PriWhisper: Enabling Keyless Secure Acoustic Communication for Smartphones

Bingsheng Zhang, Qin Zhan, *Student Member*, *IEEE*, Si Chen, *Student Member*, *IEEE*, Muyuan Li, *Student Member*, *IEEE*, Kui Ren, *Senior Member*, *IEEE*, Cong Wang, *Member*, *IEEE*, and Di Ma, *Member*, *IEEE*

***Abstract*—Short-range wireless communication technologies have been used in many security-sensitive smartphone applications and services such as contactless micro payment and device pairing. Typically, the data confidentiality of the existing short-range communication systems relies on so-called "key-exchange then encryption" mechanism, which is inefficient, especially for short communication sessions. In this work, we present PriWhisper—a keyless secure acoustic short-range communication system for smartphones. It is designed to provide a software-based solution to secure smartphone communication without the key agreement phase. PriWhisper adopts the emerging friendly jamming technique from radio communication for data confidentiality. The system prototype is implemented and evaluated on several Android smartphone platforms for efficiency and usability. We theoretically and experimentally analyze the security of our proposed acoustic communication system against eavesdropping. In particular, we study the (in)separability of the data signal and jamming signal against *blind signal segmentation* (BSS) attacks such as *independent component analysis* (ICA). The result shows that PriWhisper provides sufficient security guarantees for commercial smartphone applications and yet strong compatibilities with most legacy smartphone platforms. As an application, we also develop AcousAuth—a novel smartphone-empowered system for personal authentication.**

***Index Terms*—Acoustic short-range communication, blind signal segmentation (BSS), independent component analysis (ICA), Internet of things, out-of-band (OOB) channel, security and privacy, smartphone wireless communication.**

I. Introduction

RECENT advancement of smartphones and tablet computing devices has witnessed the increasing popularity of short-range wireless communication in many mobile applications and services, such as mobile advertisement, contactless mobile payment, device pairing, etc. For instance, Near Field Communication (NFC) enables a low-power radio communication between two NFC-enabled devices by a simple touch. Such technology has been utilized by Google Wallet [1], which allows a smartphone user to store his/her credit and debit cards information on Google servers and then tap his/her NFC-enabled smartphone at the specialized terminal to make convenient purchases. Meanwhile, the improvement in screen resolution of smartphones exacerbates the immigration of conventional 1-D/2-D barcode usages to mobile phone-related applications. Several e-commerce business giants, e.g., Alipay [2] and PayPal [3], have also rolled out barcode-based payment services for retail customers.

Typically, such wireless communication technology offers a low data rate ad hoc channel between two portable devices within close physical proximity. This "short-range" feature makes them ideal candidates of so-called Out-Of-Band (OOB) channels for secure device pairing, e.g., [4] and [5]. Since the two communicating devices must be within 1–2 inches, it is extremely hard for an adversary to perform Man-in-the-Middle (MitM) attacks. Therefore, they may serve as low-cost authenticated channels without resorting to a Public Key Infrastructure (PKI) or trusted third parties.

On the other hand, as most short-range wireless communication based applications are in the public area, the confidentiality of the transmitted data must be strictly guaranteed against eavesdroppers in the wild. Unfortunately, this has not been satisfactorily addressed by current short-range wireless communication technologies, e.g., barcode-based system. Due to its fundamental design principle, the visual nature of barcode-based short-range communication makes them extremely vulnerable to shoulder sniffing. The widespread of surveillance cameras in public areas makes the situation even worse. Although NFC-based short-range communication systems are believed to have better security guarantees, they are also subject to (long distance) eavesdropping [6], where the transmitted data between two ISO/IEO 14443 token-based NFC devices could be eavesdropped from 15 meters away. As a countermeasure, NFC forum proposed NFCIP-1 [7] and NFC-SEC-01 [8] specifications to enhance the data confidentiality of NFC communication. Namely, the sender and the receiver have to first utilize (elliptic curve) Diffie–Hellman key exchange protocol to set up a common secret key at the beginning of each session. However, most security-sensitive mobile applications just require very few round(s) of message exchange. Hence, the key exchange process might dominate the entire communication session. Similarly, as most barcode-based mobile applications only require a single-round barcode communication with very small amount of information, it is also very difficult to setup a secure connect or add security features without compromising the communication efficiency.

Manuscript received October 28, 2013; revised December 26, 2013; accepted December 27, 2013. Date of publication January 09, 2014; date of current version May 05, 2014. This work was supported in part by the U.S. National Science Foundation under Grants CNS-1318948 and CNS-1262275 and in part by Research Grants Council of Hong Kong under ECS Grant CityU 138513.

B. Zhang, Q. Zhan, S. Chen, M. Li, and K. Ren are with the Department of Computer Science and Engineering, The State University of New York, Buffalo, NY 14260 USA (e-mail: bzhang26@buffalo.edu; zhanqin@buffalo.edu; schen23@buffalo.edu; muyuanli@buffalo.edu; kuiren@buffalo.edu).

C. Wang is with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong (e-mail: congwang@cityu.edu.hk).

D. Ma is with the Department of Computer and Information Science, University of Michigan-Dearborn, Dearborn, MI 48128 USA (e-mail: dmadma@umich.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JIOT.2014.2297998

In recognizing these design challenges, in this work, we initiate the research endeavor to investigate a novel secure keyless short-range communication system, named PriWhisper, for smartphones. Different from aforementioned barcode and NFC technologies, PriWhisper is based on aerial acoustic communication, which is traditionally used in many underwater wireless communication scenarios, e.g., [9]–[11]. Here, we explore the unique properties of aerial acoustic communication to provide PriWhisper with a number of highly desirable features as well as clearly defined security strength. First of all, the transmission of acoustic signal does not require line-of-sight, which offers PriWhisper much higher usability than the barcode-based communication systems. Second, the computational power of most smartphones are sufficient to modulate/demodulate acoustic signals using a software acoustic modem; therefore, such acoustic communication systems can be easily deployed on most off-the-shelf smartphone platforms. Unlike NFC chips, it is safe to assume that all current smartphones are readily equipped with a speaker and microphone as required by the functionality of phones. Third, sound wave has inherent localization in the air medium, and it fades quickly when travels in distance. As a coin has two sides, this "terrible" feature naturally enhances the data confidentiality of acoustic communication systems against eavesdropping. Finally, when the carrier frequency of a smartphone acoustic communication system lies within audible bandwidth, it is easy to detect jamming like DoS attacks and locate the adversaries by human ears.

To achieve keyless secure communication, we adopt the friendly jamming technique [12] from radio communication. In a nutshell, the friendly jamming technique lets the receiver transmit a random jamming signal (artificial noise) while the sender is transmitting the data signal. Hence, nobody else can decode the recorded noisy signal except the receiver who knows its own jamming signal and thus can easily remove it from the received mixture signal. To deploy friendly jamming technique on a single-device receiver, it requires the receiver to have a full-duplex channel for simultaneous sending and receiving. This is a crucial reason why we choose aerial acoustic channel as a candidate. To the best of our knowledge, acoustic channel is the only full-duplex channel that we can control freely from smartphone Operating System Application Programming Interface (OS API). Namely, almost every smartphone can use its microphone and speaker simultaneously. We note that a NFC tag can only receive or send a signal, while the interrogating device can receive a signal at the same time when it sends a command. Therefore, NFC does not support sending and receiving data simultaneously with existing smartphone OS APIs, say Android 4.x series. Below, we summarize our contributions.

1) We design and implement PriWhisper—a secure keyless acoustic short-range communication system that exploits friendly jamming technique from radio communication for data confidentiality. To the best of our knowledge, it is the first work on extending friendly jamming technique to aerial acoustic communication system for smartphones.
2) We analytically and experimentally examine the security level of PriWhisper, especially in the presence of multiple-sensor eavesdroppers. In particular, we show that the adversary cannot separate the data signal and jamming signal even with multiple sensors using blind signal segmentation (BSS) technique in the recommended PriWhisper working scenarios, where the speakers of the two communicating smartphones are very close to each other.
3) We demonstrate PriWhisper has high efficiency, compatibility, and usability through system prototyping and evaluation on several Android smartphone platforms. The throughput of current prototype can reach approximately 1000 bps, which is sufficient for most security-sensitive smartphone applications. We also develop the AcousAuth system as a useful application.

The rest of this paper is organised as follows. Section II introduces our system architecture, threat model, and technical background. In Section III, we present the PriWhisper system design. In Section IV, we implement the proposed system and test its performance. We thoroughly analyze the security of our proposed system in Section V. In Section VI, we present the AcousAuth system as an application. Finally, Section VII summarizes related work, and a conclusion is given in Section VIII.

II. Preliminaries

A. System Architecture

PriWhisper is designed to enable keyless secure acoustic short-range communication in both smartphone–smartphone and smartphone–terminal scenarios. Without loss of generality, our system prototype is implemented in the smartphone environment, but it is straightforward to make it support point-of-sale (POS) terminals. PriWhisper is purely realized by software that offers great compatibilities to various smartphone platforms without additional hardware requirement. Any smartphone armed with a microphone and speaker is a potential host of PriWhisper. Similar to NFC, the communication can be achieved by a simple touch. PriWhisper automatically initializes the keyless acoustic communication when two smartphones (or a smartphone and a POS terminal) are close to each other. All the users need to do is simply tapping their devices together face-to-face, and the data transmission process is triggered by the proximity sensors. The designed working distance of PriWhisper is more flexible than that of an NFC system, but it is recommended to be less than 0.5 cm for better security guarantees. As depicted in Fig. 1, both the sender and the receiver play audible acoustic signals during a secure communication, and the communication session length is only 1–2 s for most smartphone applications.

B. Security Goal and Threat Model

PriWhisper is expected to provide secure communication in the presence of both passive eavesdroppers, and its security against active adversaries can be enhanced with additional security mechanism but it is orthogonal to this work. The system security is analyzed in the standard *line-of-sight* (LOS) channel model, where the channel h is approximated by the frequency-selective fading function $p(f_c, \ell)$, in which f_c is the carrier

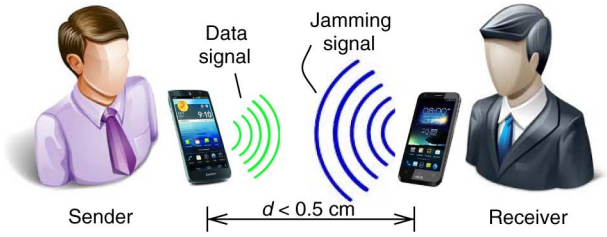

Fig. 1. System architecture.

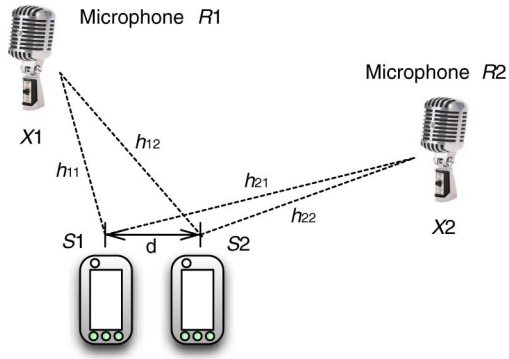

Fig. 2. Illustrative attack scenario where the eavesdropper uses two microphones.

frequency and ℓ is the distance parameter. A similar channel model can be found in [13], on which the channel model assumes that all transmitted signals experience the same channel condition through one path. PriWhisper is designed to protect confidentiality of the transmitted data against single/multiple-sensor eavesdroppers. In particular, multiple-sensor eavesdroppers may try to separate the data signal from his/her recorded mixture signals. The eavesdroppers are allowed to place their sensors (microphones) at any fixed locations in priori to the acoustic short-range communication. Fig. 2 illustrates an attack scenario where the adversary utilizes two microphones R_1 and R_2 for eavesdropping. Let s_1 and s_2 be the two acoustic signal sources and denote the mixture signals received by R_1 and R_2 as x_1 and x_2, respectively. Assume that the signal x_1 received by microphone R_1 is a (linear) mixture of $h_{11}s_1$ and $h_{12}s_2$. Denote the eavesdropper's recorded mixture signals as the vector $\boldsymbol{x} = [x_1, x_2]^T$, which can be expressed as

$$\boldsymbol{x} = \boldsymbol{H} \cdot \boldsymbol{s} + \boldsymbol{e} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \cdot \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$$

where $\boldsymbol{H}$ is the channel *mixing matrix* and $\boldsymbol{e}$ is a random channel noise vector.

A separation attack consists of two phases: online phase and offline phase. In the online phase, the mixture signals are collected by the adversary's multiple microphones through the air medium, and they are assumed to be $\mathbf{x}(t) = \boldsymbol{H} \cdot \boldsymbol{s}(t) + \boldsymbol{e}(t)$, where t is the time index and $\boldsymbol{H}$ is unknown and need to be solved by the adversary. In the offline phase, the adversary tries to estimate the data signal and jamming signal using BSS techniques such as independent component analysis (ICA). Upon success, the adversary can recover the transmitted data from the estimated data signal.

C. Blind Signal Segmentation

BSS techniques aim to separate several simultaneously active source signals from a set of mixed signals without any additional knowledge of the source signals. Typical BSS techniques are based on the assumption that all signal sources are static points, because most BSS algorithms require the stationarity of mixing matrix. The mixing process consists of a linear time-invariant filtering of the source signals. To separate the source signals, the mixture signals are studied to obtain an optimal estimation of each source signal with the best possible quality.

There are many BSS algorithms in the literature. Most of them assume that the number of recorded mixture signals (by distinct sensors) is the same as the number of signal sources, which is also known as well-determined or complete BSS. When the number of recorded mixture signals is more than the signal sources, such BSS algorithms are referred as overdetermined or under-complete BSS [14]. The signal mixtures are generally separated by multichannel time-invariant filtering, and the algorithms try to eliminate influence of certain spatial directions by applying linear de-mixing filters [15]. ICA is one of the most famous algorithms to solve well-determined and overdetermined BSS. A classic ICA approach estimates the de-mixing filters by assuming that the source signals are independent and non-Gaussian [16] or Gaussian with a nonstationary variance [13].

On the flip side, when there are less sensors than sources, the kind of BSS problems are called the under-determined or over-complete BSS. Many under-determined BSS approaches rely on more complex source models that assume a certain requirement on a specific source [17]. Under-determined mixture signals are usually separated by time-frequency masking methods [18], [19], which eliminate interference in certain time-frequency points.

D. ICA Technique Overview

ICA is one of the most successful BSS techniques. The goal of ICA is to find a linear representation of non-Gaussian signals so that the components are as statistically independent as possible, e.g., the two-sensor eavesdropping scenario. Let s_1 and s_2 be the two signal sources, and each microphone R_i records a composite signal x_i, consisting of s_1 and s_2 signal components. Due to the difference in distance of these two microphones, x_1 and x_2 have different relative component offsets between s_1 and s_2. For simplicity, we omit any time delays and the channel noise, so we have the mixing model

$$x_1 = h_{11}s_1 + h_{12}s_2$$
$$x_2 = h_{21}s_1 + h_{22}s_2.$$

Given x_1 and x_2, ICA employs information theoretic principles to find an unmixing matrix $\boldsymbol{W}$ that can maximize the statistical independence of the estimated original signal sources. Here, independence implies nonlinear uncorrelatedness; to be specific, we say $\hat{s}_1$ and $\hat{s}_2$ are independent, if any nonlinear transformations $g_1(\hat{s}_1)$ and $g_2(\hat{s}_2)$ are uncorrelated in the sense that their covariance is zero. On the other hand, for two random variables that are nearly uncorrelated, such nonlinear transformations usually do not have zero covariance. By checking the

nonlinear uncorrelatedness of $\hat{s}_1$ and $\hat{s}_2$ computed from (1), we can tell how good the estimated unmixing matrix $\boldsymbol{W}$ is

$$\hat{\boldsymbol{s}} = \begin{bmatrix} \hat{s}_1 \\ \hat{s}_2 \end{bmatrix} = \boldsymbol{W} \cdot \mathbf{x} = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \quad (1)$$

Eventually, we want to find a matrix $\boldsymbol{W}$, so that the components $\hat{s}_1$ and $\hat{s}_2$ are uncorrelated and the transformed components $g_1(s_1)$ and $g_2(s_2)$ are uncorrelated, where g_1 and g_2 are some suitable nonlinear functions. The optimal matrix $\boldsymbol{W}$ is computed iteratively; after each iteration, $\boldsymbol{W}$ is updated by $\Delta \boldsymbol{W}$ using the following two learning rules, where $\mathbf{x}(t) = \boldsymbol{H} \cdot \boldsymbol{s}(t)$.

1) *The Bell's rule*:

$$\Delta \boldsymbol{W} \propto [\boldsymbol{W}^{-1}]^T - 2 \cdot f(\mathbf{x}(t)) \cdot \boldsymbol{s}(t)^T$$

2) *The Amari's rule*:

$$\Delta \boldsymbol{W} \propto [\boldsymbol{I} - f(\mathbf{x}(t)) \cdot \boldsymbol{s}(t)^T] \cdot \boldsymbol{W}.$$

The matrix $\boldsymbol{W}$ is the estimate of the inverse of the mixing matrix $\boldsymbol{H}$ and the function f is a nonlinear sigmoid function, e.g., we can choose the hyperbolic tangent function $\tanh(\cdot)$ as $f(\cdot)$ in practice. The obtained unmixing matrix $\boldsymbol{W}$ is then used to recover the original signal sources, but they might be arbitrarily scaled. In addition, the rows of the unmixing matrix $\boldsymbol{W}$ might have a different ordering than the actual inverse of the mixing matrix $\boldsymbol{H}$, so we have $\boldsymbol{W} \cdot \boldsymbol{H} = \boldsymbol{P}$, where $\boldsymbol{P}$ is a scaling and permutation *finite impulse response* (FIR) matrix. Therefore, the output of the aforementioned separation equation will be arbitrarily scaled, permuted, and delayed original sources. Those scaling and permutation problems can be easily solved by restricting the unmixing matrix update function, and we refer interested audience to [15] for details.

III. PriWhisper System Design

A. The Software Aerial Acoustic Communication Module Design

The architecture of our software aerial acoustic communication module is depicted in Fig. 3. Its main components are the modulator and the demodulator. The narrow-sense Bose, Chaudhuri, and Hocquenghem (BCH) error correcting code [20] is adopted as our channel coding algorithm. The raw data is first channel-encoded and then modulated to an acoustic signal by the modulator. This signal is transmitted by the sender's speaker and collected by the receiver's microphone through the air medium. The received acoustic signal is demodulated by the demodulator and then channel-decoded. In addition, the transmitted data string is padded with CRC-8 to detect transmission errors.

Specifically, we employ frequency-shift keying (FSK) modulation scheme in our current prototype for its smartphone-friendly lightweight signal processing. We use M-ary FSK (MFSK), and each frequency is corresponding to one multi-bit symbol. Let f_c be the carrier frequency and Δf be the shifted frequency for each consecutive multi-bit symbol. Let T be the symbol duration time (unit interval), and we can represent the modulated signal waveform as

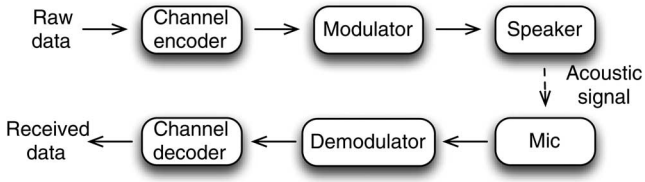

Fig. 3. Acoustic communication module architecture.

$$s(t) = \Re\left(s_m(t)e^{i2\pi f_c t}\right), \quad m \in [0, M-1], \quad t \in [0, T]$$
$$= \sqrt{\frac{2\mathcal{E}}{T}} \cos(2\pi f_c t + 2\pi m \Delta f t)$$

where $\Re(\cdot)$ returns the real component of a complex number, i is the imaginary unit, and

$$s_m(t) = \sqrt{\frac{2\mathcal{E}}{T}} e^{i2\pi m \Delta f t}, \quad m \in [0, M-1], \quad t \in [0, T].$$

Here, we use the coefficient $\sqrt{2\mathcal{E}/T}$ to guarantee that each signal has an energy equal to $\mathcal{E}$.

Once the acoustic signal is received, the receiver tries to detect the symbol transmitted over each unit interval from the received signal $r(t)$ after synchronization. Namely, it needs to determine which frequency component is present in each unit interval. We employ the quadrature receiver using a robust noncoherent detector, e.g., [21]. The quadrature receiver sums the square of the integral of the quadrature components of each frequency $\{f_c + m\Delta f\}_{m=0}^{M-1}$ of the received signal as

$$R_m = \left| \int_0^T r(t) e^{i2\pi(f_c + m\Delta f)t} dt \right|, \quad m \in [0, M-1].$$

We use a calibration sequence to normalize the signal power for each frequency, so that the threshold value can be chosen independently of the frequency to decide the modulated symbols in each unit interval.

B. Determining Optimal Carrier Frequency

The speakers and microphones of all smartphone platforms are specially tailored according to human perception capability. Therefore, as limited by the smartphone hardware, our carrier frequency has to lie in the audible spectrum between 20 and 20,000 Hz. On the other hand, the working environment of PriWhisper might be noisy, especially in outdoor scenarios. Hence, the carrier frequency should be carefully selected to avoid environmental noise. For instance, human voice frequency band ranges from approximately 300–3400 Hz. In order to avoid the ambient noise spectrum, we analyzed a number of environmental noise samples collected from various indoor and outdoor places, such as restaurants. Fig. 4 shows the frequency distribution of two ambient noise samples collected near a road and in a pub. As can be seen, majority of the ambient

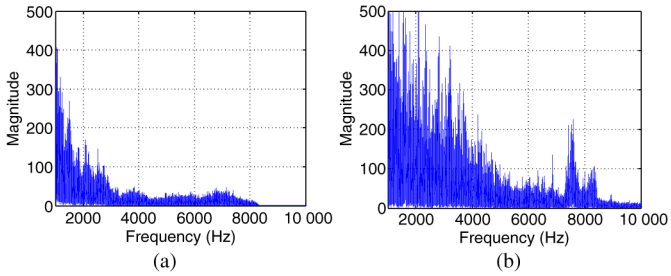

Fig. 4. Ambient noises in frequency domain (recorded by Samsung Galaxy S3). (a) Traffic noise near a road. (b) Musical noise in a pub.

noise lies below 8 kHz, and thus it is desirable to set our carrier frequency above 8 kHz. Meanwhile, we notice that the microphone and speaker hardware of a commercial smartphone typically has different sensitivities for different frequencies. Fig. 5 depicts the frequency response curve tested on Samsung Nexus S smartphone platform. The red line in Fig. 5 stands for the strength of source signal in various frequencies, and the strength of microphone's received signal is plotted in blue. It is easy to see that the channel gain starts to drop dramatically when the signal frequency goes beyond 17 kHz. After considering all the above constraints, we choose our carrier frequency $f_c = 9$ kHz and $\Delta f = 1$ kHz and $M = 2, 4, 8$ for our system prototype. Hence, the receiver is able to filter out all the noise signal components below 8 kHz from the received signal for higher demodulation accuracy.

C. Adaptive Signal Strength Selection

PriWhisper is designed for smartphone environment, and thus the receiver (smartphone) is not expected to be able to transmit arbitrarily strong jamming signal. Without specialized hardware support, the jamming signal strength is always limited by the decibel level of the receiver's speaker hardware in our system. To guarantee the confidentiality of the transmitted data, the system has to adjust the data signal strength of the sender adaptively. Ideally, the optimal decibel level of the data signal should be merely strong enough for the legitimated receiver to demodulate it without error. Once the system bit error rate (BER) performance for different SNRs is known (c.f., Section IV), the sender can adaptively select the optimal signal strength according to its current environmental noise level.

To obtain current ambient noise level, the sender records 0.1-s background noise sample after it generates MFSK modulated data signal. By processing the background noise sample, it can estimate the current noise level around the carrier frequency bandwidth; subsequently, the sender is able to determine the optimal data signal strength and scale the modulated acoustic signal accordingly right before transmission. In addition, we set an upper threshold for the data signal strength to ensure that the jamming signal is at least 10 dB stronger than the data signal for security guarantees. For example, assume the maximum decibel level of the sender's speaker is 60 dB, then the upper threshold of the data signal is defined as 50 dB. Note that the jamming signal is always transmitted at the maximum decibel level of the receiver's

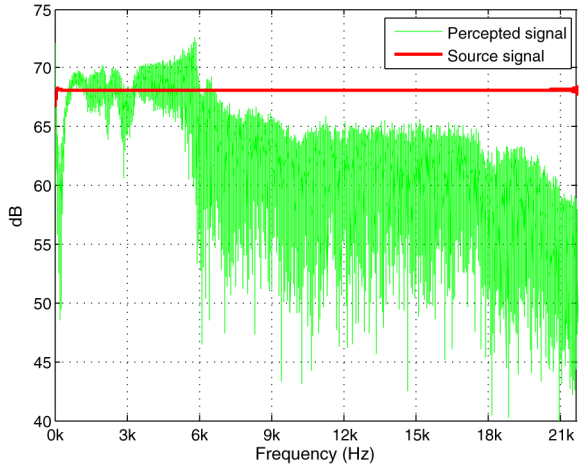

Fig. 5. Frequency response (Samsung Nexus S).

speaker, and we assume the receiver's speaker and the sender's speaker have approximately the same power limitation in practice. During an acoustic communication, our system aborts if the environment is so noisy that the estimated optimal data signal strength exceeds this threshold. When the aforementioned scenarios occur, the user is given a notice indicating that current environment is too hostile for secure communication and encouraged to try again later.

D. Jamming Signal Generation

The receiver needs to generate and transmit the jamming signal to protect the sender's data signal. The length of each communication session (period) is specified by a parameter ℓ_s, and ℓ_s is predefined to be 0.5, 1, or 2 s in our system prototype. The jamming signal should be prepared in priori to each communication session. Since the power of a smartphone speaker is limited, we have to distribute the noise energy in an effective way. The receiver first generates a random white Gaussian noise signal for ℓ_s seconds in the time domain. It then takes Fast Fourier Transform (FFT) to map the signal to the frequency domain and minimizes all the power amounts other than those frequency ranges where the carrier frequencies may lie. The receiver then takes the Inverse FFT (IFFT) of the shaped Gaussian signals as the prepared time-domain jamming signal. For instance, assume the data signal is modulated by FSK with $f_c = 9$ kHz, $\Delta f = 1$ kHz, and $M = 2$. The jamming signal is shaped to cover the frequency range 8.5–10.5 kHz. Fig. 6 depicts the periodogram power spectral density comparison between the generated jamming signal and the data signal. As we can see, these two peaks (at 9 and 10 kHz) of the data signal are well covered by the receiver's jamming signal.

On the other hand, we notice that the jamming signal generation process is quite computationally expensive for the smartphone environment. For example, generating a 2-second jamming signal with sample rate 44.1 kHz takes more than 3 seconds on a Samsung Galaxy S3 smartphone. To keep the acoustic communication smooth, we let the smartphone prepare n_j sections of jamming signals with length ℓ_s offline, where $n_j = 20$ in our prototype. Those jamming signals are stored as monotone

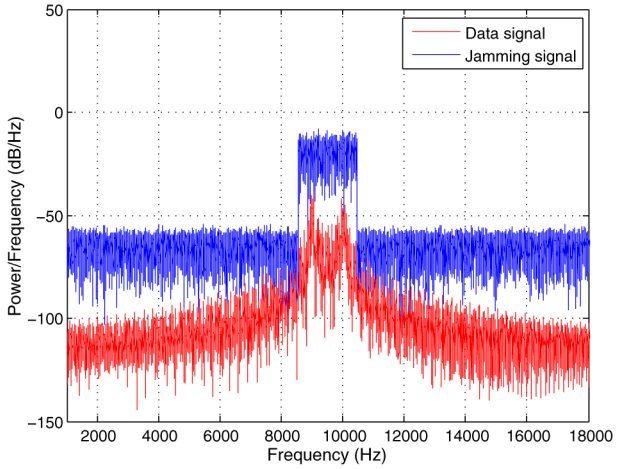

Fig. 6. Comparison of power spectral density.

Pulse-Code Modulation (PCM) 16-bit waveform audio files (WAV files) with sample rate 44.1 kHz, which requires roughly 10 MB storage for $n_j = 20$ and $\ell_s = 2$. Upon request, the receiver loads one jamming signal for the short-coming communication sessions, and it then refills the "jamming signal pool" after the communication.

E. Removing the Jamming Signal

In the smartphone environment, it is impossible to adopt the jamming signal cancellation technique that is used in many existing friendly jamming-based radio communication systems. For example, in [22], the jamming signal is cancelled by an antidote signal transmitted by a special transmit chain connected with the receive chain through a so-called self-looping channel, where the antidote signal is carefully chosen to cancel the jamming signal at the receive antenna's front end. Such jamming signal cancellation techniques require specialized hardware, which is not suitable for off-the-shelf smartphone platforms. Therefore, we would like to remove the jamming signal from the received mixture signal without transmitting an antidote signal.

To achieve the task, the receiver needs to estimate the jamming signal component in the received mixture signal. Given its own generated jamming signal, the receiver utilises a frequency selective fading estimation to obtain the estimated jamming signal received by its microphone. $p(f_i)$ denotes the frequency-selective fading factor for the acoustic signal at frequency f_i transmitted through the receiver's speaker–microphone channel. We note that $p(f_i)$ largely depends on the receiver's hardware, i.e., the sensitivity speaker and microphone of the smartphone for frequency f_i, and its value is obtained empirically from training data. The algorithm is depicted in Fig. 7. We apply an independent frequency-selective fading function to each frequency track obtained from a Short-Time Fourier Transformation (STFT) of the original jamming signal. After independent estimation in the frequency domain, the adjusted signals are combined to the estimate of the received jamming signal by the Inverse STFT.

We also add sinusoid preamble to the jamming signal to facilitate the synchronization process. The data signal can then be easily recovered from the estimated jamming signal and the

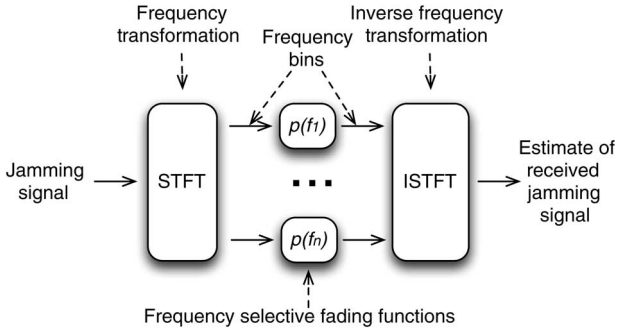

Fig. 7. Estimation of frequency selective fading.

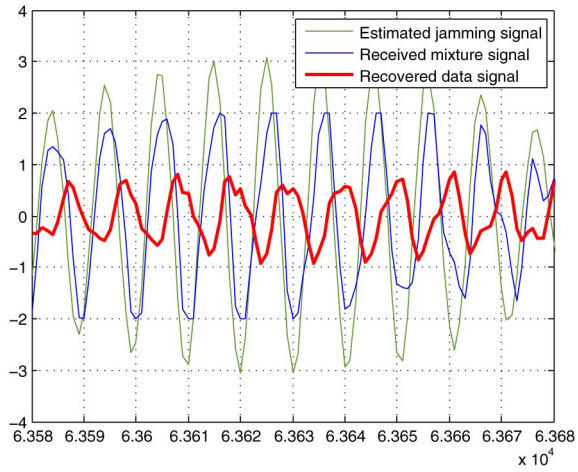

Fig. 8. Recovering the data signal.

received mixture signal. As illustrated in Fig. 8, the estimated jamming signal (denoted as $s_j(t)$) and received mixture signal (denoted as $r(t)$) are plotted in dark green and blue, respectively. As can be seen, the red recovered data signal $(r - s_j)(t)$ preserves good quality using our jamming signal removing technique.

IV. System Integration and Performance

We implement a PriWhisper system prototype on Android 4.1 OS. Notice that the security level of PriWhisper largely depends on the distance between the sender's and receiver's speakers. (c.f., Section V, for discussion.) Hence, we propose an initialization mechanism that can automatically kick-off the communication once the distance requirement is fulfilled. To achieve this task, we utilize smartphone proximity sensors to obtain the distance information between the sender and the receiver. The proximity sensor API of current Android OS can return two values: 0 ("Near") and 5 ("Far"). The threshold distances are different for various smartphone platforms, ranging from 1 to 2 in. When two smartphone users want to establish a secure communication, they simply tap their smartphones together face-to-face. During this process, the receiver is constantly checking its proximity sensor feedback information, and it starts to record and transmit (play) the prepared jamming signal once the feedback becomes "Near".

Algorithm 1 Send$(m, f_c, \Delta f, M, T)$

$x \leftarrow$ Ch_enc(m);
$y \leftarrow$ Modulate$(x, f_c, \Delta f, M, T)$;
$s \leftarrow$ AduioRecord$(0.01s)$;
$a_n \leftarrow$ Detect_background_noise_level(s);
$z \leftarrow$ Adjust(y, a_n);
While ProximitySensor $\neq$ Near **do**
  Sleep$(0.01s)$;
end While
While true do
  $n \leftarrow$ AduioRecord$(0.01s)$;
 if Noisy$(n) =$ **true then**
  Break;
 end if
  end While
AduioPlayback(z);

Algorithm 2 $\hat{m} \leftarrow$ Receive$(f_c, \Delta f, M, T)$

$n \leftarrow$ Prepare_jamming_signal(ℓ_s);
While ProximitySensor $\neq$ Near **do**
  Sleep$(0.01s)$;
end While
$r \leftarrow$ AduioRecord$(\ell_s + \varepsilon)$;
AduioPlayback(n);
$y \leftarrow$ Remove_jamming_signal(r, n);
$x \leftarrow$ Demodulate$(y, f_c, \Delta f, M, T)$;
$\hat{m} \leftarrow$ Ch_dec(x);
return $\hat{m}$.

On the other hand, we should also ensure that the sender's data signal is transmitted strictly after the receiver's jamming signal is on. Hence, the sender cannot simply use proximity sensor to initiate the data signal transmission, because the sensitivity and threshold value of different smartphone proximity sensors are not the same. Besides, it is also inefficient if we delay the data signal transmission by a (sufficiently long) constant time, e.g., 0.5 second. Alternatively, the sender can detect the presence of jamming signals itself. When its proximity sensor indicates "Near," the sender records a 0.01-second background sound sample and calculates root mean square of the amplitudes of the sample. The sender repeats above procedure until the calculated value exceeds a certain threshold t_s, where $t_s = 2000$ for 16-bit

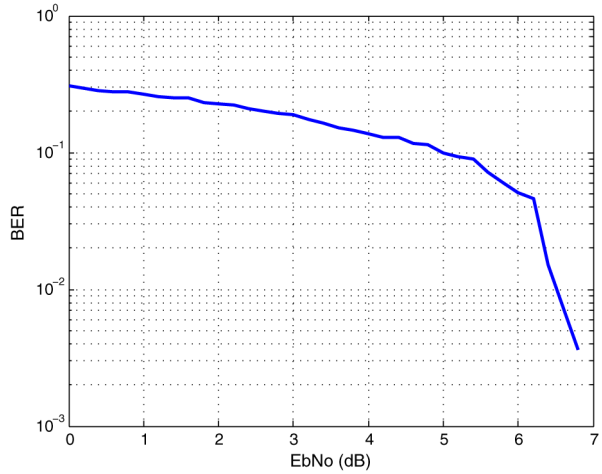

Fig. 9. BER versus SNR for PriWhisper.

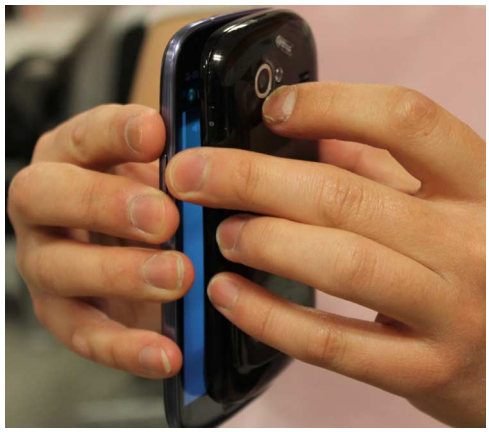

Fig. 10. PriWhisper in action (Samsung Galaxy S3–Google Nexus S).

samples in our system prototype. Once the jamming signal is detected, the sender starts to transmit (play) its modulated data signal. Note that the modulated data signal length must be slightly shorter than the communication session length ℓ_s to ensure that the jamming signal is able to cover the data signal during the entire communication. The simplified pseudo-codes for the sender and the receiver algorithms are depicted in Algorithms 1 and 2, respectively.

In our implementation, we use $n = 255$ and $k = 131$ as the parameters of the narrow-sense BCH error correcting code, which gives us coding rate $R_c \approx 0.514$. The channel-encoded data string is then 3-distance randomly interleaved before modulation. Fig. 9 shows the BER performance of PriWhisper for different SNRs in the log-scale where $T = 2$ ms, $f_c = 9000$ Hz, $\Delta f = 1000$ Hz, and $M = 8$. As can be seen, for SNRs larger than 6.7 dB, the BER reaches zero. We tested our system prototype on various Android smartphone platforms, and it works smoothly across platforms as all the smartphone proximity sensors are located at similar upper front positions. Fig. 10 illustrates an acoustic communication scenario between Samsung Galaxy S3 and Google Nexus S.

During our prototype evaluation, we optimize our prototype to overcome a few encountered subtle problems. For instance,

TABLE I
PERFORMANCE EVALUATION OF PRIWHISPER

M	$1/T$ (Hz)	R_c	Data rate (bps)	PER (indoor) (%)	PER (outdoor) (%)
2	500	0.514	257	0	0
4	500	0.514	514	0	0
8	500	0.514	1027	0.5	3

the above jamming signal detection approach is not suitable to noisy environments, as the data signal may be triggered by ambient noise. To fix it, the sender should transform its recorded sample to the frequency domain by FFT and only check the signal strength of those frequencies around f_c (its carrier frequency). We also noticed that the frequency and shape of the preamble of the receiver's jamming signal could be distorted if the jamming signal starts while two smartphones are still in motion due to the Doppler effect. This tiny distortion may cause synchronization problems and thus leads to transmission errors. The users are supposed to tap their smartphones together, but the proximity sensors usually indicate "Near" before two smartphones touch. To fix it, we utilize smartphone accelerometer sensors to detect its motion. The jamming signal is held until the receiver's accelerometer sensor indicates the smartphone is static after its proximity sensor indicates "Near".

PriWhisper prototype is extensively tested in many noisy hostile indoor/outdoor environments such as restaurants and parks. We find that most types of ambient noises have limited effect on the performance of PriWhisper, as their frequencies are way below PriWhisper's carrier frequencies and thus can be filtered. Table I shows the performance evaluation results of our PriWhisper prototype on Samsung Galaxy S3 smartphone platforms for both indoor and outdoor environments. As can be seen, there is a small package error rate (PER) (1.5%) for the outdoor environment when $M = 8$. The reason is that there is a large "vulnerable" (carrier frequencies) spectrum where $M = 8$ and the outdoor ambient noises are changing all the time. Those package errors are due to sudden noise boosts during the transmission.

We also study the battery consumption of our PriWhisper prototype on many Android platforms. Fig. 11 depicts the remaining battery percentage after 1-h continuous PriWhisper acoustic communication between two Google Nexus 4 smartphones. As can be seen, the sender has approximately 87% battery left while the receiver has about 85% battery left. The reason why the receiver costs more energy than the sender is because the generating high-quality jamming signals are more computationally intensive than data modulation.

To study the usability of PriWhisper, we test the prototype on 50 participants (students/staff/faculties) on campus. Among them, majority are graduate and post-graduate students. The task is to send a picture from one smartphone to another (by a simple touch). Not surprisingly, all the subjects can accomplish the task efficiently regardless their previous NFC experiences. In addition, there is no strong correlation between the time needed to accomplish the task and the participants' gender or age, etc.

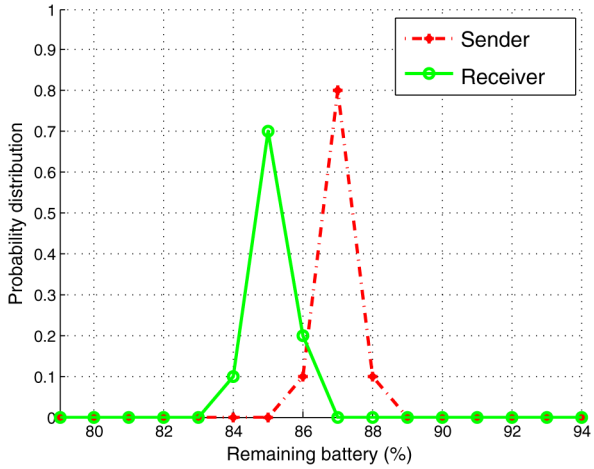

Fig. 11. Battery drain experiment. (The remaining battery after 1-h continuous communication between Google Nexus 4 smartphones.)

V. SECURITY ANALYSIS

A. Security Against Signal-Sensor Passive Adversaries

We first show that PriWhisper protects the confidentiality of the transmitted data against signal-sensor eavesdroppers. We adopt the notion of secrecy capacity as defined in [23], using the difference in the mutual information between the sender and the legitimate receiver versus the eavesdropper to quantify our system confidentiality. Let s_1 be the data signal that has zero mean and variance σ_d^2 and s_2 be the jamming signal that has zero mean and variance σ_j^2. Assume that the channel noise e_i follows Gaussian distribution with zero mean and variance σ_e^2. The acoustic mixture signal obtained by the adversary's sensor R_i can be expressed as

$$x_i = h_{i1}s_1 + h_{i2}s_2 + e_i. \tag{2}$$

Suppose the legitimate receiver Y is able to obtain signal $y = h_y s_1 + e_y$ after removing its jamming signal, where e_y has the same distribution as e_i, then the secrecy capacity can be expressed as

$$\begin{aligned}\tilde{\mathcal{C}}_{\text{sec}} &= I(Y;S) - I(R_i;S) \\ &= \log\left(1 + \frac{|h_y|^2\sigma_d^2}{\sigma_e^2}\right) - \log\left(1 + \frac{|h_{i1}|^2\sigma_d^2}{|h_{i2}|^2\sigma_j^2 + \sigma_e^2}\right).\end{aligned}$$

Hence, we can bind $I(R_i;S)<\kappa$, where κ is the security parameter. To achieve better security guarantees σ_j^2 should be significantly larger than σ_d^2. However, there is always a trade-off between the usability and security. It is easy to see that the lesser the σ_e, the higher the $\tilde{\mathcal{C}}_{\text{sec}}$ that our system can reach; therefore, it is favorable to operate PriWhisper in quiet environment. On the other hand, it is not clear whether the system is still secure if the eavesdropper is able to control multiple sensors located at arbitrary positions, and thus we examine the advantage of a multiple-sensor eavesdropper in Section V-B.

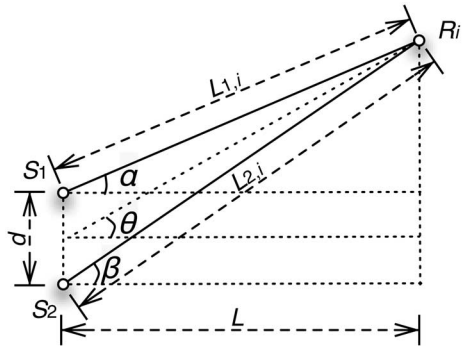

Fig. 12. Two signal sources and one sensor in the LOS channel model.

B. Security Against Multiple-Sensor Passive Adversaries

We now show that extra sensors cannot increase the adversaries' advantages when they are outside the "safe perimeter." Intuitively, we are going to show the mixture signals obtained by the adversaries' sensors are very close to linear combinations of each other. Fig. 12 illustrates a communication scenario in the LOS channel model, where the distance between two signal sources is denoted by d and R_i is an arbitrary sensor, whose location is uniquely determined by the parameters α, β, θ, and L. Let $L_{1,i}$ and $L_{2,i}$ be the distances between the signal sources to the sensor, respectively. We can express their distance difference as $\Delta L = |L_{1,i} - L_{2,i}|$.

By plugging in the frequency-selective fading function $p(f_c, \ell)$ to the above distance, we deduce the channel difference $\Delta h_{f_c} = |h_{i1} - h_{i2}|$ as

$$\Delta h_{f_c} = p\left(f_c, L/\cos(\theta) + \frac{\Delta L}{2}\right) - p\left(f_c, L/\cos(\theta) - \frac{\Delta L}{2}\right).$$

For simplicity, assuming the fading function is homogeneous and uniform, we have

$$\Delta h \approx \frac{dL \cdot p}{\sqrt{8L^2 - 4\tan(\theta)Ld + d^2}}.$$

Taking any two received mixture signals x_i and x_j in form of (2), we have

$$x_i = h_{i2}(s_1 + s_2) \pm \Delta h_i \cdot s_1 + e_i$$
$$x_j = h_{j2}(s_1 + s_2) \pm \Delta h_j \cdot s_1 + e_j.$$

Recall that PriWhisper adaptively adjusts the data signal strength according to the noise level, say $\mathrm{EbNo} = 8$ dB in practice (c.f., Section III-C). Therefore, when Δh_i and Δh_j are small, it is difficult to distinguish x_i and x_j from $h_{i2}(s_1 + s_2) + e_i$ and $h_{j2}(s_1 + s_2) + e_j$ respectively. So, x_i and x_j are nearly linear combination of each other.

In order to determine the relationship between d, L, and Δh in reality, we conduct an eavesdropping experiment on Samsung Galaxy S3–Google Nexus S. Two sensors (microphones) R_1 and R_2 are placed at within 30 cm distance to the communicating smartphones. To maximize the signal component difference of the received signals, they are aligned in the line R_1–S_1–S_2–R_2. To capture the notion of how close the received signals are linear combination of each other, we define the channel similarity as

$$\varepsilon = \left|\frac{h_{11}}{h_{21}} - \frac{h_{12}}{h_{22}}\right|.$$

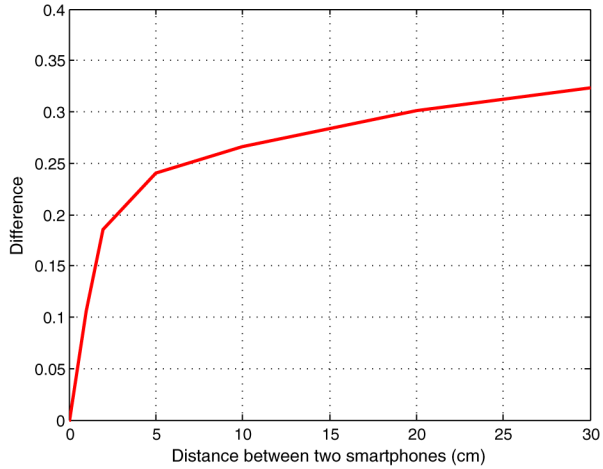

Fig. 13. Channel similarity versus distance.

Fig. 13 plots the smoothed channel similarity curve $\varepsilon(d)$ for $d \in (0, 30]$ based on our experiments. As can be seen, $\varepsilon(d)$ drops exponentially, when d tends to 0. For instance, the recommended PriWhisper working distance is less than 0.5 cm, which gives us $\varepsilon < 0.03$. We can deduce a safe distance by combining this number together with an approximate sound attenuation factor. Assume that a sound wave is propagated from a to b in distance ℓ, and let A_a (or A_b) be its amplitude at location a (or b). By the inverse square law, it is well-known that $A_b = A_a \cdot e^{-\alpha\ell}$, where α is the attenuation coefficient. However, the actual decaying effect depends on many factors, including carrier frequency, humidity, temperature, etc. In practice, we assume that the acoustic environment is a semi-reverberant field in which the sound with carrier frequency of 9–17 kHz decays by at least 10 dB when it passes the first 2 m at a relative humidity of less than 50% and temperature of above 15 °C. Given the $\mathrm{SNR} = 8$ dB and $\varepsilon < 0.03$, we can see the variances of the channel noise e_i and e_j are roughly 100 folders larger than the channel difference. Hence, using extra sensors cannot provide additional advantage to the eavesdropper if all the sensors are 2 m away from the signal sources in practice. We also conduct the outdoor acoustic signal decay experiment to validate the above security claim. As shown in Fig. 14, the SNR of the data signal already drops to nearly 0 at locations with 1.5 m distance in a standard PriWhisper communication scenario.

Remark: Most smartphone platforms are equipped with two speakers: in-call speaker and main speaker (also known as rear speaker in Android platforms). The location of the main speakers may be different for different smartphone platforms; nevertheless, their in-call speakers are always located at the same place near their proximity sensors. If one or both smartphone(s) use(s) the main (rear) speakers for acoustic communication, the

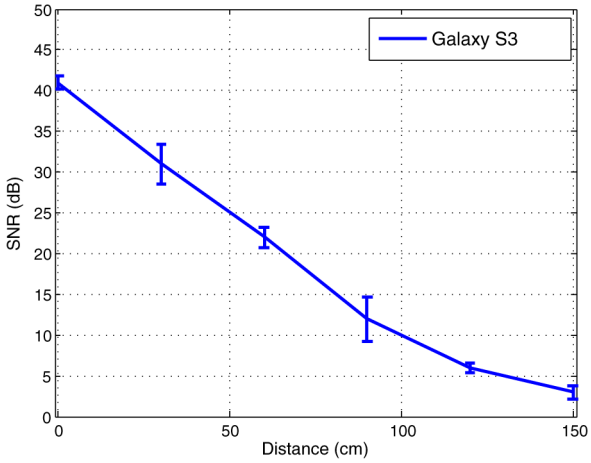

Fig. 14. Outdoor acoustic signal decay experiment on Samsung Galaxy S3 (distance from 0 to 150 cm).

actual distance between the signal sources is larger than the distance between these two smartphones. For example, when a Galaxy S3 and a Nexus S are aligned face-to-face at a distance of 0.5 cm, the distance between their two rear speakers are about 1.5 cm. Fortunately, the decibel levels of the in-call speakers are sufficient for acoustic communication on almost all recent smartphone platforms. Since the distance is a very important security factor, both data signals and jamming signals are transmitted by the smartphones' in-call speakers in our PriWhisper prototype. When old smartphone models are used, the users can always switch to main speakers, pursuing better usability. However, it may slightly decrease the system security strength.

C. Inseparability of the Mixture Signal

The system security may break down if the adversaries can separate the data signal and jamming signal using multiple sensors. Hence, PriWhisper's data confidentiality also largely depends on the hardness of separating the data signal from the eavesdropped mixture signals. In this section, we examine the feasibility of signal segmentation using ICA. The adversary's received mixture signals $\boldsymbol{x}$ can be expressed as $\boldsymbol{x} = \boldsymbol{H} \cdot \boldsymbol{s} + \boldsymbol{e}$. Ignore the channel noise $\boldsymbol{e}$ for simplicity, the task of an ICA approach is to find an unmixing matrix $\boldsymbol{W}$, and ideally we should have $\boldsymbol{W} \cdot \boldsymbol{H} = \boldsymbol{I}$. If succeeded, the eavesdropper outputs

$$\begin{bmatrix} \hat{s}_1 \\ \hat{s}_2 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}.$$

Obviously, the separability of the mixture signals is directly connected to the invertability of the mixing matrix. The accuracy of ICA algorithms decreases dramatically when the mixing matrix is a nearly rank deficit matrix. Here, we can use the concept of ε-rank to quantify the (in)separability of the data signal and jamming signal:

$$\mathsf{Rank}(\boldsymbol{A}, \varepsilon) = \min_{\|\boldsymbol{A}-\boldsymbol{B}\| \le \varepsilon} \mathsf{Rank}(B).$$

Here, the matrix norm is defined as

$$\|\boldsymbol{M}\|_p = \sup_{\|\mathbf{x}\|_p = 1} \|\boldsymbol{M}\mathbf{x}\|_p.$$

When 1-norm or 2-norm is used, it is straightforward to show the ε'-rank of the mixing matrix $\mathsf{Rank}(\boldsymbol{A}, \varepsilon') = 1$ using the linear combination arguments in the previous section for some ε'. The channel noise factor further decreases the success rate of ICA in practice.

We validate the inseparability of the data signal and the jamming signal using state-of-art ICA algorithms. During the simulated attack, the adversary's sensors are located approximately 1 meter away from the communicating smartphones. As depicted in Figs. 15 and 16, the left columns are the data signal (red) and the jamming signal (green); the middle columns contain two received mixture signals x_1 and x_2; the right columns contain the estimated (recovered) signals. As can be seen, the adversary can successfully separate the data signal and the jamming signal when the sender and receiver are 30 cm away; whereas, the estimated signals are nearly random when the distance between the sender and the receiver is 1 cm.

VI. Applications

PriWhisper has many potential security-sensitive smartphone applications such as devise pairing, personal authentication, and user data exchange. In this section, we present AcousAuth—a novel smartphone empowered system that utilizes PriWhisper for personal authentication.[1]

In particular, the AcousAuth utilizes PriWhisper in the smartphone and cloud-based terminal scenario, where the client uses his/her smartphone as a secure contactless credential to authenticate himself/herself. The system consists of three main entities: a *user authentication* running on user's smartphone, a *web-browser-based graphic user interface* (*GUI*) running on a computer, and a *cloud-based online acoustic signal-processing terminal* running on a virtual private machine (VPS). In order to achieve platform independent, we implement online GUI based on HTML5 technique, due to the fact that HTML5 has brought a surge of access to device hardware and its Web Audio API supports local microphone stream access and self-jamming signal broadcast.

At the beginning of an authentication phase, the smartphone first authenticates the user through a password or biometric-based authentication scheme. Once the user is authenticated, the smartphone sends its stored secret to the target server using modulated acoustic signals. Fig. 17 depicts the interaction between the smartphone and the front-end UI during an authentication phase. The smartphone first sends StartSignal to the front-end UI. The front-end UI then starts recording and sends SynchronizationSignal back to the smartphone. After that, the smartphone sends the DataSignal (modulated by PriWhisper sender) to the front-end web-based UI, and meanwhile the front-end UI is playing the jamming noise simultaneously. The front-end UI then forwards the recorded acoustic signal to

[1]AcousAuth is one of the top 10 participants in the finalist of the MobiCom 2013 Mobile App Competition [24].

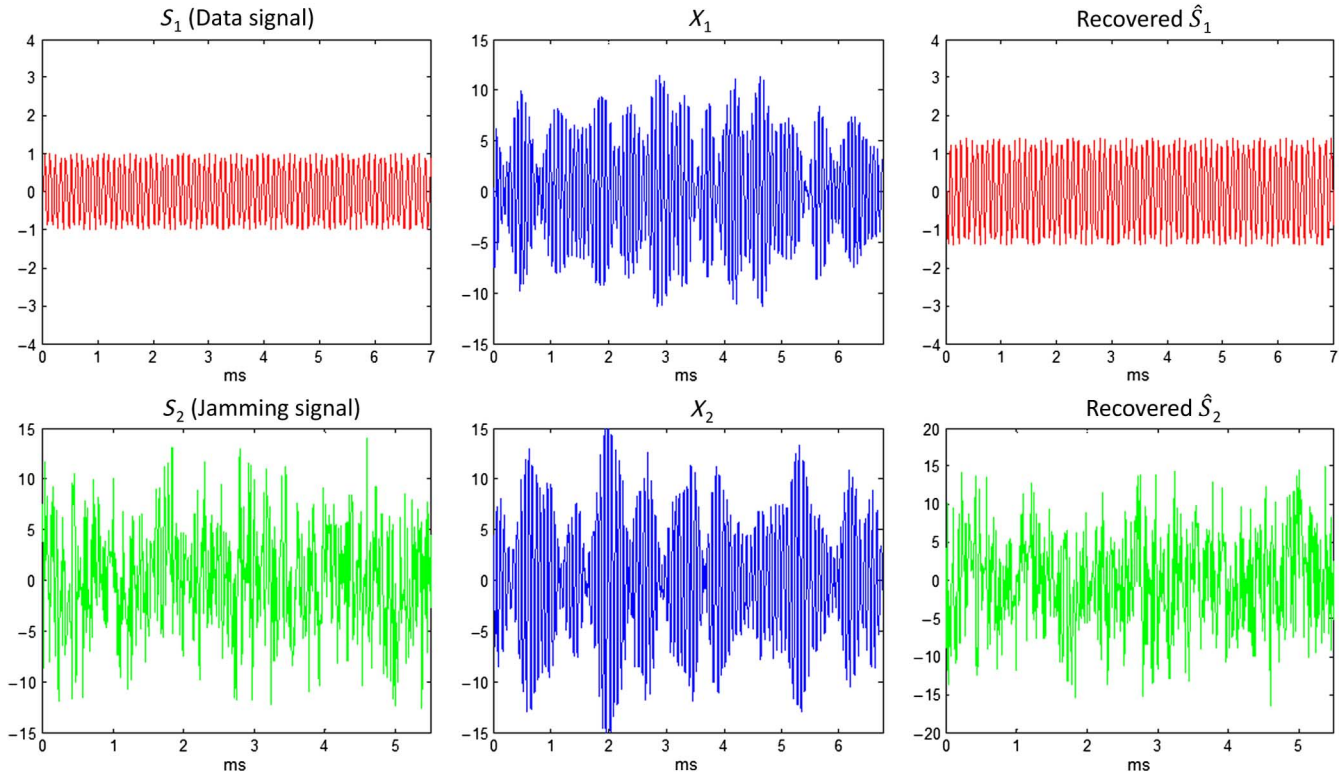

Fig. 15. Successful attack instance (two signal sources are 30 cm away).

the cloud for signal processing and authentication. Finally, the cloud server replies its decision back to the front-end UI.

With very little modification, such system can be extended to an acoustic mobile payment system. Hence, we can turn the smartphones into magnetic stripe cards. Unlike Google Wallet, the bank card information is only stored in the user's own smartphone instead of a third party server. Therefore, the users' private bank card information is safe as far as their smartphone is not compromised.

VII. Related Work

Friendly jamming technique was first proposed by Negi and Goel [25] in 2005. In their work, the jamming signals are generated from the null space of the legitimate receiver's channel vector, and thus the jamming signal does not effect the receiver but other eavesdroppers at different locations. Gollakota *et al.* [22] first extend friendly jamming technique to a single full-duplex receiver in 2011. Their system uses a specialized hardware and thus limits its application in many scenarios. Our work made the novel observation that COTS smartphones can be readily used to implement the so-called "single-channel full-duplex" approach by utilizing the speaker and microphone simultaneously without any additional hardware support. Moreover, the security analysis in [22] is hand-waving without any quantitative evaluation. Our work and the parallel work by Tippenhauer *et al.* [26] are among the first that (independently) provide the quantitative security analysis for the friendly jamming technique. Both ours and [26] showed the limitation and the attack possibilities in the respective scenarios. We recognized and studied blind signal separation-based attacks such as ICA, while MIMO attacks were discussed in [26]. Upon submission, we noticed that a parallel independent work by Nandakumar *et al.* [27] is accepted by SIGCOMM 2013. However, their work only focuses on the system implementation aspect, whereas our work also provides rigorous security analysis. We strongly believe that the insight on friendly jamming security presented in our work is very important for the future research.

Recently, Bursztein *et al.* [28] utilize BSS techniques to attack noise-based noncontinuous audio Captchas. They can show the computer is able to distinguish those audio Captchas at a human-comparable correct rate. In terms of software acoustic modem, Lopes and Aguiar [29] present an aerial acoustic communication system using software modem in 2001. Mostafa [30] released a software modem called minimodem that supports many traditional modem protocols, e.g., Bell 103 on Linux OS. Michel [31] implemented a software modem for Android system supporting amplitude-shift keying (ASK) modulation, and it can modulate data in musical tones. Houmansadr *et al.* [32] realize a software modem supporting QAM modulation, and they use it to build IP over VoIP to achieve communication unobservability against traffic analysis and standard censorship techniques. Acoustic modems are also used in ubiquitous computing [33] and navigation systems [34]. Recently, there is a trend of utilizing acoustic communication technologies in mobile payment systems, e.g., [35] and [36]; hence, we believe PriWhisper could be a great candidate for acoustic smartphone communication with build-in security mechanisms.

PriWhisper *versus NFC*: NFC requires additional hardware and thus it is not widely supported by various smartphone

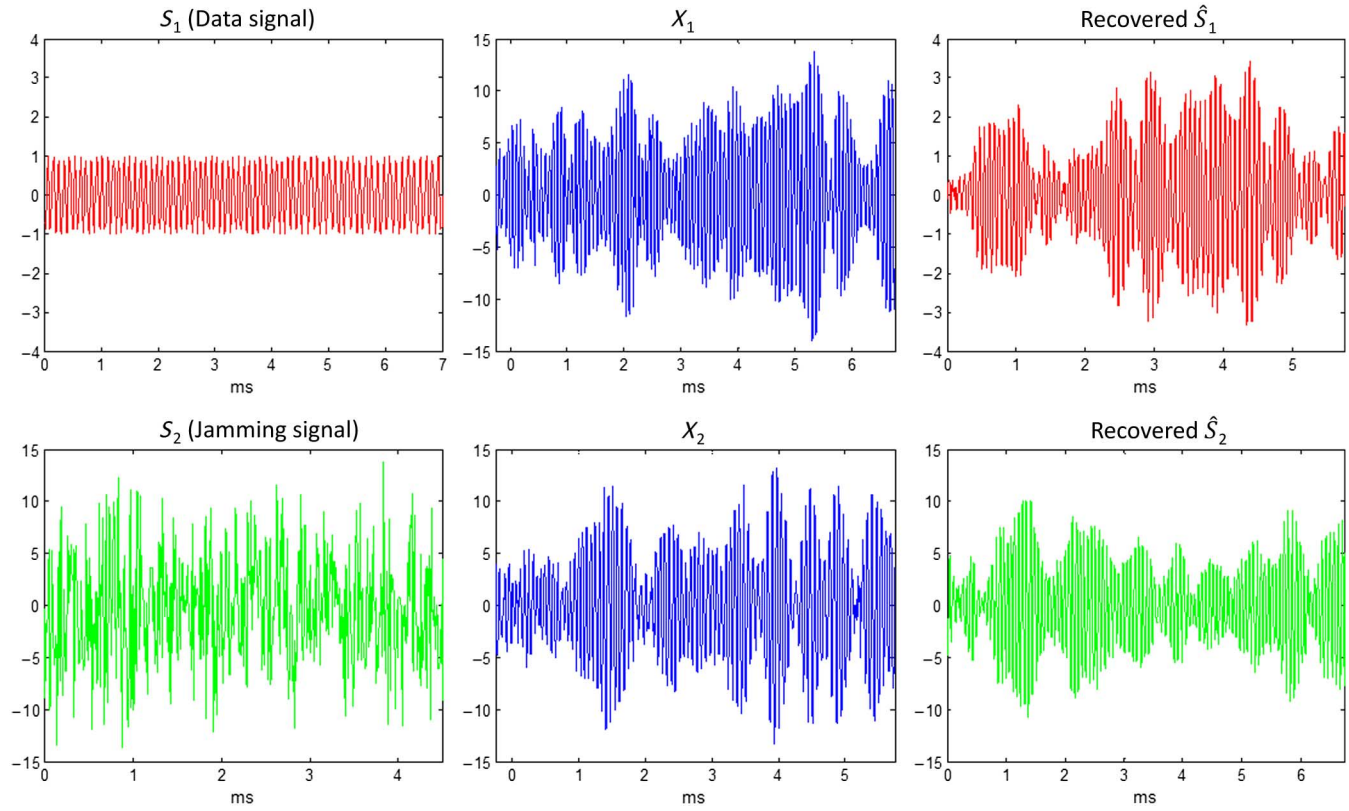

Fig. 16. Failure attack instance (two signal sources are 1 cm away).

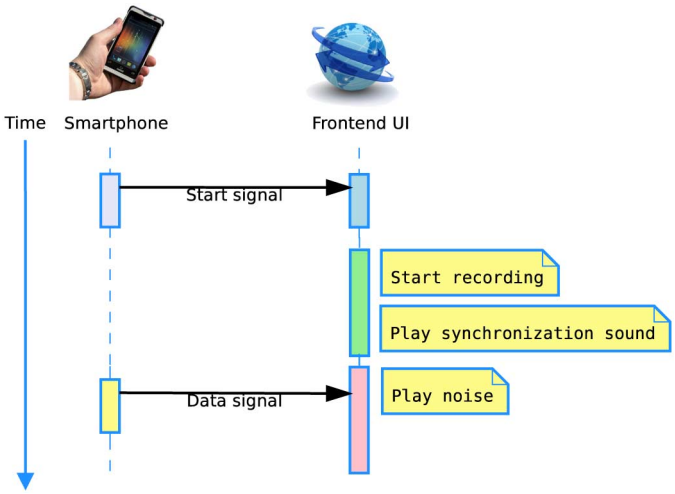

Fig. 17. Sequence diagram.

platforms such as iPhone series; whereas, PriWhisper is compatible with most off-the-shelf smartphone platforms. In addition, as mentioned before, friendly jamming technique cannot be implemented with current NFC APIs, so NFC is not able to offer build-in security features as PriWhisper does. Both PriWhisper and NFC shares great usability, i.e., the communication can be accomplished by a simple touch. Although NFC may provide higher transmission rate, we believe that PriWhisper's system throughput is sufficient for most practical security-sensitive mobile applications.

VIII. Conclusion and Future Work

We designed, implemented, and evaluated PriWhisper, a keyless secure acoustic short-range communication system for smartphones. Its security has been analytically and experimentally studied, especially against BSS attacks. We also presented AcousAuth as an application example. The system throughput of our current prototype is 1 kbps, and we would like to improve the system throughput in our future work. We will also extend PriWhisper to many other major smartphone OSs such as iOS. In addition, we want to examine the feasibility of (military level) vibration-based sound recovery attacks on PriWhisper and study effective countermeasures. As a further improvement, we plan to enhance PriWhisper's security against active adversaries by utilizing smartphone sound localization techniques to automatically detect the distance of the incoming acoustic signal source; subsequently, it is able to reject unintended signals.

Acknowledgment

J. Wang, J. Wang, Y. Tu, and C. Zhang have contributed to the implementation of PriWhisper and AcousAuth systems.

References

[1] Google, Google Wallet [Online]. Available: http://www.google.com/wallet/index.html. Accessed on Jan. 01, 2013.
[2] S. Millward, AliPay's mobile barcode payments in China [Online]. Available: http://www.techinasia.com/alipay-mobile-payments/. Accessed on Jan. 01, 2013.
[3] R. Kim, PayPal's barcode-based payment services in UK [Online]. Available: http://gigaom.com/2012/05/30/paypal-rolls-out-barcode-payments-in-the-uk/.

[4] A. Perrig, J. M. Mccune, and M. K. Reiter, "Seeing-is-believing: Using camera phones for human-verifiable authentication," in *Proc. IEEE Symp. Security Privacy,* 2005, pp. 110–124.
[5] R. Kainda, I. Flechais, and A. W. Roscoe, "Usability and security of out-of-band channels in secure device pairing protocols," in *Proc. SOUPS '09,* ACM, 2009, pp. 11:1–11:12.
[6] J. Guerrieri and D. Novotny, "HF RFID eavesdropping and jamming tests," Electromagnetics Div., Electronics and Electrical Engineering Lab., National Inst. Standards and Technology, Tech. Rep. 818-7-71, 2006.
[7] Norm ECMA-385, "NFC-SEC: NFCIP-1 Security Services and Protocol" [Online]. Available: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-385.pdf, 2010.
[8] Norm ECMA-386, NFC-SEC-01: NFC-SEC Cryptography Standard using ECDH and AES Reference [Online]. Available: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-386.pdf, 2010.
[9] M. Erol-Kantarci, H. T. Mouftah, and S. F. Oktug, "A survey of architectures and localization techniques for underwater acoustic sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 487–502, 2011.
[10] R. Headrick and L. Freitag, "Growth of underwater communication technology in the U.S. Navy," *IEEE Commun. Mag.*, vol. 47, no. 1, pp. 80–82, Jan. 2009.
[11] R. Jurdak, C. V. Lopes, and P. Baldi, "Software acoustic modems for short range mote-based underwater sensor networks," in *Proc. IEEE Oceans Asia,* 2006, pp. 1–7.
[12] S. Goel and R. Negi, "Guaranteeing secrecy using artificial noise," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2180–2189, Jun. 2008.
[13] D. Phan and J. Cardoso, "Blind separation of instantaneous mixtures of non-stationary sources," *IEEE Trans. Signal Process.*, vol. 49, no. 9, pp. 1837–1848, Sep. 2001.
[14] A. Cichocki, J. Karhunen, W. Kasprzak, and R. Vigario, "Neural networks for blind separation with unknown number of sources," *Neurocomputing*, vol. 24, no. 1, p. 5593, 1999.
[15] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain," *Neurocomputing*, vol. 15, pp. 745–770, 1998.
[16] J. Cardoso, "Blind source separations: Statistical principles," *Proc. IEEE*, vol. 86, no. 10, pp. 2009–2025, 1998.
[17] M. Reyes-Gomez, B. Raj, and D. Eliss, "Multi-channel source separation by factorial HMMS," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing* (ICASSP), 2003, pp. 664–667.
[18] S. T. Roweis, "One microphone source separation," in *Proc. Adv. Neural Inf. Process. Syst.* (NIPS13), pp. 793–799, 2001.
[19] L. Benaroya, L. McDonagh, F. Bimbot, and R. Gribonval, "Non-negative sparse representation for Wiener based source separation with a single sensor," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* (ICASSP), pp. 613–616, 2004.
[20] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Inf. Control*, vol. 3, no. 1, pp. 68–79, Mar. 1960.
[21] P. Liu, S. Gazor, I.-M. Kim, and D. I. Kim, "Noncoherent amplify-and-forward cooperative networks: Robust detection and performance analysis," *IEEE Trans. Commun.*, vol. 61, no. 9, pp. 3644–3659, Sep. 2013.
[22] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu, "They can hear your heartbeats: Non-invasive security for implantable medical devices," in *Proc. SIGCOMM,* 2011, pp. 2–13.
[23] I. Csiszar and J. Korner, "Broadcast channels with confidential messages," *IEEE Trans. Inf. Theory*, vol. 24, no. 3, pp. 339–348, May 1978.
[24] MobiCom, MobiCom 2013Mbile app competition [Online]. Available: http://www.sigmobile.org/mobicom/2013/app_finalists.html.
[25] R. Negi and S. Goel, "Secret communication using artificial noise," in *Proc. IEEE Veh. Technol. Conf.,* 2005, pp. 1906–1910.
[26] N. O. Tippenhauer, L. Malisa, A. Ranganathan, and S. Capkun, "On limitations of friendly jamming for confidentiality," in *Proc. S & P* (Oakland), 2013.
[27] R. Nandakumar, K. K. Chintalapudi, V. Padmanabhan, and R. Venkatesan, "Dhwani: Secure peer-to-peer acoustic NFC," in *Proc. ACM SIGCOMM Conf. SIGCOMM, Ser. SIGCOMM '13*, New York, NY, USA: ACM, 2013, pp. 63–74 [Online]. Available: http://doi.acm.org/10.1145/2486001.2486037.
[28] E. Bursztein, R. Beauxis, H. Paskov, D. Perito, C. Fabry, and J. C. Mitchell, "The failure of noise-based non-continuous audio captchas," in *Proc. S & P* (Oakland), 2011.
[29] C. Lopes and P. Aguiar, "Aerial acoustic communications," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.,* 2001, pp. 219–222.
[30] K. Mostafa, Minimodem [Online]. Available: http://www.whence.com/minimodem/. Accessed on Jan. 01, 2013.
[31] J. Michel, Mobile acoustic modems in action [Online]. Available: https://code.google.com/p/mobile-acoustic-modems-in-action/.
[32] A. Houmansadr, T. Riedl, N. Borisov, and A. Singer, "I want my voice to be heard: IP over Voice-over-IP for unobservable censorship circumvention," in *Proc. NDSS*, 2013, pp. 1–17.
[33] C. Lopes and P. Aguiar, "Acoustic modems for ubiquitous computing," *IEEE Pervasive Comput.*, vol. 2, no. 3, pp. 62–71, Jul. 2003.
[34] L. Freitag, M. Grund, I. Singh, J. Partan, P. Koski, K. Ball *et al.*, "The WHOI micro-modem: An acoustic communications and navigation system for multiple platforms," in *Proc. IEEE OCEANS Conf. Exhib.,* 2005, pp. 1086–1092.
[35] CNET, Naratte: Mobile payments using sound waves [Online]. Available: http://news.cnet.com/8301-19882_3-20072295-250/naratte-mobile-payments-using-sound-waves/.
[36] Alipay, Sound wave mobile payment [Online]. Available: http://techcrunch.com/2013/04/14/alipay-launches-sound-wave-mobile-payments-system-in-beijing-subway/. Accessed on Apr. 15, 2013.

Bingsheng Zhang received the B.Eng. degree in computer science from Zhejiang University of Technology, Hangzhou, China, in 2007, the M.Sc. degree in information security from University College London, London, U.K., in 2008, and the Ph.D. degree in computer science from the University of Tartu, Tartu, Estonia, in 2011.

He is a Postdoctoral Researcher with CSE Department of SUNY, Buffalo, NY, USA. Before his current appointment, he was a part-time Research Associate with University College London and a full-time Researcher with Cybernetica AS.

Qin Zhan (S'13) received the B.E. degree in information engineering from Beijing Institute of Technology, Beijing, China, in 2010, and the M.Sc. degree in electronic engineering from Columbia University, New York, NY, USA, in 2012. He is a Ph.D. candidate with CSE Department of SUNY, Buffalo, NY, USA.

He has worked at Microsoft Research Asia in the summer of 2011. His research interests focus on security and privacy of cloud computing.

Si Chen (S'13) received the B.S. degree in electrical engineering from the China Agricultural University, China, in 2010, and the M.S. degree in EE Department of SUNY, Buffalo, NY, USA in 2012. He is a Ph.D. candidate with CSE Department of SUNY, Buffalo.

He is currently working in the Ubiquitous Security and Privacy Research Laboratory under the guidance of Professor Kui Ren.

Muyuan Li (S'13) received the B.A. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2013. He is a Ph.D. candidate at CSE Department of SUNY, Buffalo, NY, USA.

His current research interests include computer/smartphone system and security. He is a student member of IEEE COMSOC and ACM.

Kui Ren (SM'12) obtained the Ph.D. degree in electrical and computer engineering from Worcester Polytechnic Institute, Worcester, MA, USA, in 2007.

He is an Associate Professor with CSE Department of SUNY, Buffalo, NY, USA. Prior to that, he had been an Associate Professor with the Electrical and Computer Engineering Department, Illinois Institute of Technology, Chicago, IL, USA. His research interests include security and privacy in cloud computing, wireless security, smart grid security, and sensor network security. His research is supported by NSF, DoE, AFRL, and Amazon.

Dr. Ren is a co-recipient of the Best Paper Award from IEEE ICNP 2011. He is a recipient of NSF Faculty Early Career Development (CAREER) Award in 2011 and Sigma Xi/IIT Research Excellence Award in 2012. He is a Member of ACM.

Cong Wang (M'12) received the B.E. and M.E. degrees from Wuhan University, Wuhan, China, in 2004 and 2007, and the Ph.D. degree from Illinois Institute of Technology, Chicago, IL, USA, in 2012, all in electrical and computer engineering.

He is an Assistant Professor with the Computer Science Department, City University of Hong Kong, Kowloon, Hong Kong. He has worked at Palo Alto Research Center in the summer of 2011. His research interests are in the areas of cloud computing and security, with current focus on secure data services in cloud computing, and secure computation outsourcing. He is a Member of the ACM.

Di Ma (M'09) received the Ph.D. degree from the University of California, Irvine, CA, USA, in 2009.

She is an Assistant Professor with the Computer and Information Science Department, the University of Michigan-Dearborn, Dearborn, MI, USA, where she leads the Security and Forensics Research Lab (SAFE). She was with IBM Almaden Research Center in 2008 and the Institute for Infocomm Research, Singapore, in 2000–2005. She is broadly interested in the general area of security, privacy, and applied cryptography. Her research spans a wide range of topics, including computation over authenticated/encrypted, fine-grained access control, secure storage systems, wireless network security, smartphone security and privacy, and so on.

Dr. Ma won the Tan Kah Kee Young Inventor Award in 2004.