PriWhisper: Enabling Keyless Secure Acoustic Communication for Smartphones

Bingsheng Zhang
University at Buffalo, USA

Qin Zhan
University at Buffalo, USA

Junfei Wang
University at Buffalo, USA

Kui Ren
University at Buffalo, USA

Cong Wang
City University of Hongkong, Hongkong

Di Ma
University of Michigan-Dearborn, USA

***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. Namely, both parties need to spend extra communication to establish a common key before transmitting their actual messages, 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 purely software-based solution to secure smartphone short-range 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 various passive and active adversaries. In particular, we also 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.**

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 and 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 of smartphones' screen resolution exacerbates the immigration of conventional 1D/2D 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], [5]. Since the two communicating devices must be within $1 \sim 2$ 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. Take the barcode based system as an example. Due to its fundamental design principle, the visual nature of barcode based short-range communication makes them extremely vulnerable to shoulder sniffing. The wide spread 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 m 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) 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.

In recognizing these design challenges, in this work we initiate the research endeavour 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], [10], [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. Secondly, 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. Thirdly, 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 our best knowledge, acoustic channel is the only full-duplex channel we can control freely from smartphone OS APIs. 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 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, which exploits friendly jamming technique from radio communication for data confidentiality. To our best 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 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 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 give two useful PriWhisper applications: smartphone key exchange and acoustic mobile payment system.

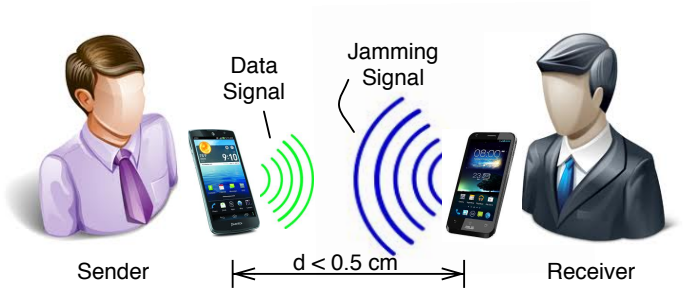


Figure 1. System Architecture.

Relationship to [13] and [14]. First of all, we emphasize that the solution realized in [13] requires specialized hardware and thus limits its application in many scenarios. This 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. We carried out the idea and successfully developed a prototype system that is demonstrated with sufficient throughput to support most security sensitive smartphone applications. Upon the submission of this paper, we believe that this is the first work that successfully realized such functionality on the commercial smartphones. Secondly, the security analysis in [13] is hand-waving without any quantitative evaluation. This work and the parallel work [14] are among the first that (independently) provides the quantitative security analysis for the friendly-jamming technique. Both ours and [14] 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 [14]. We strongly believe that the insight on friendly-jamming security presented in this work is very important for the future research.

The rest of this paper is organised as follows. Sec. II introduces our system architecture, threat model and technical background. In Sec. III, we present the PriWhisper system design. In Sec. IV, we implement the proposed system and test its performance. We thoroughly analyse the security of our proposed system in Sec. V. In Sec. VI, we provide two applications, including smartphone pairing and acoustic mobile payment system. Finally, Sec. VII summarises related work, and a conclusion is given in Sec. 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, which offers great compatibilities to various smartphone platforms without additional hardware requirement. Any

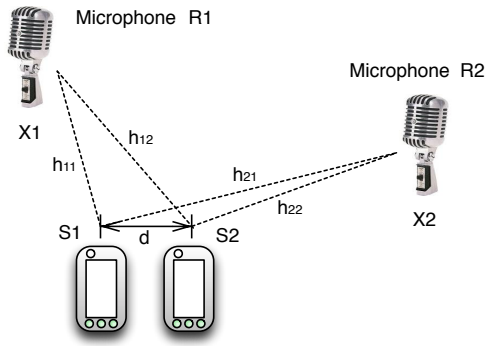


Figure 2. Illustrative attack scenario where the adversary uses two microphones.

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 a 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 $1 \sim 2$ seconds for most smartphone applications.

B. Security goal and threat model

PriWhisper is expected to provide secure communication in presence of both passive eavesdroppers and active adversaries. 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 frequency, and ℓ is the distance parameter. A similar channel model can be found in [15], on which the channel model assumes that all transmitted signals experience the same channel condition though one path. PriWhisper is designed to protect confidentiality of the transmitted data against single- and 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

$$\boldsymbol{x}(t) = \boldsymbol{H} \cdot \boldsymbol{s}(t) + \boldsymbol{e}(t) \ ,$$

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 *Blind Signal Segmentation* (BSS) techniques such as *Independent Component Analysis* (ICA). See App. II-D for details. Upon success, the adversary can recover the transmitted data from the estimated data signal.

In addition, we also briefly examine the security of PriWhisper against several active attacks. For instance, the system robustness against DoS attacks and the integrity of the transmitted data against data injection attacks will be discussed as a part of our security analysis.

C. Blind signal segmentation

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 [16]. 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 [17]. ICA is one of the most famous algorithms to solve well-determined and over-determined BSS. A classic ICA approach estimates the de-mixing filters by assuming that the source signals are independent and non-Gaussian [18] or Gaussian with a non-stationary variance [15].

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 [19]. Under-determined mixture signals are usually separated by time-frequency masking methods [20], [21], which eliminate interference in certain time-frequency points.

D. ICA technique overview

Independent component analysis (ICA) is one of the most successful blind signal segmentation techniques. The goal of ICA is to find a linear representation of non-Gaussian signals so that the components are as statistically independent as

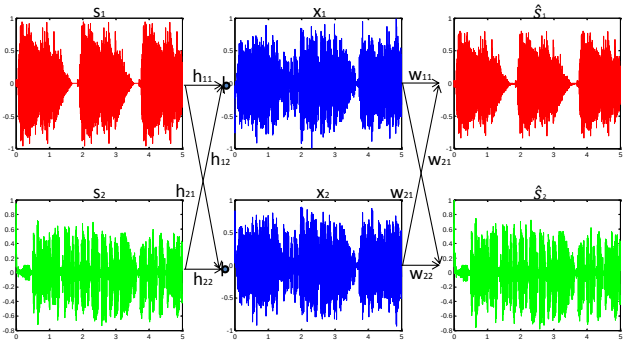


Figure 3. The ICA technique illustration.

possible. Take the two-sensor eavesdropping scenario shown in Fig. 2 as an example. 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 signal component as well as s_2 signal component. Due to the distance difference 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

$$\begin{aligned} x_1 &= h_{11}s_1 + h_{12}s_2 \\ x_2 &= h_{21}s_1 + h_{22}s_2 \ . \end{aligned}$$

Fig. 3 illustrates the basic intuition behind the ICA techniques. The red signal s_1 and green signal s_1 are shown in the left, and they are mixed through channels $h_{11}, h_{12}, h_{21}, h_{22}$. 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 non-linear uncorrelatedness; to be specific, we say $\hat{s}_1$ and $\hat{s}_2$ are independent, if any non-linear 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 non-linear transformations usually do not have zero covariance. By checking the non-linear uncorrelatedness of $\hat{s}_1$ and $\hat{s}_2$ computed from Equation 1, we can tell how good the estimated unmixing matrix $\boldsymbol{W}$ is.

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

Eventually, we want to find a matrix $\mathbf{W}$ 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 non-linear functions. The optimal matrix $\mathbf{W}$ is computed iteratively: after each iteration, $\mathbf{W}$ is updated by $\Delta\boldsymbol{W}$ using the following two learning rules, where $\boldsymbol{x}(t) = \boldsymbol{H} \cdot \boldsymbol{s}(t)$:

- The Bell's rule:

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

- The Amari's rule:

$$\Delta\boldsymbol{W} \propto [\boldsymbol{I} - f(\mathbf{x}(t)) \cdot \boldsymbol{s}(t)^T] \cdot \boldsymbol{W} \tag{3}$$

The matrix $\boldsymbol{W}$ is the estimate of the inverse of the mixing matrix $\boldsymbol{H}$ and the function f is a non-linear sigmoid function,

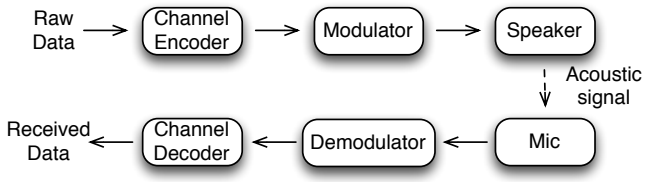


Figure 4. Acoustic communication module architecture.

e.g. we can choose $\tanh(\cdot)$ 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 [17] 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. 4. Its main components are the modulator and the demodulator. The narrow-sense BCH error correcting code 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

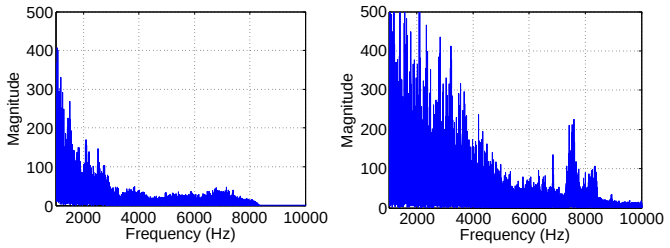
$$\begin{aligned} s(t) &= \Re\left(s_m(t)e^{i2\pi f_c t}\right), \quad m \in [0, M-1], \quad t \in [0, T] \\ &= \sqrt{\frac{2\mathcal{E}}{T}}\cos(2\pi f_c t + 2\pi m \Delta f t) \end{aligned}$$

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



(a) Traffic noise near a road. (b) Musical noise in a pub.

Figure 5. Ambient noises in frequency domain.

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 non-coherent detector. 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

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

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 $20 \sim 20000$ 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 Hz to 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. 5 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 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. 6 depicts the frequency response curve tested on Samsung Nexus S smartphone platform. The magenta line in Fig. 6 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. Sec. IV, below), the sender can adaptively select the optimal signal strength according to its current environmental noise level.

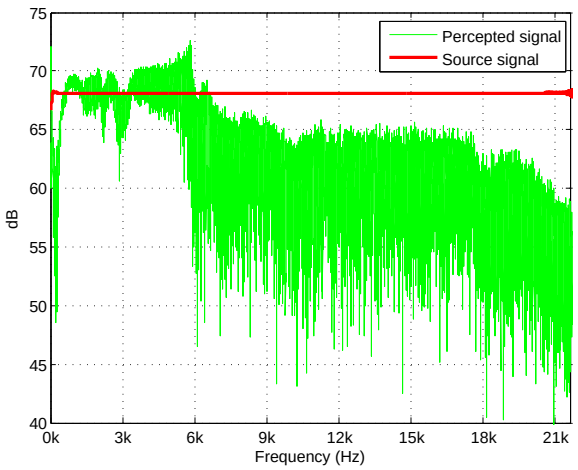


Figure 6. Frequency response (Samsung Nexus S).

To obtain current ambient noise level, the sender records 0.1-second 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, and 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 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 pre-defined to be $0.5, 1$ or 2 seconds in our system

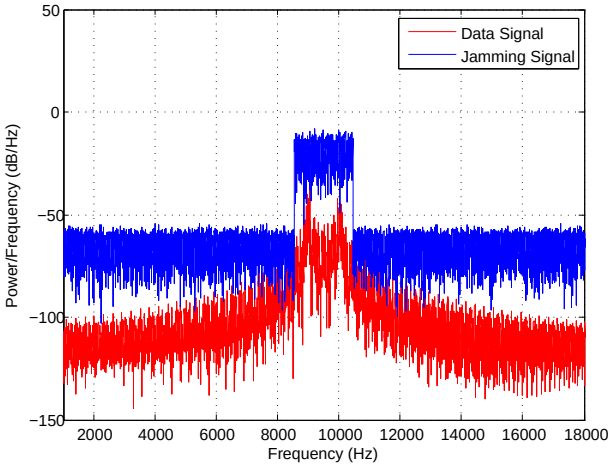


Figure 7. Power spectral density comparison.

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 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 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 $8.5 \sim 10.5$ kHz. Fig. 7 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 kHz 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 PCM 16-bit WAV files with sample rate 44.1 kHz, which requires roughly 10 M storage for $n_j = 20, \ell_s = 2$. 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 [13], 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 requires 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. Denote $p(f_i)$ as 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. 8. 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 received mixture signal. As illustrated in Fig. 9, 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 Evaluation

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. Sec. V, below 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 currently Android OS can return two values: 0 ('Near') and 5 ('Far'). The

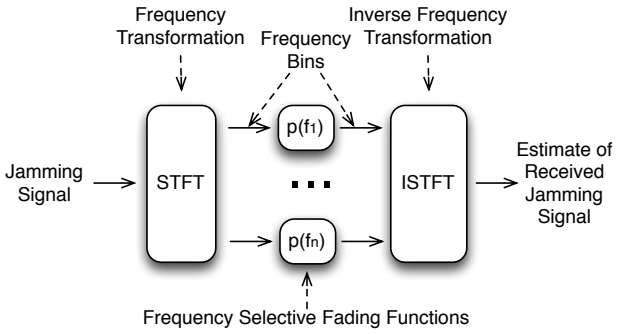


Figure 8. Frequency selective fading estimation.

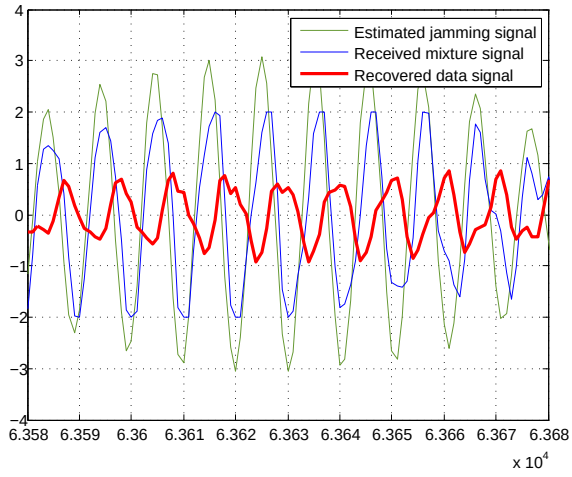


Figure 9. Recovering the data signal.

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);

threshold distances are different for various smartphone platforms, ranging from $1 \sim 2$ inches. 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'.

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, say 0.5 second. Alternatively, the sender can detect the presence of jamming signals itself. When its proximity sensor indicates 'Near', the sender records a 0.01s background sound sample and calculates root mean squared 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 samples in our system prototype. Once the jamming signal is detected, the sender starts to transmit (play) its modulated data signal. Note that

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 $\hat{m}$;

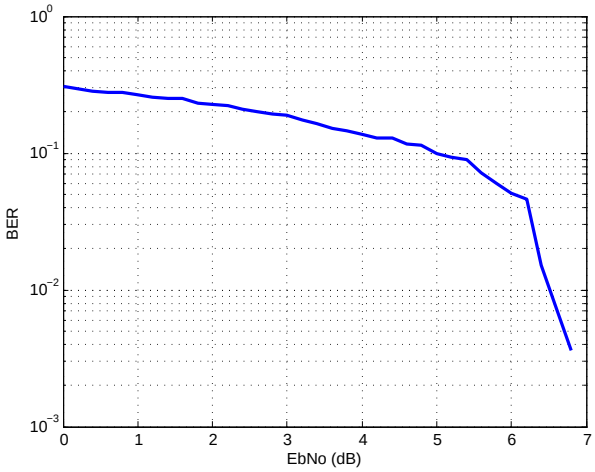


Figure 10. BER versus SNR for PriWhisper.

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 Alg. 1 and Alg. 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

Figure 11. PriWhisper in action (Samsung Galaxy S3 – Google Nexus S).

Table I. PERFORMANCE EVALUATION OF PRIWHISPER

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

before modulation. Fig. 10 shows the bit error rate (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. 11 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, 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 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 indicate the smartphone is static after its proximity sensor indicate '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. 12 depicts the remaining battery percentage after one-hour continuous PriWhisper acoustic communication between two Google Nexus 4 smartphones. As we can see, 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 is more computationally intensive than data modulation.

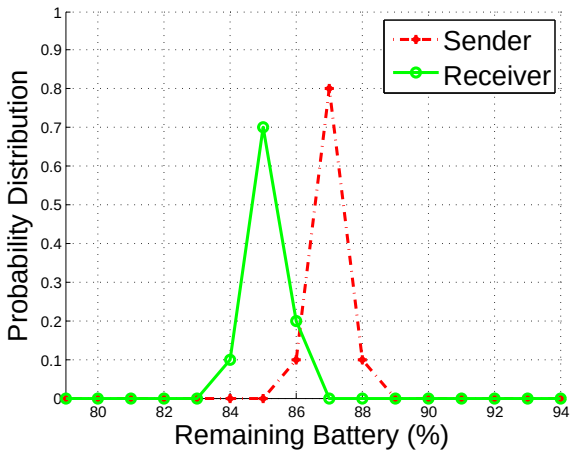


Figure 12. Battery Drain Experiment. (The remaining battery after one-hour continuous communication between Google Nexus 4 smartphones.)

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, and Table II shows the participant demographics details. 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.

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 by [22], using the difference of 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 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

$$x_i = h_{i1}s_1 + h_{i2}s_2 + e_i \ . \tag{4}$$

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. The secrecy capacity can be

Table II. PARTICIPANT DEMOGRAPHICS

Gender	Male: 56% Female: 44%
Age	18-25: 48% 26-30: 32% 31-45: 22% 46-75: 8%

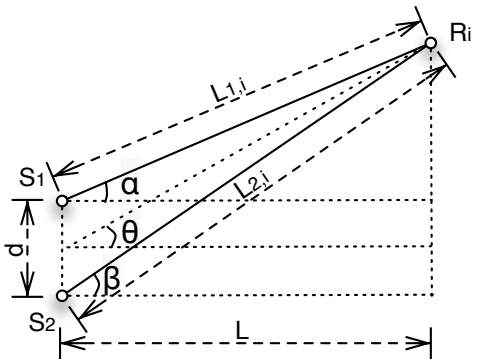


Figure 13. Two signal sources and one sensor in the LOS channel model.

expressed as

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

Hence, we can bound $I(R_i;S) < \kappa$, 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 less σ_e the higher $\tilde{\mathcal{C}}_{\mathsf{sec}}$ that our system can reach; therefore, it is favourable to operate PriWhisper in quite 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 the next section.

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 combination of each other. Fig. 13 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

$$\begin{aligned}\Delta L &= |L_{1,i} - L_{2,i}| \\ &= (L^2\tan(\theta) + d^2/4 + L^2 + Ld\tan(\theta)^{1/2} \\ &- (L^2\tan(\theta) + d^2/4 + L^2 - Ld\tan(\theta)^{1/2}) \\ &\approx \frac{dL}{\sqrt{8L^2 - 4\tan(\theta)Ld + d^2}}\ .\end{aligned}$$

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

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

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

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

Taking any two received mixture signals x_i and x_j in form of Equation 4, we have

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

Recall that PriWhisper adaptively adjusts the data signal strength according to the noise level, say EbNo $= 8$ dB in practice. (c.f. Sec. 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, 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 earch other, we define the channel similarity as

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

Fig. 14 plots the smoothed channel similarity curve $\varepsilon(d)$ for $d \in (0, 30]$ based 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, and temperature, etc. In practice, we assume the acoustic environment is a semi-reverberant field, in which the sound with carrier frequency $9 \sim 17$ kHz decays at least 10 dB when it passes the first 2 meters when the relative humidity is less than 50% temperature is above 15 C. Given the signal SNR$= 8$ dB and $\varepsilon < 0.03$, we can see the variances of the channel noise e_i, 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. 15, 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

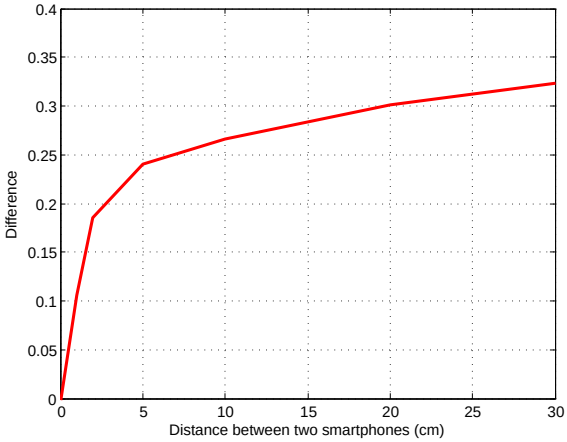


Figure 14. Channel similarity versus distance.

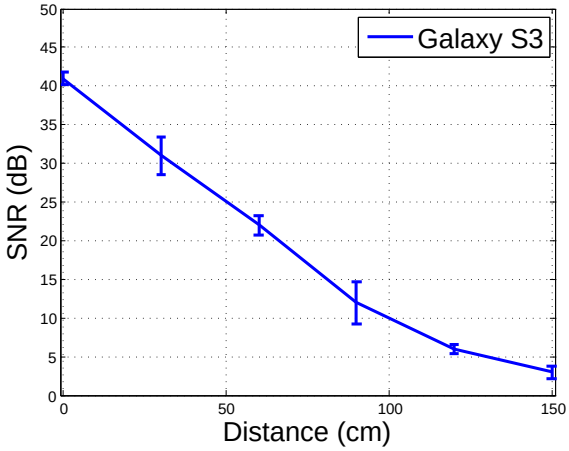


Figure 15. Outdoor acoustic signal decay experiment on Samsung Galaxy S3 (distance from 0 cm to 150 cm).

smartphone(s) use(s) the main (rear) speakers for acoustic communication, the actual distance between the signal sources are larger than the distance between these two smartphones. For example, when a Galaxy S3 and a Nexus S are aligned face-to-face at distance 0.5 cm, the distance between their two rear speakers are about 1.5 cm. Fortunately, the decibel level of the in-call speaker is sufficient for acoustic communication on almost all recent smartphone platforms. Since the distance is a very important security factor, both data signals and jamming single 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

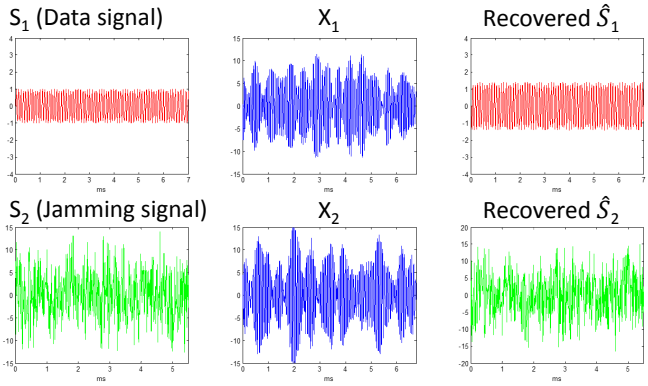


Figure 16. Successful attack instance (two signal sources are 30 cm away).

feasibility of signal segmentation using independent component analysis. 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 success, 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 decrease 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

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

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 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 m away from the communicating smartphones. As depicted in Fig. 16 and Fig. 17, 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.

D. Security against active adversaries

Finally, we will briefly discuss the security of PriWhisper against active adversaries. We emphasize

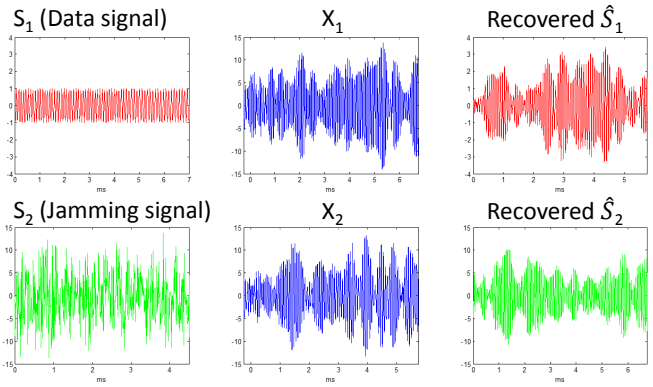


Figure 17. Failure attack instance (two signal sources are 1 cm away).

that our system is naturally resistant to many active attacks, e.g. man-in-the-middle attacks and jamming attacks. The carrier frequencies of PriWhisperlie in the audible bandwidth, so the acoustic short-range communication is noticeable by the users. To commit a jamming-like deny of service attack, the adversary has to generate and transmit the jamming signal around PriWhisper's carrier frequencies. Therefore, such jamming signals can be detected by human ears. In addition, the users can quickly locate the nearby jamming source, because acoustic signals cannot propagate over long distances. We note that some high-end directional speakers may be able to focus the jamming signal within a very small aperture angle. Hence, the adversary can jam an ongoing communication without letting the users hear the jamming sound. But the adversary's jamming device still has to be within 10 m range of the victims, and thus the users can locate the jamming source as usual once they notice that their communication has been jammed.

VI. Applications

PriWhisper has many potential security-sensitive smartphone applications. In this section, we present two useful examples: smartphone pairing and acoustic mobile payment system.

A. Smartphone pairing

Smartphone pairing is used to enable two smartphones, which share no prior common knowledge with each other, to agree on a security association that they can use to authenticate and protect their subsequent communication. The smartphone pairing process is essentially a key exchange process, which is a straightforward application of PriWhisper but yet an important primitive, which can be found useful in many mobile applications. After the pairing phase, the security association can be used to establish a secure connect between these two smartphones via some high-throughput wireless connections such as Wi-Fi.

In SiB pairing scheme [4], the short-range communication channel is only used as an authenticated OOB (A-OOB) channel. Therefore, the users still have to utilize the conventional Diffie-Hellman key exchange protocol. Alternatively, our PriWhisper channel can be viewed as an authenticated and secret OOB (AS-OOB) channel, so the pairing phase is much simpler in our case. Since both smartphones should contribute to their security association, our smartphone pairing scheme requires both smartphones to send each other their freshly generated random secret materials. Denote the secret material of smartphone Alice (or Bob) as $S_a \in \{0,1\}^\kappa$ (or $S_b \in \{0,1\}^\kappa$), where κ is a security parameter. During a smartphone pairing, Alice sends S_a to Bob, and then Bob sends S_b to Alice; after that, both smartphones return the shared key $S_{ab} = h(S_a, S_b)$.

We now give a concrete description for setup a 128-bit AES key. We set the session length to be 0.5 second. During a key exchange, Alice first sends S_a to Bob, and then Bob sends S_b to Alice. The smartphones will start to vibrate upon success, indicating users the termination of key exchange. The entire process takes approximately 1 second, and all the users need to do is simply tap their smartphones together face-to-face. After hearing two distinct noise-like short sound signals, the users separate their smartphones when they feel vibration.

B. Acoustic mobile payment system

Magnetic stripe cards are still widely used in many countries such as USA and China. As a second application example, we want to 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.

During a card payment, the customer swipe his/her magnetic stripe card at the POS terminal reader. The POS terminal reads all the information needed to complete this transaction from the magnetic stripe at one shot. A standard magnetic stripe bank card has three tracks, but Track 3 is not commonly used. According to ISO/IEC 7811 and 7813 standards, Track 1 contains 76 alphanumeric characters (7 bit per character including parity bit), and Track 2 contains 37 numeric characters (5 bit per character including parity bit). (See App. A for data format details.) A POS terminal only reads either Track 1 or Track 2 to process a payment, which requires 532 bit or 185 bit card-to-terminal data transmission.

It is easy to store the Track 1 and Track 2 data of the users' magnetic stripe cards in their smartphones and utilize PriWhisper for an acoustic contactless mobile payment. Subsequently, the user can get rid of a deck of various magnetic stripe bank cards and use his/her smartphone instead. To make a payment, all he/she have to do is tapping the front of his/her smartphone on a microphone and speaker enabled terminal. Meanwhile, the user's smartphone securely sends the card information to the terminal using acoustic signal and the terminal completes the transaction as usual. The entire process only takes about 1 second, which is much faster than the time that an average user takes to swipe a magnetic stripe card himself/herself.

In addition, PriWhisper based acoustic mobile payment system can also support *EMV Contactless Mobile Payment* specification [23]. The smartphone can interact with the terminal through Application Protocol Data Unit (APDU) command and APDU response pairs. A transaction usually consists of several-round message exchanges, and the length of each

APDU command/response message is typically only a few bytes. Since all smartphones have a speaker and a microphone, while very few smartphones are equipped NFC chips, we believe that PriWhisper may enjoy a higher penetration rate than NFC the mobile payment market.

VII. Related Work

To our best knowledge, friendly jamming technique was first proposed by Negi and Goel [24] 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.* [13] first extend friendly jamming technique to a single full-duplex receiver in 2011. Their system uses a specialized receive antennas that is connected with a transmit chain for jamming signal cancellation. Recently, Tippenhauer *et al.* [14] indicate that there is a limitation on friendly jamming techniques. They show that some friendly jamming systems such as [13] are vulnerable to nearby MIMO eavesdroppers, but our acoustic communication system does not subject to their attack. Bursztein *et al.* [25] utilize blind signal segmentation techniques to attack noise-based non-continuous 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 [26] present an aerial acoustic communication system using software modem in 2001. Mostafa [27] released a software modem called minimodem that supports many traditional modem protocols, e.g. Bell 103 on Linux OS. Michel [28] implemented a software modem for Android system supporting ASK modulation, and it can modulate data in musical tones. Houmansadr *et al.* [29] 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 [30] and navigation systems [31]. Recently, there is a trend of utilizing acoustic communication technologies in mobile payment systems, e.g. [32] and [33]; hence, we believe PriWhisper could be a great candidate for acoustic smartphone communication with build-in security mechanisms.

PriWhisper **v.s. NFC:** NFC requires additional hardware and thus it is not widely supported by various smartphone 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 blind signal segmentation attacks. We also presented two useful PriWhisper application examples: smartphone key exchange and acoustic mobile payment system. 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 OS's 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.

References

[1] Google, "Google Wallet," URL: http://www.google.com/wallet/index.html, accessed: 2013-01-01.

[2] S. Millward, "AliPay's Mobile Barcode Payments in China," URL: http://www.techinasia.com/alipay-mobile-payments/, accessed: 2013-01-01.

[3] R. Kim, "PayPal's Barcode-based Payment Services in UK," URL: http://gigaom.com/2012/05/30/paypal-rolls-out-barcode-payments-in-the-uk/, accessed: 2013-01-01.

[4] A. P. Jonathan M. Mccune and M. K. Reiter, "Seeing-is-believing: Using camera phones for human-verifiable authentication," in *In IEEE Symposium on Security and 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 *SOUPS '09*. ACM, 2009.

[6] J. Guerrieri and D. Novotny, "HF RFID Eavesdropping and Jamming Tests," in *Report*, 2012.

[7] "Norm ECMA-385. NFC-SEC: NFCIP-1 Security Serices and Protocol," 2010.

[8] "Norm ECMA-386. NFC-SEC-01: NFC-SEC Cryptography Standard using ECDH and AES Reference," 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 Communications Surveys and Tutorials*, vol. 13, no. 3, pp. 487–502, 2011.

[10] R. Headrick and L. Freitag, "Growth of underwater communication technology in the U.S. Navy," *Comm. Mag.*, vol. 47, no. 1, pp. 80–82, 2009.

[11] R. Jurdak, C. V. Lopes, and P. Baldi, "Software acoustic modems for short range mote-based underwater sensor networks," in *IEEE Oceans Asia*, 2006.

[12] S. Goel and R. Negi, "Guaranteeing Secrecy using Artificial Noise," *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, pp. 2180–2189, 2008.

[13] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu, "They can hear your heartbeats: non-invasive security for implantable medical devices," in *SIGCOMM*, 2011.

[14] N. O. Tippenhauer, L. Malisa, A. Ranganathan, and S. Capkun, "On Limitations of Friendly Jamming for Confidentiality," in *S & P (Oakland)*, 2013.

[15] 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, 2001.

[16] A. Cichocki, J. Karhunen, W. Kasprzak, and R. Vigario, "Neural networks for blind separation with unknown number of sources," *Neurocomput*, vol. 24, no. 1, p. 5593, 1999.

[17] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain," *Neurocomputing*, vol. 15, pp. 745–770, 1998.

[18] J. Cardoso, "Blind source separations: statistical principles," vol. 9, no. 10, 1998, pp. 2009–2025.

[19] M. Reyes-Gomez, B. Raj, and D. Eliss, "Multi-channel source separation by factorial hmms," 2003, pp. 664–667.

[20] S. T. Roweis, "One microphone source separation," *Advances in Neural Information Processing Systems (NIPS13)*, pp. 793–799, 2001.

[21] L. Benaroya, L. McDonagh, F. Bimbot, and R. Gribonval, "Non-negative sparse representation for wiener based source separation with a single sensor," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 613–616, 2004.

[22] I. Csiszar and J. Korner, "Broadcast Channels with Confidential Messages," *IEEE Transactions on Information Theory*, pp. 339–348, 1978.

[23] EMVCo, "EMVCo Mobile Contactless - EMV Profiles of GlobalPlatform UICC Configuration," 2010.

[24] R. Negi and S. Goel, "Secret communication using artificial noise," in *IEEE Vehicular Technology Conference*, 2005.

[25] E. Bursztein, R. Beauxis, H. Paskov, D. Perito, C. Fabry, and J. C. Mitchell, "The Failure of Noise-Based Non-continuous Audio Captchas," in *S & P (Oakland)*, 2011.

[26] C. Lopes and P. Aguiar, "Aerial acoustic communications," in *IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, 2001.

[27] K. Mostafa, "Minimodem," URL: http://www.whence.com/minimodem/, accessed: 2013-01-01.

[28] J. Michel, "Mobile Acoustic Modems in Action," URL: https://code.google.com/p/mobile-acoustic-modems-in-action/, accessed: 2013-01-01.

[29] 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 *NDSS*, 2013.

[30] C. Lopes and P. Aguiar, "Acoustic modems for ubiquitous computing," *IEEE Pervasive Computing*, vol. 2, no. 3, pp. 62–71, Jul. 2003.

[31] L. Freitag, M. Grund, I. Singh, J. Partan, P. Koski, K. Ball, and W. Hole, "The whoi micro-modem: an acoustic communications and navigation system for multiple platforms," in *IEEE OCEANS Conf. Exhib.*, 2005.

[32] CNET, "Naratte: Mobile payments using sound waves," URL: http://news.cnet.com/8301-19882_3-20072295-250/naratte-mobile-payments-using-sound-waves/, accessed: 2013-01-01.

[33] Alipay, "Sound wave mobile payment," URL: http://techcrunch.com/2013/04/14/alipay-launches-sound-wave-mobile-payments-system-in-beijing-subway/, accessed: 2013-04-15.

APPENDIX

The data format is specified by ISO 7810, 7811 and 7813 standards. For completeness, we briefly provide the information contained in a magnetic stripe bank card. There are three data tracks in each magnetic stripe stored at density 210 bits per inch.

There are 76 alphanumeric data characters in Track 1. From left to right, its data format is as follows.

- SS: Start Sentinel, displayed by symbol '%'.
- FC: Format Code, 2 digits.
- PAN: Primary Account Number, max. 19 digits.
- FS: Field Separator, displayed by symbol '^'.
- NAME: the cardholder's name, max. 26 characters.
- FS: Field Separator, displayed by symbol '^'.
- ADDITIONAL DATA: 4-digit expiration data (YYMM) and 3-digit service code.
- DISCRETIONAL DATA: 1-digit PVKI (PIN Verification Key Indicator), 4-digit PVV (PIN Verification Value) or Offset, and 3-digit CVV (Card Verification Value) or CVC (Card Validation Code).
- ES: End Sentinel, displayed by symbol '?'.
- LRC: Longitudinal Redundancy Check character.

There are 37 numeric data characters in Track 2. From left to right, its data format is as follows.

- SS: Start Sentinel, hex B, displayed by symbol ';'.
- FC: Format Code, 2 digits.
- PAN: Primary Account Number, max. 19 digits.
- FS: Field Separator, hex D, displayed by symbol '='.
- ADDITIONAL DATA: 4-digit expiration data and 3-digit service code.
- DISCRETIONAL DATA: 1-digit PVKI, 4-digit PVV or Offset, and 3-digit CVV or CVC.
- ES: End Sentinel, hex F, displayed by symbol '?'.
- LRC: Longitudinal Redundancy Check character.

There are 104 numeric data characters in Track 3. From left to right, its data format is as follows.

- SS: Start Sentinel, hex B, displayed by symbol ';'.
- FC: Format Code, 2 digits.
- PAN: Primary Account Number, max. 19 digits.
- FS: Field Separator, hex D, displayed by symbol '='.
- ADDITIONAL DATA: 3-digit country code (optional), 3-digit currency code, 1-digit currency exponent, 4-digit amount authorized per cycle, 4-digit amount remaining this cycle, 2-digit service restriction, and 9-digit card security number (opinion), etc.
- DISCRETIONAL DATA: first subsidiary Acc. No. (optional), secondary subsidiary Acc. No. (optional), 1-digit relay marker, 6-digit cryptographic check digits (optional), and discretionary data.
- ES: End Sentinel, hex F, displayed by symbol '?'.
- LRC: Longitudinal Redundancy Check character.