lost/stolen/cloned RFID cards. In other words, RF-Rhythm significantly raises the bar for launching successful attacks on RFID authentication systems.

G. Latency

We evaluate the latency performance of RF-Rhythm based on our hardware and software configurations mentioned in Section VII-A. For simplicity, we only report the data for $K = 4$ and $K = 20$, which corresponds to requiring a user to input the chosen rhythm 4 and 20 times in the enrollment phase, respectively. The average training time with SVM|NN|CNN is 0.39 s|0.07 s|0.22 s for $K = 4$ and 4.48 s|0.28 s|0.95 s for $K = 20$, respectively; the average testing time with SVM|NN|CNN is 0.337 ms|0.067 ms|0.73 ms. The classifier training and testing time is obviously negligible for the backend server which can be much more powerful than our machine. So the enrollment and authentication time is roughly K times of and equal to the tapping-rhythm duration, respectively. Since the average rhythm length chosen by our volunteers is 9.61 s, the enrollment and authentication times is both quite acceptable in practice.

H. Usability Study

Finally, we evaluate the usability of RF-Rhythm by asking each volunteer to give a score between [0,5] (with 5 being the highest) to each of the following questions: whether RF-Rhythm is easy to use (Q1), whether self-defined rhythms are easy to memorize (Q2), whether the rhythm length is appropriate (Q3), and whether RF-Rhythm would be easier

TABLE VI
USABILITY SCORES

	Mean	Standard Deviation	Min	Median	Max
Q1	4.21	0.63	3.00	4.00	5.00
Q2	4.21	0.79	3.00	4.00	5.00
Q3	4.15	0.60	3.00	4.00	5.00
Q4	4.52	0.61	3.00	5.00	5.00

to use with more practice (Q4). According to the results in Table VI, RF-Rhythm is highly usable.

VIII. RELATED WORK

Rhythm-based authentication for mobile devices has been explored. RhyAuth [11] is a two-factor rhythm-based authentication scheme for multi-touch mobile devices. It requires a user to perform a sequence of rhythmic taps/slides on a device screen to unlock the device. In the follow-on work, Beat-PIN [12] requires a user to tap the screen of a smartwatch to unlock it. RF-Rhythm differs significantly from RhyAuth and Beat-PIN in the application context, totally different rhythm-extract techniques, adversary models, and countermeasures.

There is also significant effort on RFID security. For example, novel cryptographic RFID authentication protocols are presented in [13], [14], and [15]. Haitham [16] proposes RF-Cloak to prevent eavesdropping attacks by randomizing the modulation and channel. Selective jamming is proposed in [17] to prevent unauthorized inquiries to RFID tags. Zanetti and Danev [18] explore the time interval error, average baseband power and spectral features to fingerprint RFID tags. TapPrint [19] uses the phase of backscattered signals combined with the geometric relationship to fingerprint RFID tags. Hu-Fu [20] uses the inductive coupling of two tags to fingerprint them. RF-Mehndi [21] identifies an RFID card and its user simultaneously by exploring the backscattered signal changes induced by the user's fingertip on a specially build passive tag array. RF-Rhythm explores COTS RFID tags and is complimentary to the above work.

The phase information of backscattered RFID signals has been explored in many applications, such as gesture recognition [22], [23], action recognition [24], [25], orientation tracking [26], mechanical features sensing [27], [28], and localization [29]. RF-Rhythm is the first work to extract a tapping rhythm from backscattered RFID signals and is orthogonal to the above work.

IX. CONCLUSION

In this paper, we proposed RF-Rhythm, a secure and usable two-factor UHF RFID authentication system which is resilient to lost/stolen/cloned RFID cards. Comprehensive prototyped experiments confirmed that RF-Rhythm has strong resilience to common attacks on RFID authentication systems and is also highly usable with short enrollment and authentication time.

REFERENCES

[1] *Two-Factor Authentication (2FA) Explained: RFID Access Control*. Accessed: 2018. [Online]. Available: https://blog.identityautomation.com/two-factor-authentication-2fa-explained-rfid-access-control

[2] *Duo Push*. Accessed: 2022. [Online]. Available: https://duo.com/product/multi-factor-authentication-mfa/duo-mobile-app

[3] *EPC UHF Gen2 Air Interface Protocol*. Accessed: 2018. [Online]. Available: https://www.gs1.org/standards/epc-rfid/uhf-air-interface-protocol

[4] (2019). *Impinj Speedway Revolution R420 UHF RFID Reader*. [Online]. Available: https://www.atlasrfidstore.com/impinj-speedway-revolution-r420-uhf-rfid-reader-4-port/

[5] (2013). *Speedway Revolution Reader Application Note: Low Level User Data Support*. [Online]. Available: https://support.impinj.com/hc/en-us/articles/202755318-Application-Note-Low-Level-User-Data-Support

[6] S. Pradhan, E. Chai, K. Sundaresan, L. Qiu, M. Khojastepour, and S. Rangarajan, "Rio: A pervasive RFID-based touch gesture interface," in *Proc. ACM Mobicom*, Snowbird, UT, USA, Oct. 2017, pp. 261–274.

[7] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a lie group," in *Proc. IEEE CVPR*, Columbus, OH, USA, Jun. 2014, pp. 588–595.

[8] C. Wang, L. Xie, W. Wang, T. Xue, and S. Lu, "Moving tag detection via physical layer analysis for large-scale RFID systems," in *Proc. IEEE INFOCOM*, San Francisco, CA, USA, Apr. 2016, pp. 1–9.

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

[10] P. V. Nikitin and K. V. S. Rao, "Antennas and propagation in UHF RFID systems," in *Proc. IEEE RFID*, Las Vegas, NV, USA, Apr. 2008, pp. 277–288.

[11] Y. Chen, J. Sun, R. Zhang, and Y. Zhang, "Your song your way: Rhythm-based two-factor authentication for multi-touch mobile devices," in *Proc. IEEE INFOCO*, Hong Kong, Apr. 2015, pp. 2686–2694.

[12] B. Hutchins, A. Reddy, W. Jin, M. Zhou, M. Li, and L. Yang, "Beat-PIN: A user authentication mechanism for wearable devices through secret beats," in *Proc. ACM ASIACCS*, Incheon, South Korea, Jun. 2018, pp. 101–115.

[13] L. Kulseng, Z. Yu, Y. Wei, and Y. Guan, "Lightweight mutual authentication and ownership transfer for RFID systems," in *Proc. IEEE INFOCOM*, Pisa, Italy, Mar. 2010, pp. 1–5.

[14] T. Li, W. Luo, Z. Mo, and S. Chen, "Privacy-preserving RFID authentication based on cryptographical encoding," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2174–2182.

[15] L. Yang, Q. Lin, C. Duan, and Z. An, "Analog on-tag hashing: Towards selective reading as hash primitives in Gen2 RFID systems," in *Proc. ACM MobiCom*, 2017, pp. 301–314.

[16] H. Hassanieh, J. Wang, D. Katabi, and T. Kohno, "Securing RFIDs by randomizing the modulation and channel," in *Proc. NSDI*, Oakland, CA, USA, May 2015, pp. 235–249.

[17] H. Ding *et al.*, "Preventing unauthorized access on passive tags," in *Proc. IEEE INFOCOM*, Honolulu, HI, USA, Apr. 2018, pp. 1115–1123.

[18] D. Zanetti, B. Danev, and S. Apkun, "Physical-layer identification of UHF RFID tags," in *Proc. ACM Mobicom*, Chicago, IL, USA, 2010, pp. 353–364.

[19] L. Yang, P. Peng, F. Dang, C. Wang, X.-Y. Li, and Y. Liu, "Anti-counterfeiting via federated RFID tags' fingerprints and geometric relationships," in *Proc. IEEE INFOCOM*, Hong Kong, Apr. 2015, pp. 1966–1974.

[20] G. Wang *et al.*, "Towards replay-resilient RFID authentication," in *Proc. ACM Mobicom*, Oct. 2018, pp. 385–399.

[21] C. Zhao *et al.*, "RF-mehndi: A fingertip profiled RF identifier," in *Proc. IEEE INFOCOM*, Paris, France, Apr. 2019, pp. 1513–1521.

[22] C. Wang *et al.*, "Multi—Touch in the air: Device-free finger tracking and gesture recognition via COTS RFID," in *Proc. IEEE INFOCOM*, Honolulu, HI, USA, Apr. 2018, pp. 1691–1699.

[23] Y. Bu *et al.*, "RF-dial: An RFID-based 2D human-computer interaction via tag array," in *Proc. IEEE INFOCOM*, Honolulu, HI, USA, Apr. 2018, pp. 837–845.

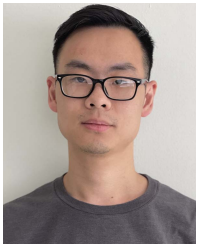
[24] C. Wang, J. Liu, Y. Chen, L. Xie, H. Liu, and S. Lu, "RF-Kinect: A wearable RFID-based approach towards 3D body movement tracking," in *Proc. ACM UBICOMP*, Singapore, Oct. 2018, pp. 1–28.

[25] H. Jin, Z. Yang, S. Kumar, and J. Hong, "Towards wearable everyday body-frame tracking using passive RFIDs," in *Proc. ACM UBICOMP*, Singapore, Oct. 2018, pp. 1–23.

[26] T. Wei and X. Zhang, "Gyro in the air: Tracking 3D orientation of batteryless Internet-of-Things," in *Proc. ACM Mobicom*, New York, NY, USA, Oct. 2016, pp. 55–68.

[27] L. Yang, Y. Li, Q. Lin, X.-Y. Li, and Y. Liu, "Making sense of mechanical vibration period with sub-millisecond accuracy using backscatter signals," in *Proc. ACM Mobicom*, New York, NY, USA, Oct. 2016, pp. 16–28.

[28] H. Jin, J. Wang, Z. Yang, S. Kumar, and J. Hong, "Wish: Towards a wireless shape-aware world using passive RFIDs," in *Proc. ACM MobiSys*, Munich, Germany, Jun. 2018, pp. 428–441.
[29] Y. Ma, N. Selby, and F. Adib, "Minding the billions: Ultra-wideband localization for deployed RFID tags," in *Proc. ACM Mobicom*, Snowbird, UT, USA, Oct. 2017, pp. 248–260.

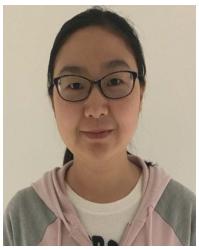


Jiawei Li (Student Member, IEEE) received the B.E. degree in telecommunication engineering from the Nanjing University of Posts and Telecommunications in 2017. He is currently pursuing the Ph.D. degree in computer engineering with Arizona State University. His research interests are security and privacy issues in wireless networks and wireless sensing.

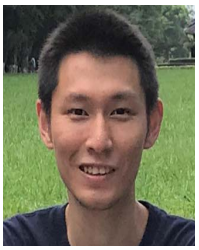
Chuyu Wang received the Ph.D. degree in computer science from Nanjing University, China, in 2018. He is currently an Assistant Professor with the Department of Computer Science and Technology, Nanjing University. His research interests include RFID systems, software-defined radio, activity sensing, and indoor localization.

Ang Li received the B.E. degree in network engineering from Guangxi University, China, in 2010, and the M.S. degree in computer science from Beihang University, China, in 2014. He is currently pursuing the Ph.D. degree in computer engineering with Arizona State University. His research interests are about security and privacy in social networks, machine learning, wireless networks, and mobile computing.

Dianqi Han (Member, IEEE) received the B.S. degree in information security from the University of Science and Technology of China, the M.S. degree in electrical and computer engineering from the University of California, Davis, and the Ph.D. degree in computer engineering from Arizona State University. He is currently an Assistant Professor with the Department of Computer Science and Engineering, The University of Texas at Arlington. His research interests include security and privacy in networked systems.



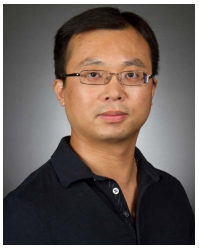
Yan Zhang (Member, IEEE) received the Ph.D. degree in computer engineering from Arizona State University. She is currently an Assistant Professor with the Electrical and Computer Engineering Department, The University of Akron. Her research interests are cybersecurity and privacy issues in mobile and networked systems, including the Internet of Things, AI/ML-powered wireless and mobile systems, mobile sensing, and mobile crowdsourcing.



Jinhang Zuo received the B.E. degree in communication engineering from the Nanjing University of Posts and Telecommunications. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Carnegie Mellon University. His research interests include online learning, social networks, and resource allocation.

Rui Zhang (Member, IEEE) received the B.E. degree in communication engineering and the M.E. degree in communication and information system from the Huazhong University of Science and Technology, China, in 2001 and 2005, respectively, and the Ph.D. degree in electrical engineering from Arizona State University in 2013. He is currently an Associate Professor with the Department of Computer and Information Sciences, University of Delaware. He was an Assistant Professor with the Department of Electrical Engineering, University of Hawaii, from 2013 to 2016. His primary research interests are network and distributed system security, wireless networking, and mobile computing. He received the US NSF CAREER Award in 2016.

Lei Xie (Member, IEEE) received the B.S. and Ph.D. degrees from Nanjing University, China, in 2004 and 2010, respectively, all in computer science. He is currently a Professor with the Department of Computer Science and Technology, Nanjing University. He has published over 100 papers in IEEE TRANSACTIONS ON MOBILE COMPUTING, ACM/IEEE TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, *ACM Transactions on Sensor Networks*, ACM MobiCom, ACM UbiComp, ACM MobiHoc, IEEE INFOCOM, IEEE ICNP, and IEEE ICDCS.



Yanchao Zhang (Fellow, IEEE) received the B.E. degree in computer science and technology from the Nanjing University of Posts and Telecommunications in 1999, the M.E. degree in computer science and technology from the Beijing University of Posts and Telecommunications in 2002, and the Ph.D. degree in electrical and computer engineering from the University of Florida in 2006. He is currently a Professor with the School of Electrical, Computer and Energy Engineering, Arizona State University. His primary research interests are network and distributed system security, wireless networking, and mobile computing. He is an Editor of IEEE/ACM TRANSACTIONS ON NETWORKING and served on the editorial board of many other journals. He also chaired the 2017 IEEE Conference on Communications and Network Security (CNS), the 2019 ARO Workshop on Proactive and Autonomous Defenses in Wireless Networks, the 2016 ARO Workshop on Trustworthy Human-Centric Social Networking, the 2015 NSF Workshop on Wireless Security, and the 2010 IEEE GLOBECOM Communication and Information System Security Symposium. He was also a TPC Area Chair for IEEE INFOCOM from 2016 to 2022.