ShakeReader: ‘Read’ UHF RFID Using Smartphone

Kaiyan Cui, *Student Member, IEEE*, Yanwen Wang, *Member, IEEE*, Yuanqing Zheng, *Member, IEEE*, and Jinsong Han, *Senior Member, IEEE*

Abstract—UHF RFID technology has become increasingly popular in stores, since it can quickly read a large number of RFID tags from afar. The deployed RFID infrastructure, however, does not directly benefit smartphone users in stores, mainly because smartphones cannot read UHF RFID tags or fetch relevant information. This article aims to bridge the gap and allow users to ‘read’ UHF RFID tags using their smartphones, without any hardware modification to either deployed RFID systems or smartphone hardware. To ‘read’ an interested tag, a user makes a predefined smartphone gesture in front of an interested tag. The smartphone gesture causes changes in 1) RFID measurement data captured by RFID infrastructure, and 2) motion sensor data captured by the user’s smartphone. By matching the two data, our system (named *ShakeReader*) can pair the interested tag with the corresponding smartphone, thereby enabling the smartphone to indirectly ‘read’ the interested tag. We build a novel reflector polarization model to analyze the impact of smartphone gesture to RFID backscattered signals. We enhance the basic version of *ShakeReader* [7] by improving its performance in densely deployed scenarios. Experimental results show that *ShakeReader* can accurately pair interested tags with their corresponding smartphones with an accuracy of $>96.3\%$.

Index Terms—Human-RFID interaction, reflector polarization model, RFID system

1 Introduction

RADIO Frequency IDentification (RFID) technology has been widely used in retail stores (e.g., UNIQLO [15], Zara [19], *etc.*) for logistics, sales tracking and shopping behavior analysis. Compared with traditional labelling technologies (e.g., QR-code, NFC), Ultra High Frequency (UHF) RFID is more attractive to stores, because it allows quick scanning of a large number of RFID-labelled items, achieving much higher operation efficiency. Leveraging the deployed RFID infrastructure, merchants can also capture customers’ interests by analyzing RFID data and optimize marketing strategy to maximize their profits [45]. As such, more and more stores are expected to deploy UHF RFID systems in the future.

Such a deployed RFID infrastructure, however, does not directly benefit customers during shopping. For example, while detailed item information (e.g., coupon, promotion, price comparisons, matching tips) could be potentially accessed, flexibly updated, and presented on smartphones, such item-specific information is not available to customers in physical stores. That is mainly because smartphones are limited by the unavailability of any direct communication with UHF RFID tags. This paper aims to enable users to ‘read’ on-the-fly item-specific information by bridging the gap between the deployed RFID infrastructure and smartphones without making any hardware modification to either RFID system or smartphones.

In this paper, we develop a system named *ShakeReader*, which allows a user to interact with an RFID-labelled item by simply performing a pre-defined gesture (e.g., shaking a smartphone) nearby the interested tag and automatically delivering item-specific information to the smartphone. Fig. 1 illustrates a usage scenario. Interested in a box of milk, a user makes a pre-defined gesture with her smartphone. Such a gesture causes changes to backscattered signal of the labelled RFID tag attached to the milk box. The changes in backscattered signal can be captured by an RFID reader. Meanwhile, the user’s smartphone detects the smartphone gesture using motion sensors. By matching the two data capturing the same smartphone gesture, *ShakeReader* can deliver the interested tag information to the corresponding smartphone user.

We note that our objective is *not* to replace other labelling technologies (e.g., QR-code, NFC), but is to provide a technology that could allow users to read the readily-deployed UHF tags in stores. We believe this technology can complement other labelling technologies in practice.

- *Kaiyan Cui is with the School of Software Engineering, Xi’an Jiaotong University, Xi’an, Shaanxi 710000, China, and also with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. E-mail: kaiyan.cui@connect.polyu.hk.*
- *Yanwen Wang is with the College of Electrical and Information Engineering, Hunan University, Changsha, Hunan 410000, China, and also with Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. E-mail: wangyw@hnu.edu.cn.*
- *Yuanqing Zheng is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. E-mail: csyqzheng@comp.polyu.edu.hk.*
- *Jinsong Han is with the School of Cyber Science and Technology, College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang 310027, China, and also with the Key Laboratory of Blockchain and Cyberspace Governance of Zhejiang Province, Hangzhou, Zhejiang 310018, China. E-mail: hanjinsong@zju.edu.cn.*

Manuscript received 3 Feb. 2021; revised 14 June 2021; accepted 9 July 2021. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding authors: Yuanqing Zheng and Jinsong Han.)
Digital Object Identifier no. 10.1109/TMC.2021.3098818

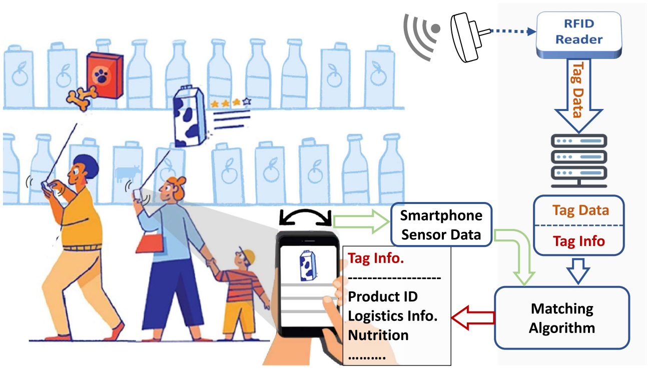


Fig. 1. Application scenario. A lady 'reads' the item-specific logistic information of a box of milk by making a gesture with her smartphone. An RFID reader reads RFID tags and measures their phase readings (i.e., Tag Data). Smartphone measures IMU sensor data (i.e., Sensor Data) and sends to a matching algorithm. The matching algorithm pairs the smartphone with the corresponding interacted tag and forwards tag information to the user (e.g., product ID and logistics information).

Although useful in practice and simple in concept, the system entails tremendous technical challenges. First, despite plenty of previous works on RFID and mobile sensing, it is still challenging to use only one tag, which remains static and is not attached on the smartphone, for accurately recognizing the smartphone gesture performed nearby. Second, users in stores may influence the gesture detection accuracy as other human activities may influence backscattered signal of RFID tags. Third, many users may perform similar gestures near multiple tags in the same store. How to correctly pair each tag with its corresponding smartphone is challenging in practice.

In this paper, we address all the above challenges. First, *ShakeReader* builds a reflector polarization model to characterize the backscattered signal of a single tag caused by smartphone gestures. This reflection model simultaneously captures backscattered signal propagation and the polarization caused by smartphone reflection. By leveraging the polarization of reflected signal from smartphones, RFID readers can identify smartphone gestures even with a single tag. Second, we notice that irrelevant user movement indeed influences the backscattered signal measurement and may cause detection errors if not handled properly. To address this problem, *ShakeReader* pre-defines a smartphone gesture (clockwise and counter-clockwise rotation of smartphone in front of an interested tag) to facilitate the detection. Third, to pair the interested tag with its corresponding smartphone, *ShakeReader* leverages the synchronicity of the changes in RFID data and smartphone sensor data simultaneously affected by the same smartphone gesture. The synchronicity allows us to differentiate the smartphone gestures performed by different users in front of their interested tags.

The key contributions can be summarized as follows:

- We present *ShakeReader*, a system that enables a flexible human-RFID interaction using smartphones. *ShakeReader* allows smartphone users to indirectly 'read' UHF RFID tags using their smartphones, without any hardware modification to either the deployed RFID infrastructure or smartphones.
- We characterize and analyze the reflector polarization and its impact on backscattered signal in RFID systems.
- We propose a new algorithm, called FSS (**F**luctuation, **S**ymmetry, and **S**imilarity) to accurately determine the real interacted tag in the product-intensive environment.
- We conduct extensive evaluations on our proposed prototype system using COTS RFID system. The experimental results show that *ShakeReader* achieves $>96.3\%$ matching accuracy.

The rest of this paper is organized as follows. Section 2 describes the background and the problem specification of this paper. In Section 3, we introduce the reflector polarization model. The *ShakeReader* design is detailed in Section 4. We enhance the basic version of *ShakeReader* to cope with practical factors in Section 5. Section 6 presents experimental results and Section 7 discusses the limitations of *ShakeReader*. Related works are summarized in Section 8. Finally, Section 9 concludes this paper.

2 Background and Motivation

2.1 UHF RFID Technology and Existing Works

UHF RFID Technology in Stores. UHF RFID technology has been increasingly used in retail stores. For example, UNIQLO is currently using UHF RFID tags to label all the items to improve operational efficiency [15]. As UHF RFID supports wireless identification from afar, retailers are freed from manually scanning items one-by-one using handheld QR-code/NFC readers. The UHF RFID technology also helps reduce customers' waiting time in the checkout queue, as RFID-labelled items can be instantly identified with RFID readers at checkout counters. As such, we expect more stores will deploy UHF RFID systems to improve operational efficiency. We note that the objective of *ShakeReader* is not to replace alternative labelling technologies (e.g., QR-code, NFC) but allow users to read the already-deployed UHF RFID tags in stores with their smartphones.

Current Smartphones Cannot Read UHF RFID Tags. While NFC tags can be read by NFC-enabled smartphones, most smartphones cannot read the deployed UHF RFID tags in stores. In order to wirelessly energize UHF RFID tags, a UHF reader needs to transmit continuous waves at high transmission power, which may quickly drain the battery of a smartphone. Although retailers can afford a handheld UHF reader and re-charge the reader more frequently in stores, customers could be reluctant to purchase extra hardware to read the UHF tags and concerned about the battery life of the smartphone.

Existing Works. Research works strive to enable smartphones to read UHF RFID tags. For example, TiFi [3] proposes to read tag IDs using RFID readers and broadcast tag IDs as Wi-Fi beacons, so that smartphones equipped with Wi-Fi modules can receive the tag IDs. However, as all tag IDs will be broadcast to smartphones, it is very challenging to correctly identify the interested tag among all the tag IDs.

2.2 System Architecture and Problem Definition

We assume that all N items are labelled with UHF RFID tags and the tags are covered by RFID readers. In practice, one reader can connect multiple reader antennas deployed in different locations. The readers continuously interrogate the tags and measure the backscattered signal of the tags

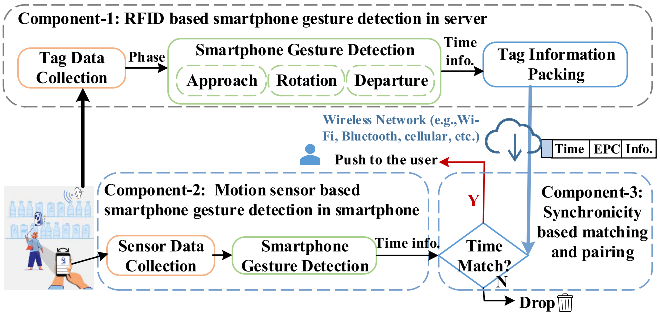


Fig. 2. System architecture of *ShakeReader*. *ShakeReader* consists of three main functional components: Component-1 processes RFID data and detects smartphone gestures in the server side, then broadcasts the packet to all clients via a wireless network; Component-2 processes IMU sensor data collected at the smartphone and detects the smartphone gestures; and Component-3 processes the timing information extracted by the other two components and pairs a smartphone with an interacted tag in the client side.

(e.g., phase, signal strength). M clients in the environment specify their interests in tags by making pre-defined smartphone gestures (i.e., clockwise and counter-clockwise rotation of smartphone) near the interested tags.

Fig. 2 illustrates the system overview and dataflow. A client makes a smartphone gesture to specify the intention to fetch information about an interested tag. The server collects tag data from RFID readers and identifies the interested tag among many co-existing tags in the environment. The server also records the starting and finishing timestamps of the smartphone gesture. Along with the coarse-grained timing information, the server next examines the fine-grained patterns in RFID measurement data caused by smartphone gesture. Such timing information is broadcast to all clients over a wireless network. Meanwhile, a mobile application running in client's smartphone records the motion sensor data and identifies the smartphone gesture.

The key objective is to pair an interested tag T_i $(1 \leq i \leq N)$ with its corresponding client C_j $(1 \leq j \leq M)$ based on RFID and sensor measurements. The smartphone gesture generates two different data streams: 1) backscattered signal data in RFID system, and 2) motion sensor data in smartphone, respectively. The synchronicity of the same event (i.e., smartphone gesture) provides an opportunity to correctly pair the interested tag with its corresponding smartphone.

3 Modelling Reflector Polarization

Referring to Fig. 3, we illustrate the signal propagation and polarization of a rotating smartphone. The RFID system uses a circularly-polarized antenna, which transmits a combination of vertical waves v and horizontal waves h with the phase difference of $\pi/2$. We use ρ_T to denote the tag polarized direction, and ρ_R to denote the long-axis direction of the reflector (i.e., smartphone). α, β, and γ represent different angles between the polarized directions.

Suppose the reader transmits $S_A(t)$:

$$S_A(t) = h \cdot \cos(kt - \phi_A) + v \cdot \sin(kt - \phi_A), \tag{1}$$

where ϕ_A is the constant phase offset induced by the transmitter circuit.

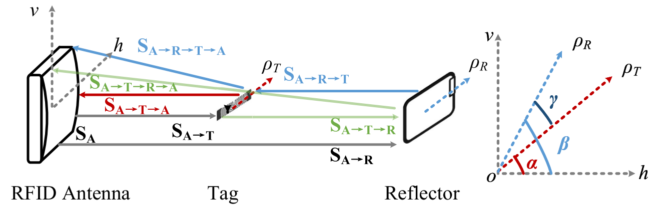


Fig. 3. Reflector polarization model and angle relationship between tag, reflector and RFID antenna.

3.1 Antenna-Tag-Antenna

Due to the tag polarization [1], [18], [22], [28], the signal emitted by the reader and arrived at the tag $S_{A\to T}(t)$ will be projected to the direction of the tag polarization ρ_T. Thus, we have:

$$\begin{cases} S_{A\to T}(t) = \rho_T \cdot S_A(t - t_{A\to T}) \\ \qquad\quad = (\rho_T \cdot h)\cos(kt - \phi_{AT} - \phi_A - \phi_T) \\ \qquad\quad\;\; + (\rho_T \cdot v)\sin(kt - \phi_{AT} - \phi_A - \phi_T) \\ \qquad\quad = \cos(\alpha)\cos(kt - \phi_{AT} - \phi_A - \phi_T) \\ \qquad\quad\;\; + \sin(\alpha)\sin(kt - \phi_{AT} - \phi_A - \phi_T) \\ \quad \phi_{AT} = 2\pi d_{A\to T}/\lambda \quad mod \quad 2\pi \end{cases}, \tag{2}$$

where $t_{A\to T}$ represents the propagation time from the reader antenna to the tag, ϕ_{AT} represents the phase change corresponding to the signal distance change $d_{A\to T}$, and ϕ_T denotes the phase shift caused by the tag's hardware.

Similarly, the backscattered signal of tag to reader $S_{A\to T\to A}(t)$ projects to both the reader polarized directions h and v. Therefore, we will receive two sub-signals $S^h_{A\to T\to A}(t)$ and $S^v_{A\to T\to A}(t)$ corresponding to the antenna polarized direction h and v, respectively. Thus, we have:

$$\begin{cases} S^h_{A\to T\to A}(t) = \cos(\alpha) S_{A\to T}(t - t_{T\to A}) \\ S^v_{A\to T\to A}(t) = \sin(\alpha) S_{A\to T}(t - t_{T\to A}) \end{cases}. \tag{3}$$

The backscattered signal of tag $S_{A\to T\to A}(t)$ is the combination of $S^h_{A\to T\to A}(t)$ and $S^v_{A\to T\to A}(t)$ as follows:

$$\begin{cases} S_{A\to T\to A}(t) = S^h_{A\to T\to A}(t) + S^v_{A\to T\to A}(t - t_{\pi/2}) \\ \qquad\qquad = \cos(2\alpha)\cos(kt - 2\phi_{AT} - \phi') \\ \qquad\qquad\;\; + \sin(2\alpha)\sin(kt - 2\phi_{AT} - \phi') \\ \quad \phi_{AT} = 2\pi d_{A\to T}/\lambda \quad mod \quad 2\pi \\ \quad \phi' = \phi_A + \phi_T + \phi'_A \end{cases}, \tag{4}$$

where ϕ'_A is the phase offset induced by the receiver circuit of the reader antenna. ϕ' is a constant value related to hardware of tag and reader. As a result, we can see that the backscattered signal of tag $S_{A\to T\to A}$ is influenced by both the distance $d_{A\to T}$ and the angle between the tag and antenna α.

Previous works [17], [40] have studied the influence of the tag's orientation on phase values (i.e., antenna-tag-antenna). However, the previous models do not consider the reflector polarization and its impact on backscattered signal.

3.2 Modelling Reflector Polarization

To further characterize the backscattered signal in our scenario, we consider a scenario with a reflector (i.e.,

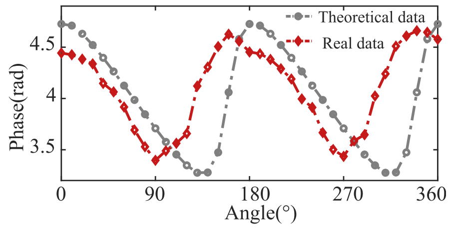


Fig. 4. The comparison between the real phases and the theoretical phases.

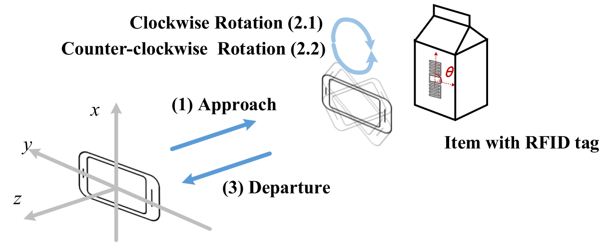


Fig. 5. Illustration of the pre-defined smartphone gesture.

smartphone). The signal emitted by the reader and arriving at the reflector $S_{A\to R}(t)$ is:

$$\begin{cases} S_{A\to R}(t) = \rho_R \cdot S_A(t - t_{A\to R}) \\ \quad = \cos(\beta)\cos(kt - \phi_{AR} - \phi_A - \phi_R) \\ \quad\quad + \sin(\beta)\sin(kt - \phi_{AR} - \phi_A - \phi_R) \\ \phi_{AR} = 2\pi d_{A\to R}/\lambda \mod 2\pi \end{cases}, \tag{5}$$

where ϕ_R is the phase offset caused by the reflector.

Then $S_{A\to R}(t)$ will be reflected to the tag and the signal $S_{A\to R\to T}(t)$ can be expressed as:

$$S_{A\to R\to T}(t) = \cos(\gamma) S_{A\to R}(t - t_{R\to T}). \tag{6}$$

$S_{A\to R\to T}(t)$ will arrive at the reader antenna and project on two antenna's polarization direction $S^h_{A\to R\to T\to A}(t)$ and $S^v_{A\to R\to T\to A}(t)$ as follows:

$$\begin{cases} S^h_{A\to R\to T\to A}(t) = \cos(\alpha) S_{A\to R\to T}(t - t_{T\to A}) \\ S^v_{A\to R\to T\to A}(t) = \sin(\alpha) S_{A\to R\to T}(t - t_{T\to A}) \end{cases}. \tag{7}$$

Thus, the final arrived signal at the reader $S_{A\to R\to T\to A}(t)$ can be formulated as follows:

$$\begin{cases} S_{A\to R\to T\to A}(t) = S^h_{A\to R\to T\to A}(t) + S^v_{A\to R\to T\to A}(t - t_{\pi/2}) \\ \quad = \cos(\alpha+\beta)\cos(\gamma)\cos(kt - \phi_{ARTA} - \phi'') \\ \quad\quad + \sin(\alpha+\beta)\cos(\gamma)\sin(kt - \phi_{ARTA} - \phi''). \\ \phi_{ARTA} = 2\pi d_{A\to R\to T\to A}/\lambda \mod 2\pi \\ \phi'' = \phi_A + \phi_R + \phi_T + \phi'_A \end{cases} \tag{8}$$

From Eq. (8), we observe that the backscattered signal $S_{A\to R\to T\to A}$ is a function of the distance and the relative angles among reader, tag and reflector.

Similarly, the received signal propagated along another path $S_{A\to T\to R\to A}$ can be modelled. Note that $S_{A\to R\to T\to A}$ and $S_{A\to T\to R\to A}$ are reciprocal with the same propagation distance and the same polarization directions.

Finally, the received signal of antenna $R(t)$ can be modelled:

$$\begin{cases} R(t) = S_{A\to T\to A}(t) + S_{A\to R\to T\to A}(t) + S_{A\to T\to R\to A}(t) \\ \quad = \cos(kt - 2\phi_{AT} - \phi' - 2\alpha) \\ \quad\quad + 2\cos(\gamma)cos(kt - \phi_{ARTA} - \phi'' - \alpha - \beta) \\ \phi_{AT} = 2\pi d_{A\to T}/\lambda \mod 2\pi \\ \phi_{ARTA} = 2\pi d_{A\to R\to T\to A}/\lambda \mod 2\pi \\ \phi' = \phi_A + \phi_T + \phi'_A \\ \phi'' = \phi_A + \phi_R + \phi_T + \phi'_A \\ \gamma = |\beta - \alpha| \end{cases}. \tag{9}$$

Key Observation: *The distance and the polarization directions of tag, reflector, and antenna jointly affect the received backscattered signal.*

We conduct an experiment to validate our proposed reflector polarization model. In the experiment, we ensure that both tag and reader antenna are fixed and only rotate the reflector (i.e., change of β) for one circle. Specifically, we use an iPhone 7 ($67.1mm \times 138.3mm$) as a reflector to rotate 360 degrees counter-clockwise at $5cm$ in front of the tag. The distance between the tag and reader's antenna is $15cm$ and the angle between them is 0 (i.e., $\alpha = 0$). The result is shown in Fig. 4. We observe that the phase changes with the rotation of the reflector and the changes of the measured phases are consistent with the theoretical phases. Note that the overall deviations of the phase values are introduced by the unknown parameters ϕ' and ϕ'' in Eq. (9). The experiment result demonstrates the validity of our reflector polarization model, which can be applied when capturing and differentiating a pre-defined gesture from other movements nearby.

4 System Design

In this section, we first design an interactive smartphone gesture between smartphones and RFID tags, and then elaborate our three key functional components: Component-1) RFID based smartphone gesture detection in server; Component-2) motion sensor based smartphone gesture detection in smartphone; and Component-3) synchronicity based matching and pairing for interested tags and their corresponding smartphones.

4.1 Smartphone Gesture

Based on our reflector polarization model, we design a simple yet effective pre-defined smartphone gesture to specify user's interest in a tag, as shown in Fig. 5. The user first holds the smartphone horizontally then approaches the interested tag. Next, the user rotates the smartphone 180 degrees clockwise rotation followed by a symmetric 180 degrees counter-clockwise rotation and finally departs from the tag. During the entire interaction, the phone should be held vertically to interact with the tag and the direction of the phone's Z-axis should point straight ahead and remain perpendicular to the direction of gravity. Note that the pre-defined gesture does not require strict rotation angle.

To visualize the changes in RFID data as well as the sensor data caused by the gesture, we ask a volunteer to perform a smartphone gesture and measure both RFID data and motion sensor data in Fig. 6.

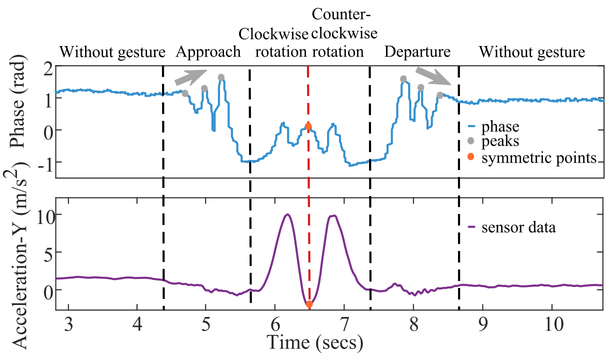


Fig. 6. Phase measurements (upper panel) and sensor data (lower panel) during the interaction.

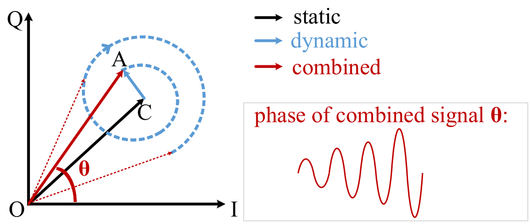


Fig. 7. The changes of received signal phase when a reflector is approaching.

We observe that the phase measurements remain flat before the smartphone gesture and start to fluctuate during the interaction. The phase changes caused by the interaction are divided into three periods: approach, rotation and departure. On the other hand, when approaching and leaving, acceleration readings in Y-axis are very small, since Y-axis is mostly perpendicular to gravity. As a user rotates the phone, the acceleration readings clearly exhibit two increasing-and-decreasing patterns. In the following, we first focus on the RFID data and analyze the phase changes.

4.2 RFID based Smartphone Gesture Detection

4.2.1 Approach and Departure Patterns

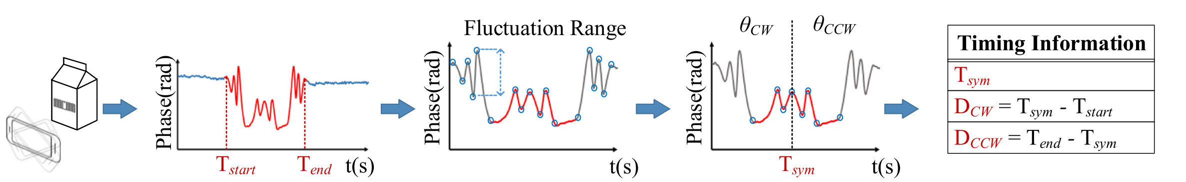
As shown in Fig. 6, when the phone is far away from the tag, the phase values remain stable. As the distance does not change during this period, the phase readings remain almost constant subject to small noise. Once the phone starts to approach or depart from the tag, the reflected signal from the smartphone will affect the phase measurements. Thus, the phase measurements of the interested tag will fluctuate with the distance change between the tag and the phone.

More importantly, as the phone approaches, the backscattered signal exhibits the specific approach pattern and its fluctuation range (i.e., the difference between the local maximum and the local minimum of phase readings) is becoming larger because the reflected signal strength from the smartphone increases. In contrast, the fluctuation range will decrease when the phone departs.

To help better understand the approach and departure patterns, we take the approach gesture as an example and illustrate in Fig. 7. When the smartphone approaches the tag, the received signal consists of two components: the static component $\overrightarrow{OC}$, and the dynamic component $\overrightarrow{CA}$ with varying phase and signal strength. In this process, the static component $\overrightarrow{OC}$ keeps unchanged because both the reader antenna and tag are static. As the length of the reflection path $d_{A \to R \to T}$ (Antenna-Reflector-Tag) decreases continuously, the signal strength of the dynamic component increases while the phase rotates, resulting in the spiral changing pattern of $\overrightarrow{CA}$ (i.e., blue spiral curve in Fig. 7). Therefore, the measured combined signal (red arrow $\overrightarrow{OA}$) will fluctuate around $\overrightarrow{OC}$ with an increasing oscillation radius [38]. As a result, the fluctuation range of the approach pattern exhibits an increasing trend. Similarly, when the reflector moves away from the tag, the peaks of the fluctuation will decrease gradually.

Based on this observation, we measure the standard deviation of phase readings to detect the start and the end of a gesture. In particular, we apply a moving window to scan the phase measurements and continuously calculate the standard deviation of the phase measurement in the window. The standard deviation will remain small without gestures. When the standard deviations of three consecutive windows exceed a threshold, we consider that one gesture starts to affect the tag. If the standard deviations of three consecutive windows are below the threshold and the phase readings return to the original phase readings measured before the gesture, we consider the gesture to be finished. We record the starting point timestamp T^{RFID}_{start} and finishing point timestamp T^{RFID}_{end} as shown in Fig. 8a. Based on the empirical measurement, we set the size of moving window in this step to 151 samples (about 0.8 seconds) to make a balance between processing time and accuracy. For the threshold, we empirically set it to 0.21, which is approximately 15 times the average standard deviation of the phase readings from 100 collected traces without gestures.

However, we note that dynamics in the environment are likely to cause various changes in the tag phase readings. In order to accurately detect approach and departure patterns, we first find the local maximums and local minimums of phase readings, then measure the *differences between two adjacent local maximum and local minimum* defined as *fluctuation range*. If there are two or more consecutive fluctuations and the fluctuation



(a) Cut the phase readings associated with the gesture

(b) Detect the approach pattern and departure pattern

(c) Find the symmetric point using DTW algorithm DTW(θ_{CW} ,*flip*(θ_{CCW}))

(d) Extract three key timing information on the tag signal

Fig. 8. Timing information extraction on tag signal.

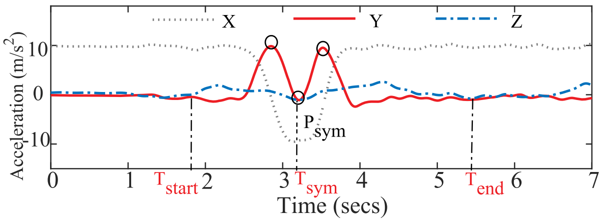
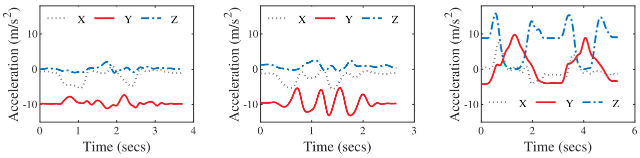


Fig. 9. The changes of acceleration readings in the x, y, and z axes during the interaction.



(a) Walking (b) Running (c) Picking up and putting down

Fig. 10. The changes of acceleration reading from common daily activities.

range exhibits an increasing trend (as illustrated in Fig. 8b), we consider that the phone is approaching. In contrast, the continuous decreases in the fluctuation range indicate that the smartphone is departing from the tag. In practice, some movements may cause similar phase changing patterns as in approach and departure events. In the following, we design a unique smartphone gesture to facilitate the detection and improve the detection robustness.

4.2.2 Rotation Pattern

To improve the detection robustness against the dynamics and background noise in the environment, we define a smartphone gesture (clockwise and counter-clockwise rotation of smartphone). As analyzed in Section 3, smartphone polarization can affect the received backscattered signal. In Fig. 6, we have an interesting observation.

Observation: Phase changes caused by the defined smartphone gesture are generally symmetric.

We observe that the phase reading shows an 'M' or 'W' shape because the smartphone gesture is symmetrical. As a result, RFID readers can leverage such prior knowledge and detect a pre-defined smartphone gesture. Note that such a symmetric pattern in our pre-defined gesture can be used to disambiguate human activities (i.e., human movement), which do not generate symmetric patterns.

Although the rotation angles of the clockwise and counter-clockwise are generally symmetrical, the rotation time and speed can be slightly different, resulting in misaligned phase waveforms. To accurately detect the symmetric point and use that as the timing information, we adopt the Dynamic Time Warping (DTW) algorithm to match the slightly misaligned phase waveforms measured in clockwise and counter-clockwise rotations. We first select the local maximums and local minimums on phase readings of rotation as a candidate set of symmetric points $\{SP_1, SP_2\ ,\dots, SP_k, \dots, SP_K\}$. Next, we divide the tag signals into two parts: clockwise signal $\theta_{CW}(k)$ before the symmetric point SP_k and counter-clockwise signal $\theta_{CCW}(k)$ after the symmetric point as shown in Fig. 8c. Then, we use DTW algorithm to calculate the distance between the $\theta_{CW}(k)$ and the flipped counter-clockwise signal, $flip(\theta_{CCW}(k))$:

$$Distance(k) = DTW(\theta_{CW}(k), flip(\theta_{CCW}(k))), k \in [1, K]. \tag{10}$$

The minimum distance indicates the highest similarity of $\theta_{CW}(k)$ and $flip(\theta_{CCW}(k))$. We notice that the time difference between clockwise and counter-clockwise rotation of smartphone performed by users are generally less than 1 second. Therefore, the DTW algorithm in our experiment tolerates clockwise and counter-clockwise rotation waveforms with a maximum misalignment of 1 second. As a result, we can find the true symmetric point and filter out noise in the environment (e.g., user movement, random signal fluctuation).

4.2.3 Timing Information Extraction on Tag Signal

Based on the observations, we can extract three key timing information on the backscattered signal of RFID tag T_i ($1 \le i \le N$) as shown in Fig. 8d:

- Absolute timestamp of symmetric point $T^{\mathtt{RFID}}_{sym}(T_i)$.
- Clockwise rotation duration $D^{\mathtt{RFID}}_{CW}(T_i)$: the difference between symmetric point timestamp and starting point timestamp $T^{\mathtt{RFID}}_{start}(T_i)$, i.e., $D^{\mathtt{RFID}}_{CW}(T_i) = T^{\mathtt{RFID}}_{sym}(T_i) - T^{\mathtt{RFID}}_{start}(T_i)$.
- Counter-clockwise rotation duration $D^{\mathtt{RFID}}_{CCW}(T_i)$: the difference between symmetric point timestamp and finishing point timestamp $T^{\mathtt{RFID}}_{end}(T_i)$, i.e., $D^{\mathtt{RFID}}_{CCW}(T_i) = T^{\mathtt{RFID}}_{end}(T_i) - T^{\mathtt{RFID}}_{sym}(T_i)$.

4.3 Motion Sensor based Smartphone Gesture Detection

After detecting the gesture from the RFID data, we need to perform gesture detection on user's smartphone and pair the smartphone to the corresponding tag.

4.3.1 Smartphone Gesture Detection

In the above discussion, we only focus on the acceleration readings in the Y-axis for concise presentation. In practice, X-axis and Z-axis acceleration readings can complement and enhance the gesture detection as shown in Fig. 9. For comparison, we also plot the acceleration readings from various human activities (e.g., walking, running, and picking up and putting down) in Fig. 10. We notice acceleration readings exhibit different patterns when a user performs our pre-defined gesture and other daily activities.

Since the phone is held horizontally in the initial state, we observe that the acceleration readings in Y-axis and Z-axis are close to zero, and the acceleration readings in X-axis are close to the gravitational acceleration $9.8m^2/s$. Therefore, we can determine the initial state of our defined gesture by measuring the initial pattern of acceleration readings.

Next, we need to detect the approach pattern and departure pattern. We find when the phone starts moving toward the tag along the Z-axis, the Z-axis acceleration readings will increase from 0. To detect the starting and finishing time of smartphone gesture, we calculate the standard deviations of Z-axis readings in each moving window. If the

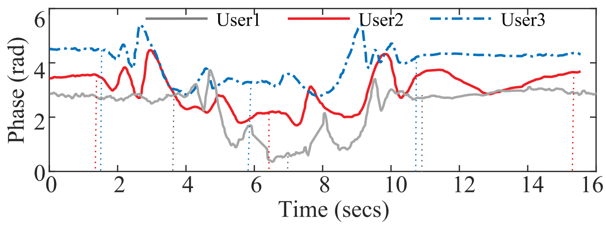


Fig. 11. Phase changes caused by different users.

standard deviations exceed a threshold for three consecutive windows, we consider that the smartphone is approaching the tag and departing when the standard deviations drop below the threshold for three consecutive windows. When a user finishes this interaction gesture, the acceleration readings in all three axes will return to the initial state. Meanwhile, we record the starting point timestamp T^{Phone}_{start} and finishing point timestamp T^{Phone}_{end}. In our experiments, the size of moving window is 0.8 seconds (80 samples at the fixed sensor sampling rate of $100Hz$), which is consistent with the threshold for RFID-based gesture detection in Section 4.2.1. In addition, to find a reliable threshold for detecting the T^{Phone}_{start} and T^{Phone}_{end}, we first measure that the average standard deviation of the Z-axis readings from 100 collected traces in the initial state is 0.19. Based on the experimental observations, we set the threshold to 0.57, which is 3 times the measured average value.

Then, we identify smartphone rotation by measuring the acceleration readings in Y-axis. In the initial state, the acceleration readings in Y-axis are expected to be small and stable. In contrast, once the phone starts rotation, its readings change from 0 to $9.8m^2/s$. As the user rotates clockwise and then counter-clockwise, the acceleration readings in Y-axis exhibit two peaks. Hence, we search for local maximum values and local minimum values and extract the key timing information. Our observation is that the smartphone gesture is symmetric, and the symmetric point is the local minimum (corresponding to the horizontal pose after clock-wise rotation) between two local maximums (corresponding to the two vertical poses during the clock-wise and counter clock-wise rotations, respectively). As a result, we can identify the symmetric point P_{sym}: the local minimum between two peaks and its Y-axis acceleration reading near zero. In this way, we obtain the timestamp of symmetric point T^{Phone}_{sym}.

4.3.2 *Timing Information Extraction on Sensor Data*

Based on the above observation, Component-2 detects the pre-defined smartphone gesture and extracts the timing information for each client smartphone C_j $(1 \leq j \leq M)$ as follows.

- Absolute timestamp of symmetric point $T^{\text{Phone}}_{sym}(C_j)$.
- Clockwise rotation duration $D^{\text{Phone}}_{CW}(C_j)$: the difference between symmetric point timestamp and starting point timestamp, i.e., $D^{\text{Phone}}_{CW}(C_j) = T^{\text{Phone}}_{sym}(C_j) - T^{\text{Phone}}_{start}(C_j)$.
- Counter-clockwise rotation duration D^{Phone}_{CCW}: the difference between symmetric point timestamp and finishing point timestamp $T^{\text{Phone}}_{end}(C_j)$, i.e., $D^{\text{Phone}}_{CCW}(C_j) = T^{\text{Phone}}_{end}(C_j) - T^{\text{Phone}}_{sym}(C_j)$.

4.4 Synchronicity Based Matching and Pairing

As the backscattered signal and the sensor data are simultaneously affected by the same gesture, we leverage the synchronicity of the signals to pair the interacted tag and the corresponding smartphone. Instead of mapping all the data points in two data streams, we only match backscattered signal and the sensor data using the extracted key time information to reduce computation time and network traffic.

We design a sequence matching algorithm based on the following three key observations: (1) The rotation gesture is generally performed within a certain period P (e.g., 5s); (2) Different users may generate different key timing information; and (3) The key timing information of backscattered signal and sensor data caused by the same gesture are synchronized. Based on these observations, we match tag T_i $(1 \leq i \leq N)$ with client C_j $(1 \leq j \leq M)$ (denoted as $T_i \mapsto C_j$), if all following conditions are satisfied:

- **C1**: $D^{\text{RFID}}_{CW}(T_i) + D^{\text{RFID}}_{CCW}(T_i) \leq P$,
- **C2**: $D^{\text{Phone}}_{CW}(C_j) + D^{\text{Phone}}_{CCW}(C_j) \leq P$
- **C3**: $T^{\text{RFID}}_{sym}(T_i) = T^{\text{Phone}}_{sym}(C_j)$
- **C4**: $D^{\text{RFID}}_{CW}(T_i) = D^{\text{Phone}}_{CW}(C_j)$
- **C5**: $D^{\text{RFID}}_{CCW}(T_i) = D^{\text{Phone}}_{CCW}(C_j)$

However, such strict timing requirements may not be satisfied in practice. For example, due to the ALOHA protocol of RFID system as well as the different sampling rates of the backscattered signal and the sensor data, the RFID signal and sensor readings may not be exactly matched. To address this practical issue, we relax the conditions (**C3** - **C5**) by tolerating a small mismatch δ in the time domain. For example, we relax **C3** as follows:

- *Relaxed C3*: $|T^{\text{RFID}}_{sym}(T_i) - T^{\text{Phone}}_{sym}(C_j)| \leq \delta$

We note that a smaller δ indicates a tighter timing requirement, which can reduce the possibility of incorrectly matching two streams generated by different gestures but meanwhile increase the chance of missing two streams originated by the same gesture. We empirically tune δ and set δ to $400ms$.

Why do we extract three key timing information for matching? Fig. 11 plots the phase readings when three volunteers perform smartphone gestures in front of their interested tags concurrently. We notice that the timestamps of three symmetric points can be very close in time, making it hard to differentiate. Fortunately, as users tend to perform gestures differently (e.g., different speed, different duration) [37], the clockwise and the counter-clockwise duration can be different in practice. For example, the gesture duration of user 1 is shorter than that of user 2. Therefore, we extract three key timing information to differentiate users and improve robustness.

As the network traffic involved in transmitting the timing information as well as tag ID is small, the server can encapsulate the timing information of RFID data and its tag ID and broadcast a message to all clients through wireless communication. As a matter of fact, a smartphone can be connected to the Internet via various wireless networks (e.g., Wi-Fi, Bluetooth, cellular, *etc*.). Our system running in the application layer does not have a specific requirement

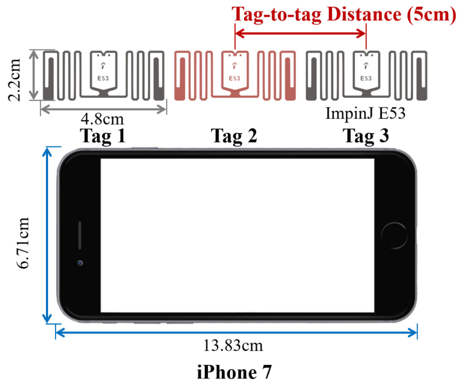


Fig. 12. Shape and size of RFID tags and smartphone.

on the networking technologies in the lower layers. In practice, messages can be transmitted using sockets from a server to a mobile client in a user's smartphone. Receiving a broadcast packet, clients test the above matching conditions if the client's smartphone has detected a smartphone gesture recently. If no smartphone gesture has been detected, a client can simply drop the broadcast packet. If all the above conditions are satisfied, the client can read the tag ID from the broadcast packet, and fetch more information about the tag from the server using the tag ID as an index. The computation overhead involved in testing the above conditions is very low and can be afforded by smartphones.

5 Cope With More Practical Factors

In practice, many factors may introduce errors. Among these factors, tag-to-tag distance and reading rates for target tags are the two crucial ones. In this section, we propose some solutions to mitigate their impacts on the performance of *ShakeReader*.

5.1 Interaction in A Product-intensive Environment

5.1.1 The Impact of Adjacent Tags

To reduce costs and increase profits, products in the store are usually placed densely. In the product-intensive environment, our system can be influenced by the adjacent tag-labelled products. As a result, our system may mis-detect the neighboring tags as the interested tag.

To visualize the effects of the adjacent tags, we place three tags (Impinj E53) in a straight line with the same tag-to-tag distance and their order from left to right is Tag 1, Tag 2, and Tag 3. A volunteer holds a smartphone (iPhone 7) to perform the pre-defined gesture in front of Tag 2 (target tag). The shape and size of RFID tags and the smartphone in this experiment are shown in Fig. 12. We vary the tag-to-tag distance (the distance between the center points of tags) from $5cm$ to $15cm$ to observe the signal changes of these three tags. Note that when the tag-to-tag distance is $5cm$, the spacing distance of two tags is only $0.2cm$ and most parts of adjacent tags are under the coverage of the smartphone during interaction.

Fig. 13 plots the phase measurements of all three tags at different tag-to-tag distances. The phase measurements of non-interacted tags (i.e., Tag 1 and Tag 3) indeed exhibit similar fluctuation patterns to that of the interacted tag (i.e., Tag 2). We mitigate the impact of adjacent tags based on the following three key observations.

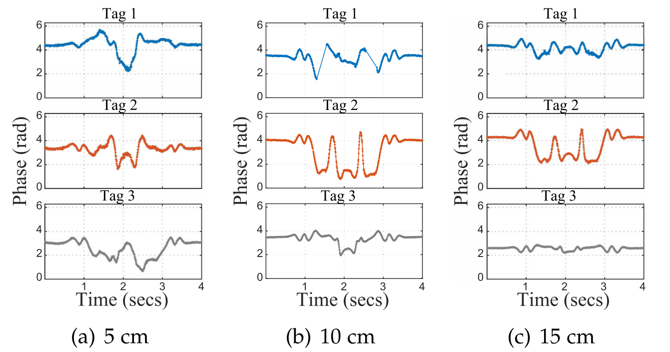


Fig. 13. The phase changes of the interacted tag and its adjacent tags.

Observation 1: The phase fluctuation of adjacent tags decreases as the tag-to-tag distance increases.

As shown in Fig. 13, as the tag-to-tag distance increases, the phase fluctuation of adjacent tags becomes less drastic compared to the interacted tag. This is because the signal strength of reflected signal on the adjacent tags becomes weaker as the distance increases. Therefore, we can find the interacted tag based on the fluctuation of the phase measurements.

To formulate the fluctuation of the signal, we measure the range of the signal phase, namely *Fluctuation*, as follows:

$$F(T_i) = max(\theta(T_i)) - min(\theta(T_i)), \tag{11}$$

where $\theta(T_i)$ is the phase measurement of tag T_i caused by smartphone gesture, where $i \in [1, N]$ and N is the number of tags.

However, we observe that signal phase exhibits similar fluctuation under a closer tag-to-tag distance, i.e., Tag 1 and Tag 2 in Fig. 13a, due to similar and strong signal strength reflected from the smartphone. Therefore, only using the fluctuation of the signal phase may not be able to detect the interacted tag under an extremely product-intensive environment.

Observation 2: The phase changes of adjacent tags show a less symmetric pattern when the tag-to-tag distance is small.

As the tag-to-tag distance increases, the influence of smartphone gesture on adjacent tags becomes weaker, which results in smaller phase changes on adjacent tags (e.g., Tag 3 in Fig. 13c). Such a flat pattern may result in higher symmetry of adjacent tags than that of the interacted tag. In contrast, when the tag-to-tag distance becomes smaller, the signal phase of the adjacent tag (e.g., Tag 3 in Fig. 13a) presents a less symmetric pattern compared to the interacted Tag 2. This is because users perform a symmetric gesture right in front of an interested tag (i.e., Tag 2) instead of adjacent tags. Therefore, when the tag-to-tag distance is relatively small, symmetry can be utilized to distinguish the interacted tag from the adjacent tags as well.

Symmetry represents the similarity of clockwise signal and flipped counter-clockwise signal induced by the symmetric gesture. Refer to the Eq. (10), a smaller distance means that these two parts of signal have a higher symmetry. To measure *Symmetry*, we reverse the distance value of Eq. (10) by multiplying -1. As a result, the tag with the maximum value of *Symmetry* is most likely the tag interacted by users. The *Symmetry* can be expressed as:

$$Sym(T_i) = max(-1 * Distance(T_i)) \tag{12}$$

However, we observe that only using the above two metrics may still be difficult to find the interacted tag. For example in Fig. 13a, we cannot determine whether the interacted tag is Tag 1 with larger *Fluctuation* or Tag 2 with higher *Symmetry*.

Observation 3: Compared with adjacent tags, the phase changes share higher similarity among the interacted tags.

Our system requires users to perform a pre-defined and fixed smartphone gesture to interact with their interested tags. During the interaction, interacted tags are within the shadow of the smartphone. As a result, the influence on the interacted tag signal is consistent. In contrast, only parts of the antenna of the adjacent tags are affected by the smartphone gesture. As a result, the similarity among the interacted tags, such as Tag 2 in Fig. 13, is higher than that of the adjacent tags. Therefore, *Similarity* can be regarded as an effective measurement to detect interacted tags.

Based on this observation, we select a phase changes caused by the pre-defined gesture as a template, and calculate the *Similarity* between the phase changes of possible tags and the template, which can be regarded as an effective metric to complement other two metrics. *Similarity* can be formulated as:

$$Sim(T_i) = max(-1 * DTW(\theta(T_i), \theta(template))), \tag{13}$$

where $\theta(template)$ is the phase measurement of the template. Note that we also adopt the Dynamic Time Warping (DTW) algorithm to calculate the similarity, since it allows the elastic transformation of time series to calculate the similarity between signals with different lengths. In addition, experiments show that the phase changes caused by our pre-defined gesture for different tags and users are relatively consistent. Therefore, we only need to collect one template even for multiple users.

5.1.2 Determine the Interacted Tags

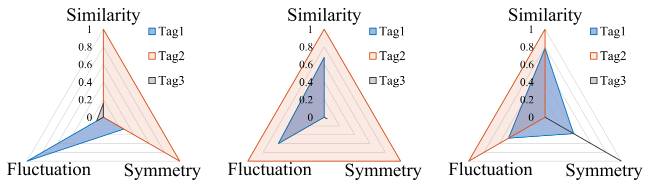
To compare these different types of metrics, we utilize min-max normalization to transfer all three metrics of $F(T_i)$, $Sym(T_i)$ and $Sim(T_i)$ into a unified range [0, 1]. The corresponding normalized metrics, $F'(T_i)$, $Sym'(T_i)$ and $Sim'(T_i)$ are denoted as:

$$F'(T_i) = \frac{F(T_i) - min(F)}{max(F) - min(F)} \tag{14}$$

$$Sym'(T_i) = \frac{Sym(T_i) - min(Sym)}{max(Sym) - min(Sym)} \tag{15}$$

$$Sim'(T_i) = \frac{Sim(T_i) - min(Sim)}{max(Sim) - min(Sim)}. \tag{16}$$

Fig. 14 plots the three metrics of corresponding tag signals in Fig. 13. We observe that only using a single metric cannot distinguish the interacted tag from others. For example, in a sparse environment in Fig. 14c, the value of *Fluctuation* of



(a) 5 cm (b) 10 cm (c) 15 cm

Fig. 14. Comparison of three metrics: Fluctuation, symmetry, and similarity.

interacted Tag 2 is larger than that of adjacent Tag 1 and Tag 3, while in densely deployed scenarios in Fig. 14a, the value of *Fluctuation* of Tag 2 is smaller than that of Tag 1. Similarly, the value of *Symmetry* can be used to find the interacted tag (i.e., Tag 2) from adjacent tags in Fig. 14b, while it fails to detect the interacted tag in Fig. 14c. The *Similarity* can serve as a complementary metric of the other two metrics to find the target tag. For example, when Tag 1 has larger *Fluctuation* and Tag 2 has higher *Symmetry* in Fig. 14a, *Similarity* can determine that Tag 2 is the interacted tag instead of the adjacent Tag 1.

Algorithm 1. Determining the Interacted Tag and the Order of Potential Tags

Input:
The set of phase measurements from potential tags, $\theta(T_1)$, $\theta(T_2)$, ..., $\theta(T_i)$, ..., $\theta(T_N)$;
The phase measurement of the template, $\theta(template)$;
Output:
The interacted tag, $T_{\mathcal{O}_1}$;
The order set about interacted probability from high to low, $\mathcal{O} = \{\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_N\}$;

for each tag T_i **do**
  calculate *Fluctuation* $F(T_i)$; {Eq. (11)}
  calculate *Symmetry* $Sym(T_i)$; {Eq. (12)}
  calculate *Similarity* $Sim(T_i)$; {Eq. (13)}
end for
for each tag T_i **do**
  normalize *Fluctuation* $F'(T_i)$; {Eq. (14)}
  normalize *Symmetry* $Sym'(T_i)$; {Eq. (15)}
  normalize *Similarity* $Sim'(T_i)$; {Eq. (16)}
end for
for each tag T_i **do**
  calculate joint metric $\mathcal{M}(T_i)$; {Eq. (17)}
end for
$\mathcal{O}$ = **Order**($\mathcal{M}$,‘descending’); {descending order of $\mathcal{M}$}
return $T_{\mathcal{O}_1}$, $\mathcal{O}$;

Therefore, we synthetically combine these three metrics and define a joint metric $\mathcal{M}$ to rank all potential tags and determine the interacted tag with the highest rank. The joint metric $\mathcal{M}$ can be formulated as

$$\mathcal{M}(T_i) = w_1 F'(T_i) + w_2 Sym'(T_i) + w_3 Sim'(T_i). \tag{17}$$

In practice, we empirically set the weights $w_1 = 0.3$, $w_2 = 0.2$, and $w_3 = 0.5$, respectively. We assign a higher weight to *Similarity* because the impact of pre-defined gesture on the interacted tag are relatively consistent among different tags and users.

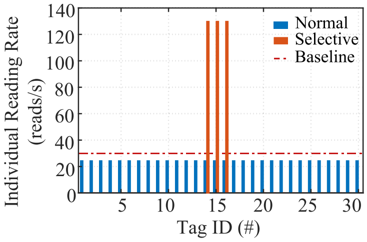


Fig. 15. The adaptive reading scheme.

The overview of the interacted tag detection algorithm is shown in Algorithm. 1, named FSS algorithm (**F***luctuation*, **S***ymmetry*, and **S***imilarity*). Specifically, we sort the joint metric $\mathcal{M}$ in descending order, and then get the order set of potential tags $\mathcal{O}$. A tag with a higher $\mathcal{M}$ is more likely to be interacted. In practice, we can push the information of Top-k tags to users to prevent information missing. We evaluate the performance of the FSS algorithm with experiments in Section. 6.

5.2 Adaptive Reading Scheme

In practice, the reading rate of the commodity RFID reader is limited. A low sampling rate in a tag-intensive environment may influence our system performance. To solve this problem, we design an adaptive reading scheme to focus on potential interacted tags while filtering out other co-existing tags in the environment.

Specifically, the adaptive reading scheme includes two reading modes: normal reading mode and selective reading mode. In the normal reading mode, the RFID reader directly sends the QUERY command to inventory all tags. By analyzing the received tag information (e.g., phase and RSS changes), we can find the potential tags that are likely being interacted by users and record their IDs.

After detecting these potential interacted tags, the adaptive reading scheme will switch to the selective reading mode and set these potential interacted tags as target tags. The selective reading mode consists of two following steps: (1) The RFID reader first sends the SELECT command to select potential tags, which is compatible with and supported by the EPC standard. (2) Then the RFID reader sends the QUERY command to only read the potential tags which can effectively increase the reading rates of potential tags.

We conduct an experiment to test the feasibility of the adaptive reading scheme. We use an RFID reader (Impinj R420) to query 30 RFID tags, which include five different types of tags (Impinj E53, Impinj H47, Alien ALN-9640, Alien ALN-9662 and Alien ALN-9629). One antenna is connected with the reader. We first adopt the normal reading mode and send QUERY commands to read all 30 tags. Then, we switch to the selective reading mode and read 3 potential tags out of the 30 tags.

Fig. 15 plots the individual reading rates in the two reading modes and the baseline is 30 reads/s. In the normal reading mode, we see that the individual reading rate of all 30 tags is about 24 reads/s, which lower than the baseline. In contrast, after adopting the selective reading mode, the individual reading rate of target tags is significantly increased to about 129 reads/s, which is 5 times larger than that of normal reading mode. Therefore, we can utilize this adaptive reading scheme to select potential tags and increase their reading rates in product-intensive environments.

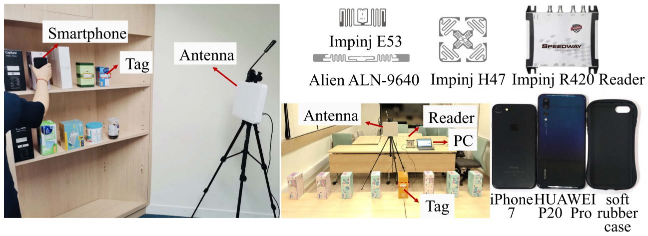


Fig. 16. Experimental environment and devices.

TABLE 1
Default RFID Configuration in *ShakeReader*

Parameter	Status
Channel List	Channel 1, $920.625MHz$
Transmit Power	$32.5dBm$
Reader Mode	Max throughput
Search Mode	Dual Target
Sampling rate of sensor	$100Hz$

6 Implementation and Evaluation

We implement a prototype of *ShakeReader* using the COTS RFID system and conduct extensive experiments to evaluate its performance in this section.

Hardware. As shown in Fig. 16, our prototype system consists of an Impinj R420 RFID reader, which is connected to a circularly-polarized directional antenna. We adopt the Network Time Protocol (NTP) to synchronize the reader's time [20] with smartphones. Three different types of RFID tags (i.e., Impinj E53, Alien ALN-9640, and Impinj H47) are tested in our experiments. A PC with Intel Core i7-10510U 2.30GHz CPU and 16GB RAM is used as the server to control the reader and process the received RFID data. We test three popular smartphones including an iPhone 7 with aluminum back cover, a HUAWEI P20 Pro with glass back cover, and an iPhone 7 with a common soft rubber case.

Data Collection. Our server adopts the LLRP (Low-Level Reader Protocol) to communicate with the RFID reader and the software is implemented using C#. We use MATLAB Mobile Apps [24] to collect sensor data and the data processing algorithm is implemented using MATLAB.

Experiment Setting. We conduct experiments in an office environment with a size of $4m \times 10m$ and a bookshelf scenario in another office to evaluate the performance of *ShakeReader*. By default, the reader uses its maximum transmit power at $32.5dBm$ and works on $920.625MHz$. In our experiment, the reading rate is about 260 tags/s. On the client side, we adopt the sampling rate of $100Hz$ to collect data from the smartphone's accelerometer. The default configuration is in Table 1.

Metrics: For each component, we mainly focus on detection accuracy. We adopt three metrics, i.e., Accuracy, False Accept Rate (FAR) and False Reject Rate (FRR) to evaluate the overall performance of the system. Accuracy is defined as the rate that one tag is correctly matched to its corresponding client. FAR is

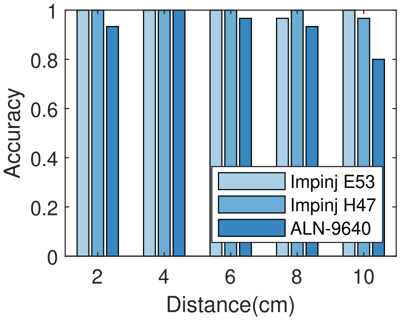


Fig. 17. Impact of smartphone-to-tag distances.

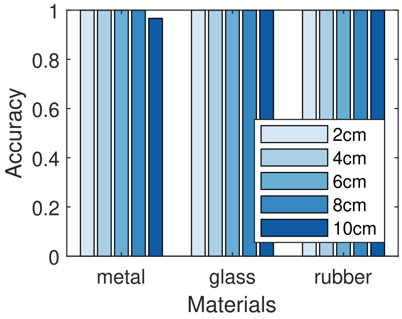


Fig. 18. Impact of different reflective materials.

the rate that *ShakeReader* incorrectly accepts the uninterested tag information and FRR is the rate that *ShakeReader* incorrectly rejects the interacted tag information.

6.1 RFID based Smartphone Gesture Detection

Component-1 detects smartphone gestures based on the phase measurements of RFID tags. In the following, we consider various factors that may affect the detection accuracy.

Impact of Smartphone-to-Tag Distance. To evaluate the effective interaction range of *ShakeReader*, we vary the smartphone-to-tag distance from $2cm$ to $10cm$. A volunteer is asked to perform the smartphone gesture 30 times at each interaction distance.

Fig. 17 shows the detection accuracy at different distances. The smartphone gestures can be detected with an average accuracy over 95 percent. In the figure, we see that within interaction distance of $10cm$, the gesture detection accuracy for Impinj E53 and Impinj H47 tags keeps stable and exceeds 95 percent at all tag-smartphone distances. The interaction with ALN-9640 tag exhibits a lower detection accuracy of around 90 percent and decreases to 80 percent at the distance of $10cm$. This is because the ALN-9640 tag is not fully covered by the smartphone, resulting in an asymmetric pattern during smartphone rotation. Therefore, we choose the Impinj E53 as our default RFID tag in the next experiments.

We note that a longer distance between the tag and the smartphone results in weaker reflected signals. As such, the smartphone may not cause sufficient impact on the backscattered signal, which degrades the detection accuracy. Therefore, to ‘read’ a tag, a user needs to make a smartphone gesture within $10cm$. More importantly, the result implies that a smartphone gesture will not cause ambiguity in identifying the interacted tags as long as the interacted ones are separated from their near tags by $10cm$. As such, we do not intend to increase smartphone-to-tag distance in the current implementation. Possible approaches to increase the distance is to increase the transmission power of readers, and decrease the distance between antenna and smartphone, thereby increasing reflected signals from smartphones.

Impact of Smartphone Materials. Different smartphones may have different back cover materials. The reflected signal is impacted by the reflection coefficient of the material. A higher reflection coefficient of the reflector can reflect more radio waves. To test the impact of smartphone materials, we conduct an experiment using 3 smartphones with different materials: an iPhone 7 with metal back cover, a HUAWEI P20 Pro with a glass back cover and an iPhone 7 with a soft rubber case. A volunteer performs the predefined smartphone gesture at $10cm$ interaction distance. Each phone is used to interact with 3 different tags 30 times.

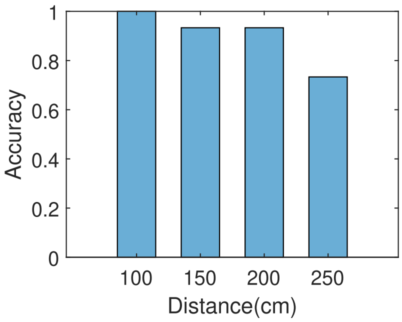


Fig. 19. Impact of tag-to-reader distances.

Fig. 18 shows gesture detection accuracy when using smartphones with different materials to interact with the tag. We observe that almost all the gestures performed using smartphones with different back cover materials can be detected. We note that along with the external back cover, the internal circuit board also reflects continuous waves to the tags. As such, smartphones with glass and rubber back cover can also be used to interact with tags.

Impact of Tag-to-Reader Distance. In the above experiments, we fixed the distance between the tag and the reader’s antenna at $1m$. To evaluate the impact of distance between the tag and the reader’s antenna, we vary the tag-reader distance ranging from $1m$ to $2.5m$. A volunteer is asked to perform the smartphone gesture 100 times in front of the tag while the tag-reader distance is varied. In the experiment, we only use the Impinj E53 tag and the interaction distance between the tag and the smartphone is within $10cm$.

Fig. 19 illustrates the gesture detection accuracy at different tag-reader distances. When the tag-reader distance is $1m$, the RFID system can reliably measure the changes in backscattered signal and our algorithm can correctly detect almost all gestures. As the tag-reader distance increases to $2.5m$, the backscattered signal becomes weak, resulting in miss detection of some gestures. In practice, one COTS reader can be connected with multiple antennas. To achieve high detection accuracy, we can deploy multiple antennas to reduce tag-to-reader distance.

Impact of Tag-to-Tag Distance. When a user is interacting with the tag of interest, the adjacent tags may be affected as well, leading to detection ambiguity. To evaluate the impact of tag-to-tag distance, we fix the tag-reader distance to $1m$

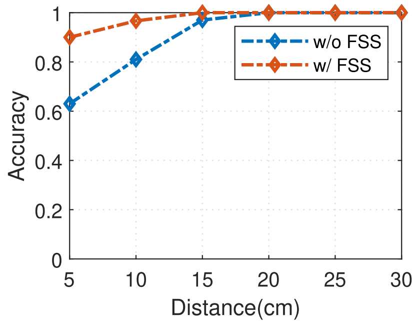
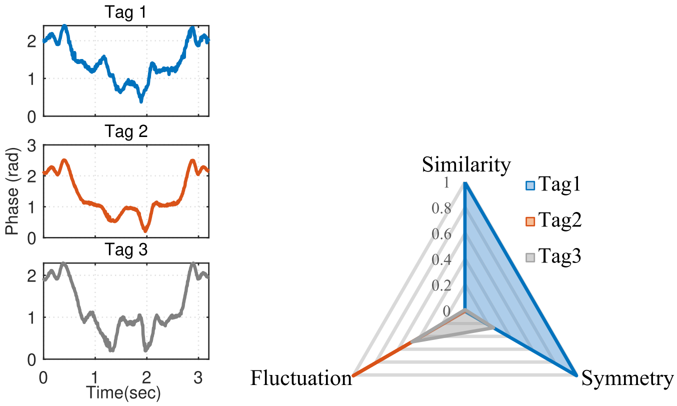


Fig. 20. Impact of tag-to-tag distances.



(a) Phase changes (b) Three metrics based on FSS algorithm

Fig. 21. Both the interacted tag and its adjacent tags are fully covered.

and the interaction distance within $10cm$ while varying the tag-to-tag distances from $5cm$ to $30cm$. A volunteer performs a gesture in front of one tag, while we move away the other tag from the interacted tag.

Fig. 20 plots the detection accuracy with and without the enhancement of FSS. On the one hand, we observe that when the tag-to-tag distance is $<15cm$, the adjacent tags could influence the detection. When tags are very close to each other (e.g., $<10cm$), without FSS our system sometimes detects an adjacent tag as the interacted tag. In contrast, with FSS, most interacted tags can be correctly identified. Even when the tag-to-tag distance is $5cm$, the detection accuracy reaches 90 percent with FSS, which is 37 percent higher than the result without FSS. In summary, FSS algorithm can effectively improve the detection accuracy and mitigate the impact of adjacent tags.

On the other hand, when the tag-to-tag distance exceeds $15cm$, the influence of the smartphone gesture on the adjacent tags is weaker and our system can detect almost all target tags. Therefore, we can regard $15cm$ as the *safe distance*, and tags outside the safe distance will receive limited interference. As such, when a user interacts with his interested tag, the interacted tag will not be affected by adjacent users. In practice, such requirement can be easily guaranteed because of the width of human body (average adult female shoulder width is about $40cm$).

In addition, we consider a scenario where the tag-to-tag distance is less than $5cm$ and both the interacted tag and its adjacent tags are completely covered by the smartphone gestures. We place three Impinj E53 tags in Fig. 12 vertically and keep the tag-to-tag distance is $3cm$ to conduct a corresponding experiment. Fig. 21 plots the phase measurements of a set of tags and the corresponding values of three metrics: fluctuation, symmetry, and similarity. We can see that when adjacent tags (Tag 1 and Tag 3) are fully covered by the smartphone gesture, their phase changes are almost the same as that of the actual interacted tag (Tag 2). This is because that they receive almost the same and strong reflected signal from smartphone under the coverage of smartphone gestures. In such cases, we cannot accurately identify the exact tag of interest from neighbor tags even if we utilize the proposed FSS algorithm to compare three metrics as shown in Fig. 21b. Therefore, we recommend that the setting of tag-to-tag distance should ensure that the adjacent tags are not fully covered by the smartphone.

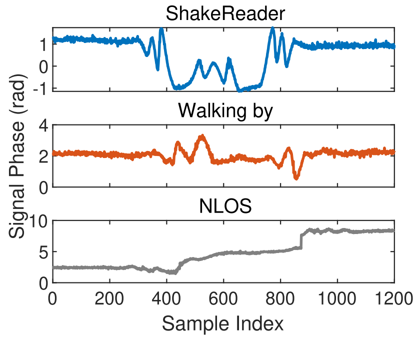


Fig. 22. Impact of human movements.

Impact of Tag Orientation. In real applications, an interested tag can be attached to an item in various orientations. To investigate the impact of tag orientation relative to the smartphone, we vary the tag's orientation θ from 0° to 180° as shown in Fig. 5. We perform the pre-defined gesture 30 times at each tag's orientation and measure recognition accuracy. In the experiment, the smartphone rotates in the XY plane, while the tag's initial orientation attached to the item is varied as illustrated in Fig. 5. According to our experiments, the tag orientation does not affect the gesture recognition accuracy. That is because we leverage the symmetry of our predefined gesture to pair the interested tag with its corresponding smartphone, which is irrelevant to the tag's initial orientation. We note that if the smartphone rotates in the XZ plane, since the reflection from the smartphone to the tag is weak due to small reflection surface, it becomes hard to notice substantial phase changes during smartphone gesture. In this case, we need to manually adjust the RFID tag to ensure that the tag plane is parallel to the smartphone.

Impact of Human Movement. Human movements near a tag may cause the change in its backscattered signal. We consider the human movement near a tag as well as the blockage of the line-of-sight path between a tag and reader's antenna by a user. In the first scenario, we ask a volunteer to walk near a tag and stay in front of the tag for a while. In the second scenario, we ask a volunteer to stand between the tag and the reader to block the line-of-sight path. Fig. 22 plots the phase measurements in the two scenarios. Compared with the pre-defined gesture of *ShakeReader*, the phase measurements in the two scenarios exhibit different patterns. Even if Component-1 accidentally triggers a false alarm and incorrectly broadcasts a potential smartphone gesture to clients, the clients can filter out the packets using

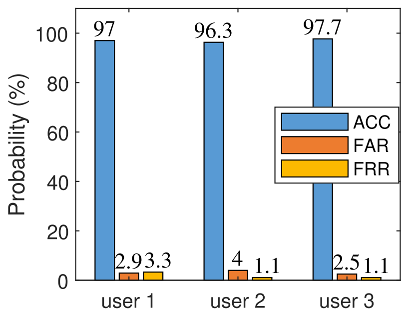


Fig. 23. Overall performance.

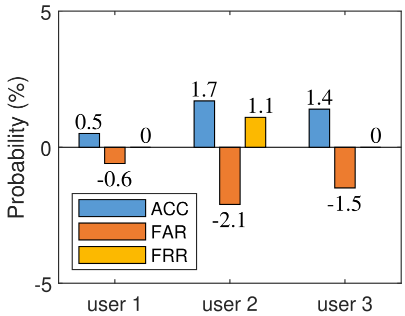


Fig. 24. Differences between with and without the FSS algorithm.

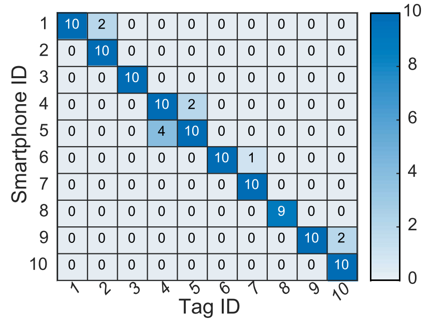


Fig. 25. Matching result between smartphones and tags in a shelf scenario.

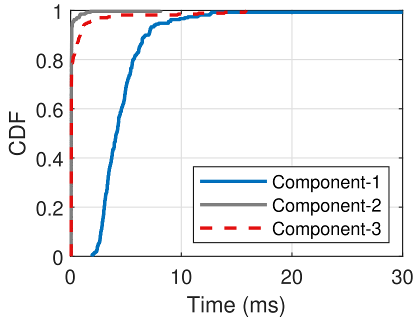


Fig. 26. Execution time.

Component-3 (i.e., synchronicity based matching and pairing). In addition, the limitation of the interaction distance between smartphones and tags (within $10cm$) prevents the interference of human movements.

6.2 Overall System Performance

System Performance in a Multi-User Scenario. To evaluate the system performance in a multi-user scenario, we invite three volunteers (2 males and 1 female) to simultaneously interact with any of the 9 tags. We note that volunteers do not interact with the same tag simultaneously, but they can interact with different tags at the same time. We conduct this experiment in an office environment and the 9 tags are attached on paper boxer separated by $15cm$ in Fig. 16. Each volunteer interacts with one of the tags within $10cm$ interaction range. We record the ground truth of the interactions and test whether our system can accurately match the interacted tags to their corresponding smartphones.

In the dynamic environment with multiple users, we collect 810 RFID tag records and 270 smartphone gesture records in total. As shown in Fig. 23, *ShakeReader* with FSS achieves the matching accuracy of $>96.3\%$. Even in the case of multi-user interaction, the FAR and FRR of each user are less than 4 and 3.3 percent respectively. The results indicate that *ShakeReader* can accurately match the interacted tags to their corresponding smartphones. In our applications, we care more about FRR than FAR, because false rejects mean a user performs the pre-defined gesture but does not receive any item information. In contrast, false accepts indicate that it is possible for a user to receive broadcast information of an uninterested tag. When two users interact with two different tags at the same time and their phase and accelerometer waveforms exhibit similar patterns, *ShakeReader* may not be able to differentiate the two gestures and associate the tags to their corresponding tags. To address this problem, we can examine tag location and phone location to further improve matching accuracy in future work.

System Performance Improvements Introduced by the FSS Algorithm. To illustrate the performance of FSS algorithm, we plot the differences of overall performance between with and without FSS algorithm in Fig. 24. We can see that with the help of the FSS algorithm, the FAR is significantly reduced and the maximum reduction is 2.1 percent. This is because the FSS algorithm considers three metrics synthetically to mitigate the influence of adjacent tags, so that more interacted tags could be pushed to the corresponding users instead of other non-interested tags. As such, the overall accuracy is correspondingly increased. Therefore, the FSS algorithm can effectively prevent the influence of adjacent tags.

System Performance in a Shelf Scenario. To simulate real application scenarios, we divide 10 items attached with RFID tags into two columns and put them on the shelf to conduct the experiment as shown in Fig. 16. The shape of selected items is various and the distance of the tag on the items is around $10cm$. A volunteer randomly chooses an item and performs the pre-defined gesture in front of the interested item. In this process, we read phase samples when performing 100 smartphone gestures in total and each tag is interacted 10 times.

Fig. 25 plots the matching result between smartphones and tags. For a pair of tags and smartphones, we set the same ID. Overall, under the shelf scenario, *ShakeReader* can effectively push the tag information to the corresponding users. The overall accuracy reaches 98.8 percent, FAR is 1.22 percent and FRR is 1 percent. We notice that 4 pieces of irrelevant information from the adjacent tag #4 are received by the smartphone #5. Although the interference of the

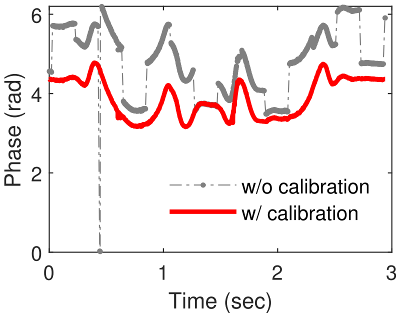
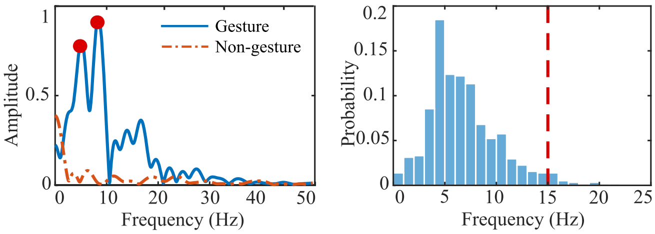
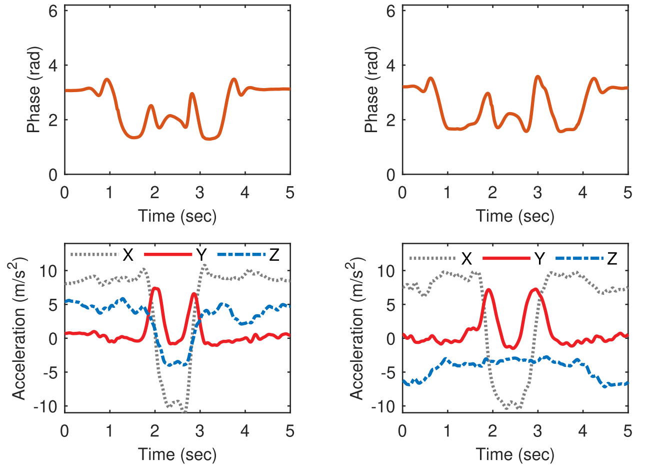


Fig. 27. Phase calibration under frequency hopping mode.



(a) After FFT (b) Frequency Distribution

Fig. 28. Gesture frequency component analysis.



(a) Tilt forward 30 degrees (b) Tilt backward 30 degrees

Fig. 29. Impact of phone tilt on tag data and sensor data.

adjacent tag #4 is strong, the interested tag #5 can still be 'read' with a very high accuracy.

System Latency. We measure the execution time of each component as shown in Fig. 26. The average values are around $4.83ms$, $0.13ms$, and $0.48ms$ for Component-1, Component-2 and Component-3, respectively. We find that the DTW algorithm in Component-1 is most time-consuming. To reduce the time complexity, instead of scanning all sampling points of tag signals, we select the segments between the local maximums and local minimums to execute the DTW algorithm to find the symmetric point. In addition, our system matches interacted tags and corresponding users using timing information rather than raw data, which further reduces computational complexity. Overall, the average processing time of *ShakeReader* is $7.6ms$ for each smartphone gesture matching, which is acceptable for most interaction applications.

System Generalization. For the countries that adopt Frequency Hopping Spread Spectrum (FHSS), commercial RFID readers must run in the frequency-hopping mode to reduce co-channel interference, which will cause phase discontinuity and impact our system performance. To address this issue, we first conduct a phase calibration step to map different hopping frequencies to a single fixed frequency as described in [38], [40]. Fig. 27 plots the results after phase calibration. The grey line indicates the phase measurements caused by the smartphone gesture in the Hong Kong frequency-hopping mode. According to the regulation of Hong Kong Office of the Telecommunications Authority (OFTA), commercial UHF RFID readers must randomly hop to one of 10 center frequencies within the 920-925 MHz band every 200 ms. Therefore, we can see that the phase measurements collected directly from the RFID reader are discontinuous. After applying the phase calibration, the phase measurements show the continuous pattern. As a result, we can use phase calibration to generalize our system and support frequency hopping.

System Capacity. A low reading rate of reader will result in a low resolution of measured timing information extracted from RFID data, which may affect the matching accuracy. To determine the maximum capacity of *ShakeReader*, we first analyze the frequency component of the predefined interaction gestures with different users. We use the Fast Fourier Transform (FFT) to measure the frequency domain information of RFID data when users perform gestures as shown in Fig. 28a. We can see that the main frequency components corresponding to the gestures are concentrated below $20Hz$. Thus, we plot the top-2 frequency distribution from 370 RFID tag records of four users in Fig. 28b. We can see that 96.8 percent of gesture frequencies is less than $15Hz$. According to the Nyquist theorem, the reading rate of the RFID reader needs to be higher than 30 readings/s for a single tag. As a result, we can utilize the adaptive reading scheme in Section. 5.2 to improve the reading rate of target tags to meet this requirement.

7 Discussion

In this section, we discuss limitations of *ShakeReader* and room for improvement.

Design of Interactive Smartphone Gestures. Based on our proposed reflector polarization model, we have carefully defined our interactive smartphone gesture in Section 4.1. In practice, these pre-defined rules are flexible and users do not need to follow them strictly. For example, Fig. 29 illustrates the impact of phone tilt on tag data and sensor data during interaction. We can see that tilting smartphone forward 30 degrees (Fig. 29a) or backward 30 degrees (Fig. 29b) relative to the tag will still produce the specific smartphone gesture patterns as expected. As a result, the tilt of smartphones does not greatly affect our Component-1) RFID based smartphone gesture detection. However, there are slight differences in the changes of smartphone sensor data. The Z-axis acceleration readings no longer change from 0 due to the influence of gravity. Fortunately, our sensor based smartphone gesture detection mainly relies on the standard deviations of Z-axis acceleration readings, which are independent of the initial state. In addition, the acceleration readings of X-axis and Y-axis are almost unaffected when the phone is tilted to interact with the tag. Thus, our system tolerate slight smartphone tilt when users interact with tags.

Tag-to-Tag Distance. Based on our experimental results in Figs. 20 and 21, we recommend that the setting of tag-to-tag distance should exceed $5cm$ to ensure that the adjacent tags are not fully covered by the smartphone gesture. However, in practice, tag-to-tag distance may not be guaranteed. To mitigate the impact of neighbor tags, we may broadcast both the information of interacted tag and neighbor tags to the user. Then the user can perform double-check and pick the interacted tag. Besides, as a workaround, users can also pick up the interested product and make sure the to-be-interacted tag is sufficiently separated from other tags before performing a smartphone gesture.

Tag-to-Reader Distance. In our system, we need to control the distance between tag and reader to ensure the detectability of the backscatter signal. If the tag-to-reader distance is too large, the backscatter signal becomes too weak to be accurately detected. Based on our experimental results, we suggest the tag-to-reader distance should be within 2 meters. In practice, a commercial RFID reader can be connected to multiple antennas. For example, the Impinj R420 reader has 4 antenna ports, which can be further extended to connect up to 32 antennas with an antenna hub [16]. Therefore, we can deploy multiple antennas to ensure the coverage of RFID tags.

System Cost. In this work, we utilize ubiquitous smartphones to enable a flexible human-RFID interaction without making any hardware extension to either deployed RFID infrastructure or smartphones. Compared with traditional solutions with external UHF modules, we indeed increase the cost of server and wireless network deployment and power consumption, as our system requires users to connect to the server through a wireless network to receive the broadcast tag information. *ShakeReader* adds a new function to the smartphones that allow smartphones to 'read' RFID tags without any hardware modification or extension. As such, ordinary users in the logistics and retail industry can use their smartphones to query the item-specific information stored in RFID tag instead of using expensive specialized equipment (e.g., handheld RFID readers).

Privacy Issue and System Security. ShakeReader leverages the synchronicity of the changes in RFID data and smartphone sensor data simultaneously caused by a smartphone gesture to receive the interested tag information over a wireless network. In this process, sensor data from users' smartphones is recorded locally and the smartphones connect to the server to obtain the tag information, which may raise privacy concerns. In practice, synchronicity based matching and pairing (Component-3) can run on the client side. In this way, clients keep sensor data local, and receive and match the broadcast messages encapsulating the tag information from the server. We note that clients do not need to send any data to a server during the interaction process, meaning that the sensor data that could potentially reveal a user's privacy would not leave the user's smartphone.

Leveraging Tag and Smartphone Localization. RFID and smartphone localization have been extensively studied in previous works. Some works can achieve very high localization accuracy with calibration and fingerprinting. Our original idea was to locate both tag and smartphone in the environment and pair collocated tag and smartphone. However, it turns out such an approach requires highly accurate localization performance (e.g., with localization error $< 15cm$), which is very challenging to achieve in practical scenarios. *ShakeReader* can be optimized if there are pre-deployed RFID or smartphone localization systems that can ensure high localization accuracy. However, it is worth noting that *ShakeReader* does not rely on any deployed localization services.

8 Related Work

ShakeReader is related to past works in the following three areas: UHF RFID identification with smartphones, RFID-based human-object interaction, and contact-free human-computer interaction. To the best of our knowledge, *ShakeReader* is the first work that enables commodity smartphones to 'read' passive RFID tags without any hardware modification to either smartphones or RFID readers.

Reading UHF RFID Using Smartphone. Most commercial smartphones available on the market cannot directly read UHF RFID tags. In order to read UHF RFID tags, one may extend smartphone by adding external UHF modules [2], which incurs extra cost and power consumption to smartphones. Recent research aims to allow smartphone users to read UHF RFID tags using Cross-Frequency Communication technologies. For example, TiFi [3] first reads RFID tags using RFID readers and broadcasts the tag IDs as Wi-Fi beacons. However, the signal strength based association in TiFi is subject to background noise and interference. In addition, it is very challenging to correctly identify the interested tag among all tag IDs. Unlike the previous work, our work uses a pre-defined smartphone gesture and leverages the synchronicity of RFID and sensor data to accurately match an interacted tag to the corresponding smartphone.

RFID-Based Human-Object Interaction. Human-object interaction based on passive RFID has attracted much attention in recent years. COTS RFID systems have been used to achieve high accuracy in tracking RFID-labelled objects [4], [10], [23], [25], [27], [31], [32], [33], [34], [39], [41], [42] and enable innovative RFID sensing applications [6], [8], [13], [36], [43], [44]. RF-IDraw [35] tracks the trajectory of an RFID tag by measuring the angle of arrival using customized antenna arrays. Tagyro [40] attaches RFID tags to an object and measures the object orientation by leveraging the polarity of tag antenna. PolarDraw [26] infers the orientation and position of RFID-labelled items based on tag polarization. Spin-Antenna [30] enhances object tracking accuracy by combing tag arrays and spinning polarized antenna, which can effectively suppress ambient signal interference and noise. Unlike these works, *ShakeReader* does not need to attach tags to smartphones. Instead, *ShakeReader* detects the symmetric smartphone rotation by leveraging the polarization of the reflected signal.

Contact-Free Human-Computer Interaction. Recent work explores the possibility of RFID sensing without attaching tags directly to target objects. ShopMiner [45] mines customer shopping behavior by analyzing the backscatter signal using deployed RFID infrastructure in stores. TagFree [11] recognizes various human activities by analyzing the multipath signals using deep neural networks. TACT [38] builds a contact-free reflection model for activity recognition without attaching tags to users. RFIPad [9] enables in-air handwriting using an array of RFID tags. RFIPad does not require users to carry any RFID tags. RF-finger [29] tracks finger writings and recognizes multi-touch gestures using tag arrays deployed in the environment. LungTrack [5] builds a contactless respiration monitoring system and minimizes dead zones to facilitate accurate respiration sensing. LungTrack can support the simultaneous monitoring of two human targets. TagSleep [21] proposes a contactless sleep sound-activity recognition system, which can sense snore, cough, respiration and sleep postures. Au-Id [14] captures the human's physical and behavioral features from RFID data to build a non-intrusive automatic user identification and authentication system. RFID light bulbs [12] enable various home-scale interactions, including infrastructure monitoring, location and guided navigation. The contact-free reflection model shows that a moving object in the environment can cause changes to the backscatter channel between the tag and reader antenna, which can be captured by RFID systems. These works largely overlook the polarization of reflected signals from objects, mainly because the objects do not rotate in the target applications. To the best of our knowledge, *ShakeReader* is the first work that models reflector polarization in the contact-free smartphone gesture detection using RFID systems.

9 Conclusion

In this paper, we aim to enable smartphone users to interact with UHF RFID tags using their smartphones without making any hardware extension to either deployed RFID infrastructure or smartphones. To this end, we define a smartphone gesture which can be simultaneously detected by both RFID systems and smartphones. We overcome many technical challenges involved in smartphone gesture detection especially using RFID systems. In particular, we characterize the polarization of reflected signals from smartphone and detect smartphone rotations. We leverage the synchronicity of RFID data and sensor data caused by the same smartphone gesture to match the interacted tag with the corresponding smartphone. Experimental results show that *ShakeReader* can achieve up to 96.3 percent matching accuracy.

Acknowledgments

This work was supported in part by the National Nature Science Foundation of China under Grants 61702437 and 61872285, in part by Hong Kong GRF under Grant PolyU 152165/19E, in part by the major project of the National Social Science Foundation under Grant 20ZDA062, in part by Research Institute of Cyberspace Governance, Zhejiang University, Leading Innovative and Entrepreneur Team Introduction Program of Zhejiang under Grant 2018R01005, in part by Zhejiang Key R&D Plan under Grant 2019C03133, and in part by the Fundamental Research Funds for Central Universities under Grant 531118010612.

References

[1] S. A. Ahson and M. Ilyas, *RFID HANDBOOK: Applications, Technology, Security, and Privacy*. Boca Raton, FL, USA: CRC Press, 2017.

[2] Amazon. "RFID ME: MINI ME UHF RFID reader for android powered devices," 2019. [Online]. Available: https://www.amazon.co.jp/-/en/RFID-ME-Android-Powered-Devices/dp/B0178WLGPK

[3] Z. An, Q. Lin, and L. Yang, "Near-field identification of UHF RFIDs with WiFi!" in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 826–828.

[4] Y. Bu *et al.*, "RF-3DScan: RFID-based 3D reconstruction on tagged packages," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 722–738, Feb. 2021.

[5] L. Chen *et al.*, "LungTrack: Towards contactless and zero dead-zone respiration monitoring with commodity RFIDs," *ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 3, no. 3, pp. 1–22, 2019.

[6] Z. Chen, P. Yang, J. Xiong, Y. Feng, and X.-Y. Li, "TagRay: Contactless sensing and tracking of mobile objects using COTS RFID devices," in *Proc. IEEE INFOCOM, IEEE Conf. Comput. Commun.*, 2020, pp. 307–316.

[7] K. Cui, Y. Wang, Y. Zheng, and J. Han, "ShakeReader: 'Read' UHF RFID using Smartphone," in *Proc. IEEE INFOCOM*, 2021.

[8] H. Ding *et al.*, "A platform for free-weight exercise monitoring with passive tags," *IEEE Trans. Mobile Comput.*, vol. 16, no. 12, pp. 3279–3293, Dec. 2017.

[9] H. Ding *et al.*, "RFIPad: Enabling cost-efficient and device-free in-air handwriting using passive tags," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 447–457.

[10] C. Duan, W. Shi, F. Dang, and X. Ding, "Enabling RFID-based tracking for multi-objects with visual aids: A calibration-free solution," in *Proc. IEEE INFOCOM - IEEE Conf. Comput. Commun.*, 2020, pp. 1281–1290.

[11] X. Fan, W. Gong, and J. Liu, "TagFree activity identification with RFIDs," *ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 7:1–7:23, 2018.

[12] J. Gummeson, J. Mccann, C. Yang, D. Ranasinghe, S. Hudson, and A. Sample, "RFID light bulb: Enabling ubiquitous deployment of interactive RFID systems," *ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 2, pp. 1–16, 2017.

[13] J. Guo, T. Wang, Y. He, M. Jin, C. Jiang, and Y. Liu, "Twinleak: RFID-based liquid leakage detection in industrial environments," in *Proc. IEEE INFOCOM - IEEE Conf. Comput. Commun*, 2019, pp. 883–891.

[14] A. Huang, D. Wang, R. Zhao, and Q. Zhang, "Au-Id: Automatic user identification and authentication through the motions captured from sequential human activities using RFID," *ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 3, no. 2, pp. 1–26, 2019.

[15] HUAYUAN, "UNIQLO global stores applied RFID tags," 2020. [Online]. Available: https://www.huayuansh.com/uniqlo-global-stores-applied-rfid-tags/

[16] M. Lenehan. "Impinj antenna hubs product briefs," 2021. [Online]. Available: https://support.impinj.com/hc/en-us/articles/202755698-Impinj-Antenna-Hubs-Product-Briefs

[17] C. Jiang, Y. He, X. Zheng, and Y. Liu, "OmniTrack: Orientation-aware RFID tracking with centimeter-level accuracy," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 634–646, Feb. 2021.

[18] C. Jiang, Y. He, X. Zheng, and Y. Liu, "Orientation-aware RFID tracking with centimeter-level accuracy," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 634–646, Feb. 2021.

[19] N. Kumar "Zara: Fast fashion and RFID," 2018. [Online]. Available: https://nirmalyakumar.com/2018/06/02/zara-fast-fashion-and-rfid/

[20] M. Lenehan "Synchronize and set the clock on speedway RAIN RFID readers," 2021. [Online]. Available: https://support.impinj.com/hc/en-us/articles/202756558-Synchronize-and-Set-the-Clock-on-Speedway-RAIN-RFIDReaders/

[21] C. Liu, J. Xiong, L. Cai, L. Feng, X. Chen, and D. Fang, "Beyond respiration: Contactless sleep sound-activity recognition using RF signals," *ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 3, no. 3, pp. 1–22, 2019.

[22] J. Liu, S. Chen, M. Chen, Q. Xiao, and L. Chen, "Pose sensing with a single RFID tag," *IEEE/ACM Trans. Netw.*, vol. 28, no. 5, pp. 2023–2036, Oct. 2020.

[23] Z. Liu, X. Liu, and K. Li, "Deeper exercise monitoring for smart gym using fused RFID and CV data," in *Proc. IEEE INFOCOM - IEEE Conf. Comput. Commun.*, 2020, pp. 11–19.

[24] MATLAB. "MATLAB mobile APP," 2020. [Online]. Available: https://www.mathworks.com/products/matlab-mobile.html

[25] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," in *Proc. 1st IEEE Int. Conf. Pervasive Comput. Commun.*, 2003, pp. 407–415.

[26] L. Shangguan and K. Jamieson, "Leveraging electromagnetic polarization in a two-antenna whiteboard in the air," in *Proc. 12th Int. Conf. Emerg. Netw. Exp. Technol.*, 2016, pp. 443–456.

[27] X. Shi *et al.*, "TagAttention: Mobile object tracing without object appearance information by vision-RFID fusion," in *Proc. IEEE 27th Int. Conf. Netw. Protocols*, 2019, pp. 1–11.

[28] W. L. Stutzman, *Polarization in Electromagnetic Systems*. Norwood, MA, USA: Artech House, 2018.

[29] C. Wang *et al.*, "Multi-touch in the air: Device-free finger tracking and gesture recognition via COTS RFID," in *Proc. IEEE INFOCOM - IEEE Conf. Comput. Commun.*, 2018, pp. 1691–1699.

[30] C. Wang, L. Xie, K. Zhang, W. Wang, Y. Bu, and S. Lu, "Spin-antenna: 3D Motion tracking for tag array labeled objects via spinning antenna," in *Proc. IEEE INFOCOM - IEEE Conf. Comput. Commun*, 2019, pp. 1–9.

[31] G. Wang *et al.*, "HMO: Ordering RFID tags with static devices in mobile environments," *IEEE Trans. Mobile Comput.*, vol. 19, no. 1, pp. 74–89, Jan. 2020.

[32] G. Wang *et al.*, "A universal method to combat multipaths for RFID sensing," in *Proc. IEEE INFOCOM - IEEE Conf. Comput. Commun.*, 2020, pp. 277–286.

[33] J. Wang, J. Xiong, X. Chen, H. Jiang, R. K. Balan, and D. Fang, "Simultaneous material identification and target imaging with commodity RFID devices," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 739–753, Feb. 2021.

[34] J. Wang and D. Katabi, "Dude, where's my card?: RFID positioning that works with multipath and non-line of sight," in *Proc. ACM SIGCOMM Conf. SIGCOMM*, 2013, pp. 51–62.

[35] J. Wang, D. Vasisht, and D. Katabi, "RF-IDraw: Virtual touch screen in the air using RF signals," in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, 2014, pp. 235–246.

[36] Y. Wang and Y. Zheng, "TagBreathe: Monitor breathing with commodity RFID systems," *IEEE Trans. Mobile Comput.*, vol. 19, no. 4, pp. 969–981, Apr. 2020.

[37] Y. Wang, J. Shen, and Y. Zheng, "Push the limit of acoustic gesture recognition," in *IEEE INFOCOM - IEEE Conf. Comput. Commun.*, 2020, pp. 566–575.

[38] Y. Wang and Y. Zheng, "Modeling RFID signal reflection for contact-free activity recognition," *ACM Interactice Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 4, pp. 193:1–193:22, 2018.

[39] Z. Wang, M. Xu, N. Ye, R. Wang, and H. Huang, "RF-Focus: Computer vision-assisted region-of-interest RFID tag recognition and localization in multipath-prevalent environments," *ACM Interactice Mobile Wearable Ubiquitous Technol.*, vol. 3, no. 1, pp. 1–30, 2019.

[40] T. Wei and X. Zhang, "Gyro in the air: Tracking 3D orientation of batteryless Internet of Things," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 55–68.

[41] H. Xu, D. Wang, R. Zhao, and Q. Zhang, "AdaRF: Adaptive RFID-based indoor localization using deep learning enhanced holography," *ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 3, no. 3, pp. 1–22, 2019.

[42] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu, "Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 237–248.

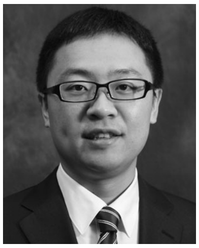
[43] L. Yang, Q. Lin, X. Li, T. Liu, and Y. Liu, "See through walls with COTS RFID system!" in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 487–499.

[44] P. Yang, Y. Feng, J. Xiong, Z. Chen, and X.-Y. Li, "RF-ear: Contactless multi-device vibration sensing and identification using COTS RFID," in *Proc. IEEE INFOCOM - IEEE Conf. Comput. Commun.*, 2020, pp. 297–306.

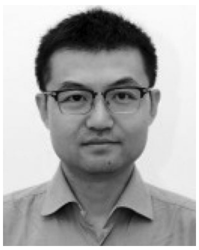
[45] Z. Zhou, L. Shangguan, X. Zheng, L. Yang, and Y. Liu, "Design and implementation of an RFID-based customer shopping behavior mining system," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2405–2418, Aug. 2017.

Kaiyan Cui (Student Member, IEEE) received the BS degree from the Taiyuan University of Technology in 2016. She is currently working toward the joint PhD degree with Xi'an Jiaotong University and The Hong Kong Polytechnic University. Her research interests include RFID, mobile computing, and smart sensing. She is a student member of the ACM.

Yanwen Wang (Member, IEEE) received the BS degree in electronic engineering from Hunan University, Changsha, China, and the MS degree in electrical engineering from the Missouri University of Science and Technology, Rolla, MO, USA, in 2010 and 2013, respectively, and the PhD degree from the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China, in 2020. He is currently an associate professor with the College of Electrical and Information Engineering, Hunan University, China. His research interests include mobile and network computing, RFID systems, and wireless sensing. He is a member of the ACM.



Yuanqing Zheng (Member, IEEE) received the BS degree in electrical engineering and the ME degree in communication and information system from Beijing Normal University, Beijing, China, in 2007 and 2010, respectively, and the PhD degree from the School of Computer Engineering, Nanyang Technological University, in 2014. He is currently an associate professor with the Department of Computing, The Hong Kong Polytechnic University. His research interests include wireless networking and mobile computing, acoustic and RF sensing, and Internet of Things (IoT). He is a member of the ACM.



Jinsong Han (Senior Member, IEEE) received the PhD degree in computer science from the Hong Kong University of Science and Technology in 2007. He is currently a professor with the School of Cyber Science and Technology, Zhejiang University. His research interests include IoT security, smart sensing, and wireless and mobile computing. He is a senior member of the ACM.