# Dolphin: Real-Time Hidden Acoustic Signal Capture with Smartphones

Man Zhou, *Student Member, IEEE*, Qian Wang, *Member, IEEE*, Kui Ren, *Fellow, IEEE*,
Dimitrios Koutsonikolas, *Senior Member, IEEE*, Lu Su, *Member, IEEE*,
and Yanjiao Chen, *Member, IEEE*

**Abstract**—Dual-channel screen-camera communication has been proposed to enable simultaneous screen viewing and hidden screen-camera communication. However, it strictly requires a well-controlled camera-screen alignment and an obstacle-free access. In this paper, we propose Dolphin, a novel real-time acoustics-based dual-channel communication system. Leveraging masking effects of human auditory system and readily available audio signals, Dolphin enables real-time unobtrusive speaker-microphone data communication without affecting the primary audio-hearing experience of human users. Compared with screen-camera communication, Dolphin supports non-line-of-sight transmissions and more flexible speaker-microphone alignments. Dolphin can also automatically adapt the data rate to various channel conditions. We further develop a secure data broadcasting scheme on Dolphin, where only designated privileged users can recover the embedded information in the acoustic signals. Our Dolphin prototype, built using COTS (Commercial Off-The-Shelf) smartphones, realizes (potentially secure) real-time hidden information communication, supports up to 8-meter signal capture distance and $\pm 90^\circ$ listening angle, and achieves an average goodput of 240 bps at 2 m.

**Index Terms**—Speaker-microphone communication, unobtrusive acoustic communication, dual-mode communication

## 1 Introduction

With the ever-increasing popularity of smart devices in our daily lives, people are more and more dependent on them for retrieving and distributing information in the cyber-physical world. At the meantime, many broadcasting media equipped with screens and speakers, e.g., stadium screens & sports displays, advertising electronic boards, TVs, desktop/tablet PCs, and laptops, have become a readily available information source for human users. According to Sandvine's semiannual "Global Internet Phenomena report" [3], video and audio streaming accounts for more than 70 percent of all broadband network traffic in North America during peak hours. Under this trend, dual-channel communication is desirable, where the screens and speakers convey primary information through videos and audios to human users as well as deliver hidden customized information to their smart devices. For example, a sports fan who watches NBA live streams on the stadium screen may also receive statistics for players and teams on his/her smart device without resorting to the Internet. Another real-life example could be a person watching advertisements on TV while receiving instant notifications, offers, and promotions on his/her device.

In the existing dual-channel screen-camera communication, the side information is either directly displayed on top of the video content or encoded into visual patterns shown on the screen. Nevertheless, the coded images on the screen (reserved for devices) interfere with the primary content (reserved for users), leading to unpleasant and distracting viewing experience for human users. Recent research endeavors [4], [5], [6], [7], [8], [9], [10] have tried to mitigate this contention between users and devices by developing real-time unobtrusive screen-camera communication techniques that allow the screen to concurrently display contents to users and communicate side information to devices.

However, a screen-camera communication system has several practical limitations in real-world scenarios. First, it needs to control screen-camera alignment and is highly sensitive to device shaking, which undermines the flexibility of the system. Second, it requires a direct visible communication path between the screen content and the camera capture window. In other words, the screen-camera system is restricted to line-of-sight (LOS) communication. If there are obstacles or moving objects between the screen and the camera, the camera cannot capture or decode any useful information from the screen. Third, the communication/viewing distance is restricted by the size of the screen, which cannot be freely adjusted once deployed.

In this paper, we take a new and different approach, proposing a novel acoustics-based real-time dual-channel communication system, Dolphin, which leverages speaker-microphone links rather than screen-camera links. Dolphin

- M. Zhou and Q. Wang are with the School of Cyber Science and Engineering, Collaborative Innovation Center of Geospatial Technology, Wuhan University, Wuhan, Hubei 430072, China. E-mail: {zhouman, qianwang}@whu.edu.cn.
- K. Ren, D. Koutsonikolas, and L. Su are with the Department of Computer Science and Engineering, University at Buffalo, The State University of New York, New York, NY 14260. E-mail: {kuiren, dimitrio, lusu}@buffalo.edu.
- Y. Chen is with the School of Computer Science, Wuhan University, Wuhan, Hubei 430072, China. E-mail: chenyanjiao@whu.edu.cn.

*Manuscript received 2 Jan. 2018; revised 28 Apr. 2018; accepted 20 May 2018. Date of publication 1 June 2018; date of current version 4 Feb. 2019.*
*(Corresponding author: Qian Wang.)*
*For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.*
*Digital Object Identifier no. 10.1109/TMC.2018.2842771*

generates composite audio by multiplexing the original audio (intended for the human listener) and the embedded data signals (intended for smartphones). The composite audio, played by the speaker, does not affect users' listening experience of primary contents while delivering hidden data that can be captured and extracted by the smartphone microphones. We further consider a secure communication scenario, where the embedded data is intended for a set of designated privileged users. For example, only VIP users can gain the promotion code of goods or the permission of online premium movies. To address this issue, we develop a secure data broadcasting scheme on Dolphin based on broadcast encryption [11].

The inherent properties of audio signals help overcome the limitations of screen-camera communication systems. First, compared with the highly-directional visible light beams, the sound transmits at all directions and thus renders a broader signal receiving angle. Second, the sound bypasses obstacles by diffraction and reflection while visible light can easily be blocked. Third, the communication distance of speaker-microphone links can be adjusted by changing the speaker volume, while the fixed screen size limits the communication distance of screen-camera links. Moreover, the acoustics-based system is also robust to mild device motion.

The design of Dolphin addresses three major challenges. First, there is a tradeoff between audio quality and signal robustness. While a stronger embedded signal can resist the speaker-microphone channel interference, it may also be obtrusive to the human ear. To seek the best tradeoff, we propose an adaptive signal embedding approach, which carefully chooses the modulation method and the embedded signal strength according to the energy characteristics of the carrier audio. Second, the speaker-microphone links suffer from serious distortion caused by both the acoustic channel (e.g., ambient noise, multi-path interference, device mobility) and the smartphone hardware limitations (e.g., the frequency selectivity of the microphone). To combat ambient noise and multi-path interference, we adopt OFDM for the embedded signals and determine the system parameters according to the characteristics of speaker-microphone links. We further adopt channel estimation based on a hybrid-type pilot arrangement to minimize the impact of frequency-time selective fading and Doppler frequency offset. Third, several users may receive side information simultaneously, but the link qualities of different users are highly diverse due to many factors, such as smartphone models, angles, and distances. Therefore, it is necessary to perform adaptive information transmission. However, speaker-microphone links have no feedback channel. To address this problem, we adopt the two-tier rateless coding scheme [12], [13] to design a Bi-level rateless code-based orthogonal error correction (ROEC) scheme to automatically adapt the data rate to various channel conditions, such that each receiver achieves as high a goodput as its link quality allows.

We built a Dolphin prototype using a HiVi M200MKIII loudspeaker as the sender and different smartphone models as receivers. We evaluated the user perception, practical considerations, data communication performance, and extra overhead of SDB scheme. The results show that Dolphin is able to achieve an average goodput of 240 bps over various audio contents at 2 m. Our prototype supports a range of up to 8 m and a listening angle of up to $\pm 90^\circ$ (given the reference point facing the speaker) when the speaker volume is 80 dB. Furthermore, Dolphin realizes real-time hidden data transmissions with an average symbol encoding time of 1.35 ms and an average symbol decoding time of $25.3 \sim 37.9$ ms on different smartphones.

The main contributions of this work are summarized as follows.

- We propose Dolphin, a novel dual-channel real-time unobtrusive speaker-microphone communication, which allows information data streams to be embedded into audios and transmitted to smartphones without affecting the primary audio-hearing experience of human users.
- We propose an adaptive embedding approach based on OFDM and energy analysis of the carrier audio signals. To enhance Dolphin's robustness and reliability, we leverage pilot-based channel estimation during signal extraction and leverage two-tier rateless coding to automatically adapt the data rate to various channel conditions, supporting both real-time and offline decodings.
- We develop and implement a secure data broadcasting scheme on Dolphin, where the embedded data can only be recovered by designated privileged users. The sender can flexibly determine the set of targeted users by adding or revoking members.
- We build a Dolphin prototype using COTS smartphones and verify that it enables flexible data transmissions in real-time unobtrusively atop arbitrary audio content.

## 2 Related Work

*Unobtrusive Screen-Camera Communication.* In recent years, extensive research efforts have been made on designing special color barcodes for barcode-based VLC [14], [15], [16], [17], [18], [19]. To resolve resource contention, several recent studies seek to achieve unobtrusive screen-to-camera communication. [4], [7] leveraged the temporal flick-fusion of human vision to switch barcodes with complementary hues. PiCode and ViCode [5] integrated barcodes with existing images to enhance viewing experience. [6] proposed to embed data hidden in brightness changes upon two consecutive frames. Hilight [8] encoded bits into the pixel translucency change by leveraging the alpha channel. In [9], the authors encoded the data as complementary intensity changes over RGB color channels. [10] developed content-adaptive encoding techniques that exploit visual features such as edges and texture to unobtrusively communicate information. Compared to Dolphin, unobtrusive screen-to-camera communication requires well-controlled screen-camera alignment and obstacle-free access.

*Aerial Acoustic Communication.* Aerial acoustic communication over speaker-microphone links has been studied in [20], [21], [22], [23], [24], [25], [26], [27], [28]. Dhwani [20] and PriWhisper [21] aim to realize secure acoustic short-range communication on mobile phones. In [22], chirp signals were used to realize an aerial acoustic communication system. In [24], the authors proposed to hide information in audios and use the loudspeaker and the microphone with flat frequency

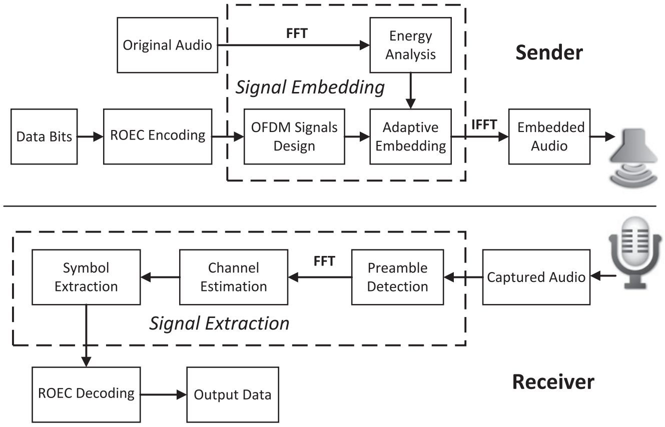


Fig. 1. System architecture of Dolphin.

response to display and record data-embedded audio. The feasibility of ultrasonic data communication between devices that are implanted or worn on the human body was investigated in [25]. [26] proposed a near-ultrasound chirp signal-based communication for the TV's 2nd screen services. BackDoor [27] leveraged the acoustic non-linearity in microphone hardware to enable ultrasound recordable by unmodified microphones. DSSS modulation was leveraged in [28] for sound-based communication where both the transmitter and the receiver are mobile devices. However, [20], [21] only focus on reliable speaker-microphone data communication, while the system in [24] was not designed for off-the-shelf smartphones without considering the characteristics of acoustic channel, and [22], [23], [26] used the inaudible audio signals to achieve very low-rate communication. In contrast, Dolphin aims at establishing dual-mode unobtrusive communication using off-the-shelf smartphones.

*Audio Watermarking*. With the development of network and digital technologies, digital audio is easy to be reproduced and retransmitted. Audio watermarking [29], [30], [31], [32], [33], as a means to identify the owner, encodes hidden copyright information into the digital audio. Unlike audio watermarking which directly provides copyright embedded audio files to users and aims to ensure copyright information cannot be removed, Dolphin seeks to enable unobtrusive data communication and provides relevant side information which users can obtain through their smartphones when the speaker plays the audio, by addressing several challenges unique to the nature of the acoustic signal propagation and speaker-microphone characteristics. Therefore, Dolphin needs to address real-world signal degradation over the speaker-microphone channel while watermarking does not. To achieve our goal, modifying the original audio and decoding the signals in Dolphin must take into account ambient noise, the characteristics of commercial speakers and microphones, and channel estimation.

# 3 Dolphin Design

The design of Dolphin is based on the properties of human auditory system [34] and the characteristics of acoustic speaker-microphone channel. To ensure that the embedded data stream does not affect the user perception of the original audio content, we utilize the "auditory masking effects". Due to space limitation, interested readers can refer to our preliminary version of this paper [1] for more details.

## 3.1 System Overview

Fig. 1 illustrates the system architecture of Dolphin which consists of two parts: the sender (e.g., a TV) and the receiver (e.g., a smartphone).

*The Sender*. Raw data bits are encapsulated into packets, and bits in each packet are encoded by ROEC codes (Section 3.4), divided into symbols, and further modulated by OFDM. We analyze the original audio stream on the fly to locate the appropriate parts to carry the embedded information packets. First, we perform energy distribution analysis to select the subcarrier modulation method for each packet. Then, we perform energy analysis on every audio segment that corresponds to a symbol. If the energy of a segment is high enough to mask the embedded signals, we adaptively embed the symbol into the segment according to its energy characteristics; otherwise, we do not perform any modifications. Finally, the sender transmits the data-embedded audio via the speaker.

*The Receiver*. After the audio is captured by the smartphone microphone, we first detect the preamble of each packet. Then, we partition the audio into segments that correspond to symbols. Since signals usually suffer severe frequency-time selective fading over the speaker-microphone link, we perform channel estimation before symbol extraction to improve decoding rate. Finally, we convert audio signals into symbols, extract the data bits in each symbol, and recover the original data after ROEC decoding.

## 3.2 Signal Embedding

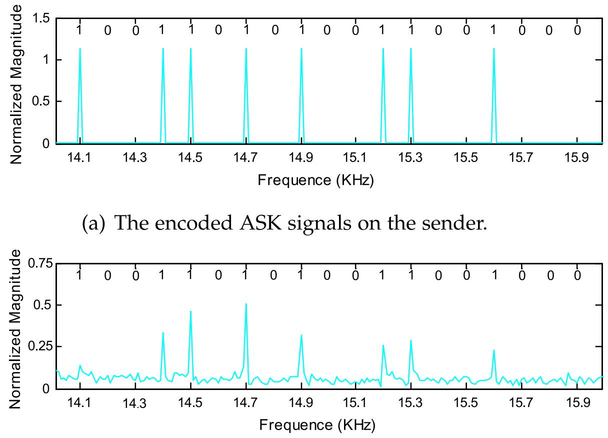
### 3.2.1 OFDM Signal Design

We adopt orthogonal frequency division multiplexing (OFDM) in Dolphin to combat frequency-selective fading and multi-path interference.

*Operation Bandwidth Selection*. The response frequency of most COTS speakers and microphones is from 50 Hz to 20 kHz, and the interference from the ambient noise is negligible when the frequency exceeds 8 kHz [1]. It has been shown that the frequencies in the bandwidth of 17 kHz$\sim$20 kHz are mostly inaudible [23], thus a small amount of energy of the original audio can mask the embedded signals. Since this bandwidth is relatively limited, we also propose to use the bandwidth below 17 kHz to improve throughput. Finally, we choose 8 kHz$\sim$20 kHz as the frequency bandwidth for the embedded data.

*Symbol Sub-Carrier Modulation*. The unpredictable phase shifts due to non-ideal synchronization of the preamble [1] makes PSK unsuitable for Dolphin. The limited subcarrier width in OFDM makes it hard to decode FSK-modulated signals. Hence, Dolphin uses ASK to modulate the signals on each subcarrier.

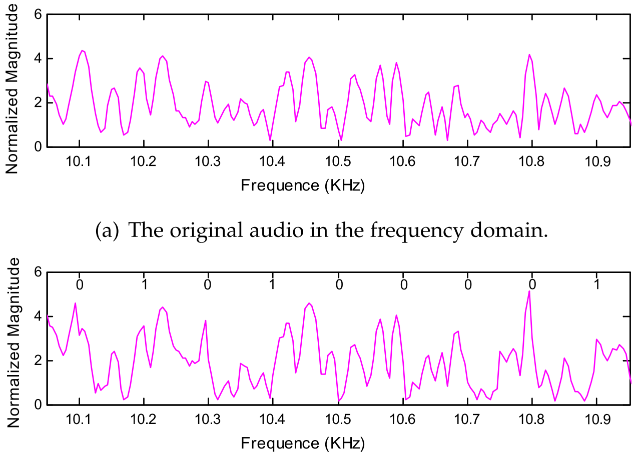
To ensure the embedded data stream is unobtrusive to the human ear, we cannot embed strong signals into a subcarrier. Therefore, we use a special form of ASK, namely On-Off Keying (OOK). As shown in Fig. 2a, the embedded signals appear as peaks in the frequency domain. To decode the embedded data, the receiver must set a threshold to determine the existence of peaks on the subcarrier. However, selecting this threshold is difficult due to speaker-microphone frequency selectivity and channel interference. As shown in Fig. 2b, peaks may be distorted or even disappear. A drawback of ASK is that the energy distribution of the original audio in the



(a) The encoded ASK signals on the sender.

(b) The captured ASK signals on the receiver.

Fig. 2. Amplitude shift keying signals.



(a) The original audio in the frequency domain.

(b) The encoded EDK signals.

Fig. 3. Energy difference keying signals.

bandwidth that we embed data must be very low. Hence, we cut off the energy of the original audio in these bandwidths before embedding data bits. In order to make the changes as unobtrusive as possible, we only use ASK to embed data in 14 kHz~20 kHz, which means we cut off the energy of the original audio beyond 14 kHz.

If the energy distribution of the original audio is relatively high in the bandwidth of embedded data, we use a different modulation method called energy difference keying (EDK) instead of ASK. EDK adjusts the energy distribution around the central frequency of the subcarrier to represent bits 0 and 1. For example, higher energy on the left of the subcarrier central frequency indicates 0, and higher energy on the right of the subcarrier central frequency indicates 1, as shown in Fig. 3. Since the energy of the original audio is usually concentrated on the bandwidth that is much lower than 15 kHz, we only use EDK to embed data in 8 kHz~14 kHz. To mitigate speaker-microphone frequency selectivity and channel interference, the diversity of the energy on the left and right side of the central frequency must be sufficiently large. Therefore, we adjust the energy in a continuous frequency band $B_{si}$ around the subcarrier central frequency rather than at some discrete frequencies. To guarantee the same level of robustness, the change of energy distribution in the original audio with EDK is usually larger than that with ASK. But in EDK, we do not need to cut off the energy in certain bandwidths of the original audio. In addition, since the frequencies in the left and right subcarriers are very close, the energy adjustment is hard to perceive. Hence, EDK is suitable in cases where the original audio has relatively high energy in high frequencies.

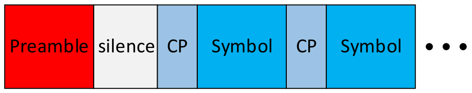


Fig. 4. Dolphin packet format.

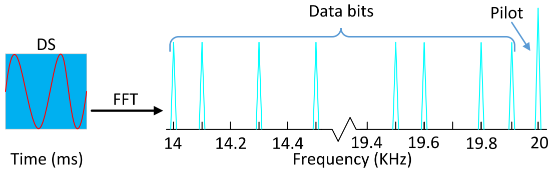


Fig. 5. The data bits of an amplitude shift keying symbol.

*Packet Format Design*. For the convenience of data encoding/decoding, we divide the embedding data stream into packets. As shown in Fig. 4, a packet consists of a preamble and a number of symbols (the number of symbols in a packet is set to 30 in our implementation), each preceded by a cyclic prefix (CP). The preamble is used to synchronize the packet, and the symbols contain the data bits.

Following the approach of previous aerial acoustic communication systems (e.g., [20] and [22]), we use a chirp signal as the preamble. Its frequency ranges from $f_{min}$ to $f_{max}$ in the first half of the duration and then goes down to $f_{min}$ in the second half. In our implementation, we chose $f_{max}$ = 19 kHz and $f_{min}$ = 17 kHz, and the duration of the preamble is 100 ms. Due to its high energy, we pad each preamble with a silence period of 50 ms to avoid interference to data symbols.

The data bits in a symbol are integrated into a small piece of audio. As shown in Fig. 5, after a symbol signal is converted from the time domain to the frequency domain, 60 subcarriers in the range 14 kHz~19.9 kHz are used to encode the data bits, and the signal in 20 kHz is a pilot used for time-selective fading and Doppler frequency offset estimation. The pilot is easy to detect as it lies on the rightmost of the symbol spectrum. To estimate the frequency-selective fading, we insert additional pilots on each subcarrier of the first symbol. Longer data symbol durations and fewer subcarriers increase the decoding rate but reduce the throughput. In our experiments (Section 5.3), we found that a duration of 100 ms and a total of 60 subcarriers achieve a good tradeoff between robustness and throughput.

In RF OFDM radios, a cyclic prefix (CP) is designed to combat Inter-Symbol Interference (ISI) and Inter-Carrier Interference (ICI). A certain length of signals at the end of the symbol is copied and inserted in front of the symbol. We adopt the cyclic prefix in acoustic OFDM to combat ISI and ICI. In our implementation, the CP duration is set to be 10 ms.

### 3.2.2 Energy Analysis

We perform energy distribution analysis to select the subcarrier modulation method (ASK or EDK) for each packet. Let $f$ (in kHz) denote the frequency, $F(f)$ denote the normalized signal magnitude at frequency $f$, $l$ denote the number of sampling points in a packet, $F_s$ denote the sampling rate, and $\Delta f_{(f_i,f_j)}$ denote the bandwidth of the frequency band $f \in [f_i, f_j]$ ($f_i$ and $f_j$ are in kHz). The average energy spectrum density (ESD) of the audio corresponding to a packet $E_{pt}$ is calculated as:

$$E_{pt} = \frac{l \cdot \sum_{f=0}^{20} |F(f)|^2}{2 \cdot F_s \cdot \Delta f_{(0,\ 20)}}. \tag{1}$$

The average energy spectrum density in the lower frequency band $E_{pl}$ is calculated as:

$$E_{pl} = \frac{l \cdot \sum_{f=0}^{8} |F(f)|^2}{2 \cdot F_s \cdot \Delta f_{(0,\ 8)}}. \tag{2}$$

Similarly, the average energy spectrum density in the higher frequency band $E_{ph}$ is calculated as:

$$E_{ph} = \frac{l \cdot \sum_{f=8}^{20} |F(f)|^2}{2 \cdot F_s \cdot \Delta f_{(8,\ 20)}}. \tag{3}$$

The default modulation method is ASK. We switch to EDK if the energy distribution satisfies the following two conditions, based on two thresholds $E_{high}$ and $R_{hl}$:

$$E_{ph} > E_{high} \quad \text{and} \quad \frac{E_{ph}}{E_{pl}} > R_{hl}. \tag{4}$$

In our implementation, we empirically set $E_{high}$=$10^{-7}$ J/Hz and $R_{hl}$=$\frac{1}{700}$. We embed a control signal at 19.6 kHz into each preamble to indicate the selected modulation method to the receiver.

It is observed that the voice is intermittent in the time domain [1] due to speech pauses, and the music volume often changes with time. Therefore, if we embed a data symbol into a piece of low-volume audio, it will be easily perceived by the user. To tackle this problem, we perform energy analysis on every piece of audio corresponding to a symbol. The calculation of the average ESD of a symbol is similar to that of a packet. Let $E_{st}$, $E_{sl}$, and $E_{sh}$ denote the ESD of the whole frequency band, the lower frequency band, and the higher frequency band, respectively. We embed symbol bits into a piece of audio only when the average energy of this audio piece $E_{st}$ is higher than a threshold $E_{min}$, which measures the minimum audio energy spectrum density needed by the data symbol.

Choosing a larger $E_{min}$ will enhance audio quality, but it also means that fewer audio pieces can be used for data embedding. Based on our subjective perception experiments and energy statistics of audio pieces, we set $E_{min} = 2 \times 10^{-8}$ J/Hz. The receiver only needs to detect the pilot in 20 kHz to know whether this piece of audio is embedded with data bits or not.

### 3.2.3 Adaptive Embedding

To avoid the embedded signals to be obtrusive, we leverage auditory masking effects of human ears. Due to the temporal masking effects, a low noise is often unobtrusive when the energy of the original audio is very high, but will be perceived when the energy of the original audio is low. We propose an adaptive signal embedding approach, which carefully chooses the embedded signal strength according to the energy characteristics of the carrier audio. The energy of embedded signals is adapted with the average energy of the carrier audio as follows:

1) For ASK, the embedded signal energy of a symbol is calculated as:

$$E_{am} = \begin{cases} N \cdot \beta^2 E_{sl}, & E_{sl} < E_{max}, \\ N \cdot \beta^2 E_{max}, & E_{sl} \geq E_{max}. \end{cases} \tag{5}$$

2) For EDK, the embedded signal energy of a symbol is calculated as:

$$E_{en} = \begin{cases} N \cdot \beta^2 E_{sl} B_{si}, & E_{sl} < E_{max}, \\ N \cdot \beta^2 E_{max} B_{si}, & E_{sl} \geq E_{max}, \end{cases} \tag{6}$$

in which $N$ is the number of subcarriers, $\beta$ is the embedding strength coefficient, and $B_{si}$ is the adjusting bandwidth in EDK. In our implementation, $B_{si}$ is set to be 20 Hz when the subcarrier bandwidth is 100 Hz. $E_{max}$ is a threshold to measure the maximum embedding signal energy spectrum density, set to $3 \times 10^{-7}$ J/Hz. If the energy of the original audio further increases, the strength of embedded signals remains unchanged since the signal is robust enough, and if we further increase the strength, the noise would be too large and quite perceiptible. As can be seen from Equations (5) and (6), the changes in the original audio in the case of EDK are usually larger than in the case of ASK. To facilitate channel estimation (Section 3.3.2), the receiver must know the signal energy of the pilots at the sender. Hence, we fix the energy of the pilots at the sender.

## 3.3 Signal Extraction

Embedded signal extraction on the receiver side after the audio is captured by the smartphone microphone includes three steps: preamble detection, channel estimation, and symbol extraction.

### 3.3.1 Preamble Detection

The preamble is used to locate the start of a packet. The control signal at 19.6 kHz of the preamble indicates the modulation method of the symbol subcarrier (Section 3.2.2).

We utilize envelope detection to detect the preamble chirp signals. In theory, the maximum envelope corresponds to the location of the preamble. But in practice, the envelopes in proximity to the location of the preamble are very similar in amplitude due to the ringing and rise time [20], resulting in synchronization errors within 5 data sample points in our preliminary experiments.

Though such synchronization errors normally cause unpredictable phase shift, in Dolphin, as each symbol corresponds to 4,410 data sample points, errors of up to 5 data sample points have little effect on the amplitude and energy distribution of the subcarrier signals. This is also the reason why we choose ASK and EDK instead of PSK.

### 3.3.2 Channel Estimation

After the preamble is detected and located, each symbol of a packet can be accurately segmented. As mentioned above, frequency selectivity estimation (FSE), time selectivity estimation (TSE), and Doppler frequency offset elimination

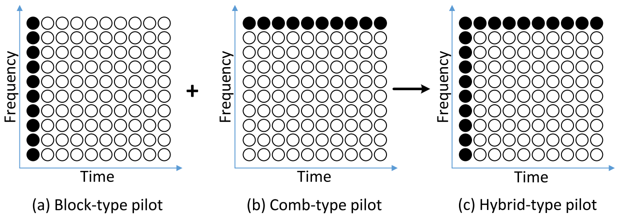


Fig. 6. Hybrid-type pilot scheme. The black dots are the pilots, and the white dots are the data bits.

(DFOE) are required before symbol extraction. In Dolphin, we adopt a channel estimation technique based on pilot arrangement [35].

*Pilot Type Selection*. The block-type pilot and the comb-type pilot [35] are two commonly-used pilots, as shown in Figs. 6a and 6b. Block-type pilot channel estimation inserts pilots at every subcarrier, and the estimation is used for a specific number of following symbols. It is effective in estimating the frequency-selective fading channel under the assumption that the channel transfer function is not changing very rapidly. Comb-type pilot channel estimation inserts pilots at a specific subcarrier of each symbol. It is effective in estimating the time-selective fading and Doppler frequency offset of each symbol and thus suitable for time-varying channels. Considering the high speaker-microphone frequency selectivity and large Doppler frequency offsets caused by mobility, we design a hybrid-type pilot, as shown in Fig. 6c. As mentioned in Section 3.2.1, we insert pilots on each subcarrier of the first symbol in a packet to estimate the frequency-selective fading and additional pilots at 20 kHz of each symbol to estimate the Doppler frequency offset and time-selective fading of each symbol.

*Channel Transform Function Estimation*. We first discuss how to estimate the frequency-selective fading function (FSE) based on the pilots on the first symbol of each packet. Least Square Estimation (LSE) or Minimum Mean-Square Estimation (MMSE) are often used to calculate the channel impulse response. MMSE performs better than LSE, but it is more complex and requires more computation resources. For real-time signal extraction, we adopt LSE in Dolphin. After removing the cyclic prefix, without taking into account ISI and ICI, the received signals in the first symbol can be expressed as:

$$y(n) = x(n) \otimes h(n) + w(n), n = 0, 1, \ldots, N-1, \tag{7}$$

where $w(n)$ denotes the ambient noise, $h(n)$ is the channel impulse response, and $N$ is the number of sample points in a symbol. We convert $y(n)$ from the time domain to the frequency domain via FFT as:

$$Y(k) = X(k) * H(k) + W(k), k = 0, 1, \ldots, N-1. \tag{8}$$

Let $Y_p(k)$ denote the pilot signals extracted from $Y(k)$, and $X_p(k)$ denote the known pilot signals added at the sender side. The estimated channel impulse response $H_e(k)$ can be computed as:

$$H_e(k) = \frac{Y_p(k)}{X_p(k)} = H_p(k) + \frac{W_p(k)}{X_p(k)}, \tag{9}$$

where $H_p(k)$ denotes the channel impulse response of pilot signals, $W_p(k)$ is the ambient noise of pilot signals, and $\frac{W_p(k)}{X_p(k)}$ is the estimation error. Since we only encode signals at frequencies higher than 8 kHz (Section 3.2.1), the ambient noise is negligible, resulting in very small estimation error. In fact, the frequency selectivity is mainly the result of the electro-mechanical components in the microphone/speaker rather than that of multi-path propagation [20]. Hence, the frequency-selective fading of the symbols following the first symbol is very close to $H_p(k)$.

Next, we discuss how to estimate the time-selective fading function (TSE) and Doppler frequency offset (DFOE) based on the pilots on 20 kHz subcarrier of each symbol. We also use LSE. Note that when the receiver is moving, the amplitude and phase of the channel response within one symbol will change due to Doppler frequency offset. To compensate for the estimation error, we need to take mobility into account. As the pilot frequency $f_s$ of transmitted signals is known (at 20 kHz), we can detect the pilots of received signals to obtain their frequencies $f_r$. Then, we can calculate the Doppler frequency shift determinant $\nu_0 \cos\theta$ as:

$$\nu_0 \cos\theta = \frac{(f_r - f_s)\nu}{f_s}. \tag{10}$$

We further calculate the frequency shift of all subcarriers in each symbol. After frequency offset elimination, all data signals are accurately located.

### 3.3.3 Symbol Extraction

After DFOE, every subcarrier's embedded data is accurately located, thus, we can extract the embedded signals of each symbol. We define a "data window" whose length equals the subcarrier bandwidth. To begin with, the data window encompasses the data whose center frequency is the frequency of the first subcarrier. We demodulate the signals according to the modulation method used for the subcarrier. Then, the data window moves forward at a step of one subcarrier bandwidth until the embedded bits of all subcarriers are extracted. Note that the power of the embedded signals is adaptive based on the average energy of a piece of audio corresponding to a symbol. Hence, we adjust the decision threshold for each symbol according to its average energy. In practice, the speaker-microphone link may suffer from serious distortion caused by the acoustic channel and hardware limitations. Though channel estimation is performed to minimize the impact of frequency-time selective fading and Doppler frequency offset, some signals still have a low SNR (signal to noise ratio) and it is hard to accurately determine their values. In the ASK and EDK modulation of Dolphin, the data values can only be '0' or '1'. We set the value of signals with a low SNR as 'x'.

## 3.4 Rateless Code-Based Orthogonal Error Correction

### 3.4.1 Data Error Analysis

We repeatedly test the error distribution of a packet under the same conditions (as described in our experimental settings), as shown in Fig. 7. In each test, many symbols have errors of fewer than 2 bits, probably caused by the noise rather than the speaker-microphone frequency selectivity, since the error

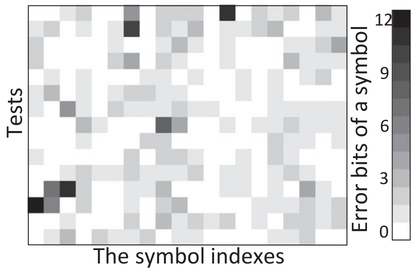


Fig. 7. The error distribution of a packet under repeated tests.

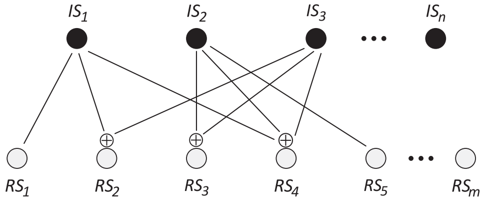


Fig. 8. An illustration of rateless symbol generation. The black dots indicate intermediate symbols, and the gray dots indicate rateless symbols.

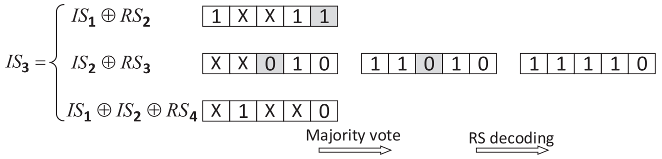


Fig. 9. An example of the ROEC decoding. The erased bits are marked as ‘x’ and the false positive bits are highlighted in gray.

distribution of a symbol in the frequency domain is random. A small intra-symbol error correction redundancy can correct most of these symbols. In some cases, the number of error bits in a symbol may exceed 10, probably due to high multi-path interference, and the distribution of these symbols in a packet is random. In those cases, we have to use inter-symbol erasure correction to guarantee reliability.

### 3.4.2 ROEC Encoding

Based on our observations of the characteristics of data errors, we adopt the two-tier rateless coding scheme [12], [13] to design and implement a rateless code-based orthogonal error correction (ROEC) scheme. In Dolphin, we intend to use as little intra-symbol error corrections as possible to avoid a high overhead due to limited encoding capacity. In addition, the channel conditions are diverse for different users. The same level of intra-symbol error correction redundancy may be enough for some users to decode most symbols correctly, but is not for others. To reduce overhead (e.g., checksum and error correction) and adapt to dynamic channel qualities, the ROEC scheme leverages all symbols and establishes a bit-level erasure channel by discarding low-confidence bits ( i.e., the bits set as ‘x’ in symbol extraction). Our ROEC scheme comprises of intra-symbol error correction and inter-symbol erasure correction in two orthogonal dimensions.

*Intra-Symbol Error Correction*. Inside a symbol, we focus on errors caused by the noise. In our implementation, we add Reed-Solomon (RS) codes [36] into the original data to generate intermediate symbols. Based on a finite field with 15 elements (1 element represents 4 bits), the RS$(n;k)$ code is able to correct up to $\lfloor (n-k)/2 \rfloor$ error elements and detect any combinations of up to $n-k$ error elements. In order to reduce overhead, we set the error correction coefficient $n-k$ as 2, which is relatively small. Therefore, the intra-symbol error correction only works well if the corruption in intermediate symbols is mild.

*Inter-Symbol Erasure Correction*. Inter-symbol bit-level erasure correction aims to erase most errors in intermediate symbols, such that the intra-symbol error correction can correct the remaining (small) errors. After each data symbol is encoded by RS codes, we encode all intermediate symbols into a stream of rateless symbols, each of which is generated by XORing some randomly chosen intermediate symbols, as shown in Fig. 8. The random choosing is controlled by a seed of random number generator. We insert the number of intermediate symbols and the seed into the first data symbol of the packet with sufficient error correction, to enable the receiver to reproduce the generation process of intermediate symbols and rateless symbols. In our implementation, the number of rateless symbols is $10\times$ larger than the number of intermediate symbols. Such a high redundancy is robust in case of varying channel qualities, even for some extremely decayed channels.

### 3.4.3 ROEC Decoding

After symbol extraction, the value embedded in each subcarrier has three cases, i.e., ‘0’, ‘1’ and ‘x’. The result of XORing ‘x’ with any value is ‘x’, which means that the bit is erased due to low confidence. Some extracted bits of ‘0’ or ‘1’ with high confidence (thus not to be erased) may be wrongly demodulated (false positive). Therefore, the ROEC decoding aims to recover the erased bits (‘x’) and correct the false positive bits.

Upon receiving a data packet, we first extract the number of intermediate symbols and the seed, both inserted in the first data symbol of the packet with a stronger error correction compared with other data symbols, to reproduce the generation process of intermediate symbols and rateless symbols. With more rateless symbols received, more instances of each intermediate symbol can be calculated. Take the intermediate symbol $IS_3$ in Fig. 8 as an example, $IS_3$ will have three instances after rateless symbols $RS_1 \sim RS_4$ are received. As shown in Fig. 9, for a certain bit in the intermediate symbol, different instances may yield different results. We record the number of ’1’ and ’0’ calculated in all instances, and use majority vote to set the bit to the value with the highest number of occurrences. After rateless decoding, most errors in intermediate symbols are erased, then the remaining errors can be corrected by RS decoding to recover the original bits.

Fig. 9 gives an example of the ROEC decoding. The intermediate symbol $IS_3$ contains five bits, which can be calculated by three instances with the decoded intermediate symbols and the received rateless symbols $RS_1 \sim RS_4$. We can see that all three instances contain erased bits (‘x’), and two instances contain false positive bits (highlighted in gray). We can not decode the original bits from any single instance, but after majority vote, most uncertain and error bits are recovered. Then, the remaining error in the third bit position can be corrected by RS decoding.

## 4 Secure Data Broadcasting Scheme

Dolphin achieves real-time unobtrusive speaker-microphone data communication without affecting the primary audio-hearing experience for users. In our previous design, everyone can decode the data embedded in the audio with his/her smartphone. But in some special cases, the sender may expect that only designated privileged users can recover the embedded information via Dolphin. For example, only VIP users can obtain the promotion code of goods or the permission of online premium movies. This motivates us to develop a secure data broadcasting (SDB) scheme for Dolphin based on broadcast encryption [11].

### 4.1 Framework

*Problem Definition.* Define the privilege user set as $G$, which has at most $n_p$ members. At each session, the sender specifies the valid privilege user set $S \subseteq G$, which the secret information are intended for. The sender broadcasts the secret information, and every user $u_i \in S$ can decode the secret information while other users cannot acquire any secret information. The sender can dynamically change the valid privileged user set $S$ by adding or evicting members. More specifically, for Dolphin, the sender broadcasts the data-embedded audio via speakers, and all users can use their smartphone equipped with Dolphin to receive and extract the acoustic signals, but only the users $u_i \in S$ can decode the secret information.

*Bilinear Maps.* We briefly review the basic definition of bilinear maps, based on which our SDB scheme is developed. $\mathbb{G}$ and $\mathbb{G}_1$ are two multiplicative cyclic groups of prime order $p$, and $e : \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_1$ is a bilinear map with the following properties:

1) *Bilinear*. $\forall g, h \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, $e(g^a, h^b) = e(g, h)^{ab}$.
2) *Non-degenerate*. $\exists g, h \in \mathbb{G}$, $e(g, h) \neq 1$.
3) *Efficient*. $\forall g, h \in \mathbb{G}$, $e(g, h)$ can be computed efficiently.

*Addition and Revocation.* The SDB scheme needs to enable the sender to dynamically change the set of privileged users by adding or revoking members. The privileged user set $G$ can contain no more than $n_p$ users. Assume that $m$ users are currently in $G$. If $m < n_p$, a new user can be directly added into $G$ and assigned the corresponding private key; otherwise, the sender has to abolish the current privilege user set and rebuild a new one in order to include new members (e.g., by eliminating ones who are no longer valid privileged users). To revoke current privileged users, we can simply stipulate the valid member of $S$ at each session and embed the related information in the broadcast header.

Our scheme consists of three steps: 1) Key initiation. We generate the public key $PK$ and $n_p$ private keys $SK_1, \ldots, SK_{n_p}$, then assign private key $SK_i$ to user $g_i \in G$. 2) Encryption and broadcast. Take $S$ and $PK$ as the input, we compute a pair $(Hdr, K)$ where $Hdr$ is the header and $K$ is the session key. Given a message $M$, we encrypt it using $K$ and obtain the ciphertext $C_M$. $S$, $Hdr$ and $C_M$ are embedded into the audio and broadcasted to all receivers. 3) Extraction and decryption. Any user can use their smartphone to receive the acoustic signals and extract the embedded ciphertext $C_M$ , but only the valid privileged users $u_i \in S$ can use their private key $SK_i$ to deduce $K$. Then the session key $K$ is used to decrypt $C_M$ and obtain the message $M$. Note that all privileged users $g_i \in G$ possess a private key, but only the valid ones $u_i \in S$ can use their private key to decrypt the session key $K$. The sender can dynamically adjust the valid privileged user set $S$ at each session through the computation of the session key $K$.

### 4.2 Implementation

*Key Initiation.* When the user first registers as privileged user online using the Dolphin App, he/her will be granted the public key $PK$ and assigned a corresponding private key $SK_i$, which are stored on the user's smartphone. Let $\mathbb{G}$ denote a bilinear map group of prime order $p$, $|\mathbb{G}| = n_p$. We first choose a random generator $g \in \mathbb{G}$ and a random number $\alpha \in \mathbb{Z}_p$ (integer modulo $p$). Then, we compute $g_i = g^{\alpha^i} \in \mathbb{G}$ for $i = 1, 2, \ldots, n_p, n_p + 2, \ldots, 2n_p$. A random number $\gamma \in \mathbb{Z}_p$ is picked, and we calculate $\upsilon = g^\gamma \in \mathbb{G}$. The public key is computed as:

$$PK = (g, g_1, \ldots, g_{n_p}, g_{n_p+2}, \ldots, g_{2n_p}, \upsilon) \in \mathbb{G}^{2n_p+1}. \tag{11}$$

The private key for user $g_i \in G$ is computed as:

$$SK_i = g_i^\gamma \in \mathbb{G} = \upsilon^{(\alpha^i)}. \tag{12}$$

Note that the public key $PK$ is public and can be accessed by anyone, while the private key $SK_i$ is only assigned to user $g_i \in G$.

*Encryption and Broadcast.* Given a valid privileged user set $S$, the sender picks a random number $t \in \mathbb{Z}_p$. The session key is computed as:

$$K = e(g_{n_p+1}, g)^t \in \mathbb{G}, \tag{13}$$

and the header is set as:

$$Hdr = \left(g^t, \left(\upsilon \cdot \prod_{j \in S} g_{n_p+1-j}\right)^t\right) \in \mathbb{G}^2. \tag{14}$$

We generate an indicator (denoted as $SI$) to indicate whether or not a user $u_i \in G$ is a member of $S$. In our implementation, $SI$ is an $n_p$-bit number, and the initial value of all bits are set as "0". If $u_i \in S$ is true, the $i$th bit of $SI$ will be set as "1". In Dolphin, we leverage AES [37] to encrypt $M$ block by block (each block data is 16-byte) using $K$. Finally, $SI$, $Hdr$, and $C_M$ are embedded into the audio and broadcast to users via the speaker.

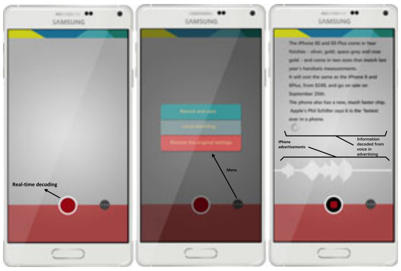
*Extraction and Decryption.* The users use their smartphone, installed with Dolphin, to receive the acoustic signals and extract the embedded broadcast information (i.e., $\langle SI, Hdr, C_M \rangle$). After extracting $SI$ and $Hdr$, each valid privileged user $u_i \in S$ can deduce the session key $K$ as follows:

$$K = \frac{e(g_i, (\upsilon \cdot \prod_{j \in S} g_{n_P+1-j})^t)}{e(SK_i \cdot \prod_{\substack{j \in S \\ j \neq i}} g_{n_p+1-j+i}, g^t)}. \tag{15}$$

while other users (including invalid privileged users $u_i \in G \setminus S$) cannot calculate $K$. Then $K$ is used to decrypt $C_M$ to recover the original message $M$.

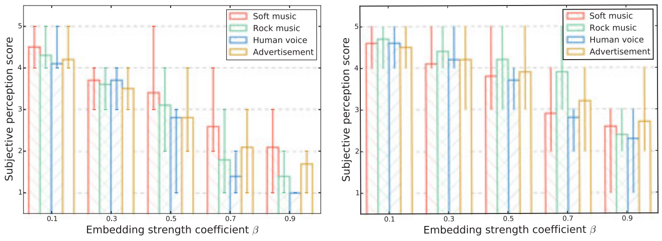
## 5 Implementation and Evaluation

We implement a prototype of Dolphin using commodity hardware. The sender is implemented on a PC equipped with a loudspeaker and the receiver is implemented as an



The iPhone 6S and 6S Plus come in four finishes – silver, gold, space grey and rose gold – and come in two sizes that match last year's handsets measurements.
It will cost the same as the iPhone 6 and 6Plus, from $199, and go on sale on September 25th.
The phone also has a new, much faster chip. Apple's Phil Schiller says it is the 'fastest ever in a phone.

Fig. 10. Implementation of Dolphin on the smartphone.



(a) Static embedding (b) Adaptive embedding

Fig. 11. Adaptive embedding improvement on subjective perception.

Android App on different smartphone platforms. The App interface on GALAXY Note4 is shown in Fig. 10.

We use a DELL Inspiron 3647 with 2.9 GHz CPU and 8 GB memory controlling a HiVi M200MKIII loudspeaker as the sender. The default speaker volume is 80 dB (measured by a decibelmeter APP at 1 m distance), and the default distance is 1 m. At the receiver side, we use Galaxy Note4 in most of our experiments. We show the performance comparison across different smartphones in Section 5.2.5. The sampling rate on the receiver is 44.1 kHz.

## 5.1 Subjective Perception Assessment

We conduct a user study to examine whether Dolphin has any noticeable auditory impact on the original audio content and identify a set of design parameters that are ideal for a better auditory experience. Our user study is conducted with 40 participants (22 males and 18 females) whose ages range from 18 to 46. The quality of data-embedded audio is evaluated with a score from 5 to 1, namely "5: completely unobtrusive", "4: almost unnoticeable", "3: not annoying", "2: slightly annoying", "1: annoying". We test four different types of audio sources, including soft music, rock music, human voice, and advertisements. Each type of sources is evaluated using 10 different samples. The experiments are conducted in an office with the speaker volume set as 80 dB and a speaker-smartphone distance of 1 m.

### 5.1.1 *Embedding Strength Coefficient $\beta$*

The embedding strength coefficient $\beta$ is the most critical parameter that determines the embedded signal energy and affects the subjective perception. A large value of $\beta$ makes communication more robust but makes it easier for users to perceive the change in the received audio. To isolate the impact of $\beta$ and show the effectiveness of our adaptive embedding approach, we use ASK as the modulation method for all symbols and let the energy of each symbol signal stay unchanged with the energy of its carrier audio (called static embedding). In static embedding, we measure $E_{sl}$ of 10 different samples for each type of audio source, and calculate the average value $\overline{E}_{sl}$.

As shown in Fig. 11a, the subjective perception score decreases as $\beta$ increases. However, different types of audio have different sensitivity to $\beta$. The scores of soft music and advertisements are generally higher than those of voice and rock music. In the case of human voice with no background music, the noise is easy to perceive when the speech pauses. As for rock music, some pieces contain a high energy in high frequencies. If we embed data symbols into such pieces and change the energy distribution, such changes are also easy to perceive. Overall, we observe that for $\beta \geq 0.3$, almost all subjective perception scores drop below 4 for all types of audios. On the other hand, a low $\beta$ reduces the robustness of our system.

### 5.1.2 *Adaptive Embedding Improvement*

In adaptive embedding, we calculate the energy of each piece of carrier audio corresponding to a symbol in real time, based on which the energy of a symbol signal is changed adaptively according to Equations (5) and (6). Fig. 11b shows the performance of our adaptive embedding method (Section 3.2.3) that balances the tradeoff between audio quality and signal robustness. Compared with Fig. 11a, the scores of all types of audios are improved. In particular, we observe that the use of EDK significantly enhances the scores of rock music, as we explained in Section 3.2.1. The scores of voice are also improved since we do not embed data bits when the speech pauses. Nevertheless, the improvement of soft music is not obvious, since the energy of soft music is relatively steady. When $\beta = 0.5$, all types of audio sources achieve a score close to 4. Hence, $\beta$ should be no more than than 0.5 to ensure satisfactory auditory experience in practice.

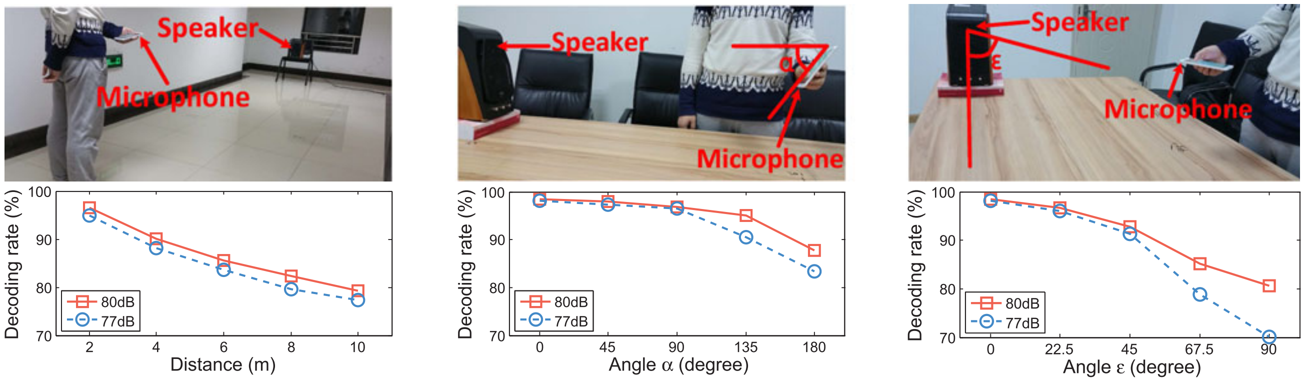
## 5.2 Practical Considerations

We now evaluate the impact of practical factors on Dolphin's performance. Since the ROEC scheme can correct most errors caused by these factors, we focus on the decoding rate before ROEC to better understand the impact of these factors. The default values of system parameters are set as follows: embedding strength coefficient $\beta = 0.5$, the symbol duration $T = 100$ ms, and number of subcarriers $N = 60$.

### 5.2.1 *Distance and Angle*

The impact of distance on decoding rate is significant, because the acoustic power decays with the square of distance. As shown in Fig. 12a, the decoding rate decreases as the distance increases but remains above 80 percent for distances up to 6 m with a volume of 77 dB and 8 m with a volume of 80 dB. Obviously, Dolphin can support even longer distances by adjusting the speaker volume.

We conduct two sets of experiments regarding the smartphone rotation and horizontal angles. In both experiments, the speaker-smartphone distance remains as 1 m. As shown in Fig. 12b, Dolphin's overall performance is relatively stable when the smartphone rotates vertically from 0° to 90°. When $\alpha$ = 180°, i.e., the speaker and the smartphone face towards opposite directions, the decoding rate is still above 80 percent. This demonstrates the practicality of Dolphin, which does not require the users to keep the microphone strictly towards the direction of the sound source. Fig. 12c shows that the

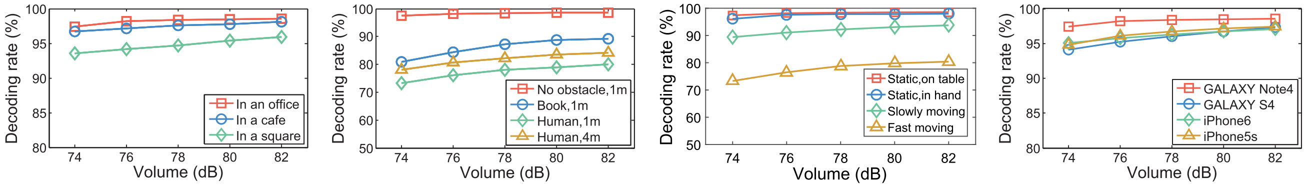


(a) Distance.

(b) Rotation angle $\alpha$.

(c) Horizontal angle $\varepsilon$.

Fig. 12. The impact of distance and angle on decoding rate.



(a) Ambient noise.

(b) Obstacles.

(c) Handhold and motion.

(d) Different Smartphones.

Fig. 13. The impact of various practical settings on decoding rate.

decoding rate remains relatively stable when the horizontal angle $\varepsilon$ varies from $0^\circ$ to $45^\circ$, but decreases sharply for larger angles. This is because the HiVi M200MKIII speaker transmits directionally. If the smartphone lies within the speaker's transmission conical beam, the microphone can capture the audio directly; otherwise, the audio can only arrive at the receiver by reflection. Even so, the decoding rate is still above 80 percent with a speaker volume of 80 dB when $\varepsilon = 90^\circ$. This confirms that Dolphin can ensure good performance for most places around the speaker.

### 5.2.2 Ambient Noise

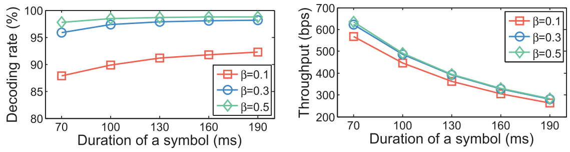
As shown in Fig. 13a, we conducted experiments regarding the impact of ambient noise at three different locations: an office, a cafe, and a square. We observe that Dolphin is resilient to ambient noise, maintaining a decoding rate above 90 percent at all three locations. This is because we select the appropriate frequency bandwidth for the embedded signal to reduce the influence of ambient noise. In the office, the ambient noise is quite small. In the cafe, the ambient noise is mainly due to conversations among customers. However, the frequency range of human voice is relatively low, thus does not cause severe interference to the sound signals above 8 kHz. In a square, there are multiple sound sources, some of which generate higher frequency sounds; hence, Dolphin performs slightly worse compared to the other two locations.

### 5.2.3 Obstacles

In this section, we discuss the impact of obstacles between the sound source and the receiver microphone on the decoding rate. The obstacles include a $28 \times 21 \times 5$ cm book or a human between the HiVi M200MKIII (sender) and the Galaxy Note4 phone (receiver). The LOS between the speaker and the microphone is completely blocked by the obstacles. From Fig. 13b, we can observe that the presence of an obstacle obviously decreases the decoding rate while the sound signals can still reach the receiver via diffraction. On the one hand, the size of obstacles will affect the performance. When the volume of speaker is above 80 dB, the decoding rate with the book blocking at the distance of 1 m is about 90 percent but the decoding rate with a human blocking at the distance of 1 m is about 80 percent. On the other hand, the distance also affects the performance. The decoding rate with the human blocking at the distance of 4 m is even higher than that at the distance of 1 m. As can be seen from Fig. 12a, the decoding rate decreases as the communicating distance increases. The HiVi M200MKIII speaker transmits directionally. When the human stands very close to the speaker, the sound conical beam will be completely blocked. When the human gradually moves away from the speaker, the unblocked signals can still reach the receiver via diffraction. Dolphin will perform better with obstacles by using a speaker with a wider transmission angle. This is a great advantage of Dolphin compared to unobtrusive screen-to-camera communication systems which are very sensitive to any obstacle.

### 5.2.4 Device Motion

We evaluate three types of motion: (i) a static user holds the Galaxy Note4 in the air facing the HiVi M200MKIIII; in this case, the motion is due to the slight hand shaking; (ii) the user slowly moves the smartphone towards and away from the speaker; and (iii) the user quickly moves the smartphone towards and away from the speaker. As shown in Fig. 13c, when a static user holding the phone, the performance is very close to the case where the phone is placed on a table, i.e., the impact of slight hand shaking is negligible. In comparison, the impact of the other two motion modes is more prominent due to Doppler frequency shift. In *Channel Estimation*, we estimate the frequency offset and mitigate its



(a) Decoding rate (b) Throughput

Fig. 14. The impact of $T$ with different $\beta$ on the decoding rate and throughput.

effect. However, the frequency offset mitigation does not perform very well when the frequency shift is large in a short time. The frequency shift is proportional to the moving speed, thus the decoding rate is lower under fast movement than that under slow movement. However, the decoding rate still remains higher than 90 percent with slow movement when the the volume is above 76 dB.

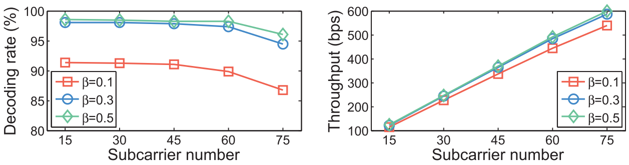
### 5.2.5 Different Smartphone Models

We examine the impact of different smartphone models and operating systems on Dolphin's performance. We use four smartphone models: GALAXY Note4, GALAXY S4, iPhone 6, and iPhone 5s. Our current implementation of the Dolphin receiver is based on Android operating system. To test Dolphin on iPhone 6 and iPhone 5s, we use the smartphones to capture the audio signals and decode them on the PC. As shown in Fig. 13d, the performance of GALAXY Note4 is the best and that of GALAXY S4 is the worst, mainly due to different frequency selectivity of microphones. Nonetheless, all four models maintain a decoding rate higher than 95 percent when the volume is above 76 dB, thanks to the use of pilots in the first symbol of each packet that allow the receiver to estimate the frequency selective fading function and eliminate the impact of frequency selectivity.

*Discussion.* Dolphin mainly focuses on signal broadcasting application scenarios (including secure data broadcasting) rather than device-to-device communication, thus we implement data encoding on the PC (connected to a high-power loudspeaker) as the transmitter. However, to test the performance of Dolphin using a speaker of inferior quality, we use GALAXY Note4 or GALAXY S4 as the sender and GALAXY Note4 as the receiver for comparison. Compared with the HiVi M200MKIII loudspeaker, the smartphone speakers have a lower volume and a higher frequency selectivity. In our test, the volume of smartphone speakers is set to 100 percent, which is around 65 dB. We focus on the performance under several practical considerations (e.g., distance, angle and obstacles). We found that Dolphin supports up to 5 m signal capture distance and 360° listening angle. In addition, the decoding rate with the human blocking at the distance of 1 m is above 85 percent. The results show that the volume of the smartphone speaker limits the signal capture distance, but a better performance in listening angle and obstacles is achieved due to a wider transmission angle of smartphone speakers.

## 5.3 Communication Performance

In this section, we evaluate the communication performance of Dolphin in terms of: 1) decoding rate and throughput, 2) encoding and decoding time, 3) goodput. Note that goodput is the application-level throughput, which equals overall throughput excluding the coding overhead.



(a) Decoding rate (b) Throughput

Fig. 15. The impact of $N$ with different $\beta$ on the decoding rate and throughput.

### 5.3.1 Decoding Rate and Throughput

The decoding rate and throughput are mainly affected by two factors: the symbol duration $T$ and the number of subcarriers $N$. The default values of $N$ and $T$ are 60 and 100 ms, respectively. The test audio sources for different $\beta$ include soft music, rock music, human voice, and advertisements. The experiments are conducted in an office with the speaker volume set as 80 dB and a speaker-smartphone distance of 1 m.

As shown in Fig. 14a, the decoding rate increases when $T$ increases, as a longer duration allows for more repetitions of the same signal. When $T$ is larger than 100 ms, the average decoding rate over all audios with $\beta = 0.5$ is above 98 percent. However, 100 percent reliability is very hard to achieve in practice. In addition, we observe that the decoding rate for a given $T$ is different for different $\beta$. When $\beta = 0.1$, the subjective perception score is ideal, but the decoding rate is obviously lower than that with $\beta = 0.3$. Therefore, there exists a tradeoff between audio quality and signal robustness. Fig. 14b shows the effect of symbol duration on the throughput performance. As expected, the throughput decreases when $T$ increases. Similar to the decoding rate, a given $T$ yields different throughputs for different $\beta$.

As shown in Fig. 15a, the decoding rate drops significantly when $N$ is larger than 60. To ensure the same level of subjective perception, the total energy embedded in a symbol is constant once the piece of audio carrier is determined. If the number of subcarriers increases, the energy per subcarrier decreases. Furthermore, we observe that the performance with $\beta = 0.1$ is still poor. Since a larger $\beta$ can improve the signal robustness with acceptable unobtrusiveness, we set $\beta = 0.5$ in the following experiments. Fig. 15b shows that the throughput increases when $N$ increases, because more subcarriers can carry more information.

When $T$=100 ms and $N$=60, the average throughput of different types of audios is about 500bps. We believe this throughput is sufficient for most of our targeted application scenarios because the embedded information is usually side information (e.g., verbal descriptions of video/audio contents). Take a 1-minute advertisement as an example. The ad can load about $500 \times 60/8 = 3750$ letters. Assume a word consists of 6 letters on average. Then, there are about 625 words which can be instant notifications, offers, and promotions, etc.

### 5.3.2 Encoding and Decoding Time

To evaluate Dolphin's ability to support real-time communications, we measure per-symbol encoding and decoding time. We use the default setting $T = 100$ ms and $N = 60$. At the sender, we measure the encoding time of each symbol including *ROEC Encoding*, *Energy Analysis*, and *Adaptive Embedding*. At the receiver, we first perform *Preamble*

TABLE 1
The Average Encoding Time of a Symbol

| | |
|---|---|
| ROEC Encoding (ms) | 0.82 |
| Energy Analysis (ms) | 0.35 |
| Adaptive Embedding (ms) | 0.18 |
| **Total (ms)** | 1.35 |

TABLE 2
The Average Time of Pre-Processing a Packet and Decoding a Symbol

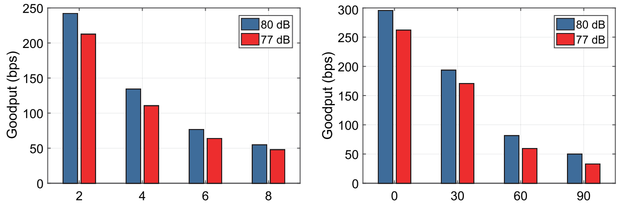
| | | Note4 | S4 |
|---|---|---|---|
| **Prep.(ms)** | Preamble Detection | 369.2 | 542.9 |
| | FSE | 23.4 | 34.5 |
| **Prep. Sub-total (ms)** | | 392.6 | 577.4 |
| **Dec.(ms)** | TSE and DFOE | 22.3 | 32.2 |
| | Symbol Extraction | 0.8 | 1.6 |
| | ROEC Decoding | 2.2 | 4.1 |
| **Dec. Sub-total (ms)** | | 25.3 | 37.9 |

*Detection* and *Frequency Selectivity Estimation (FSE)* for each packet, then we decode each symbol. Therefore, the decoding time of each symbol only consists of *Time Selectivity Estimation (TSE)*, *Doppler Frequency Offset Elimination (DFOE)*, *Symbol Extraction* and *ROEC Decoding*.

Table 1 shows that the average real-time encoding time of a symbol is much shorter than the symbol duration (100 ms), hence the sender is able to support real-time operation. Table 2 shows the average time of pre-processing a packet and decoding a symbol in real time on two smartphones. The results show that *Preamble Detection* is the most time-consuming operation. This is because the envelopes of different pieces of audios need to be calculated to find the maximum and it involves iterative operations. In spite of this, *Preamble Detection* belongs to packet pre-processing, which is only necessary for each packet rather than each symbol, such that the total decoding time of each symbol is still smaller than the symbol duration. Hence, the receiver can support real-time decoding, but with a short delay due to the time of packet pre-processing.

#### 5.3.3 *Goodput*

In this experiment, we use the ROEC scheme to adaptively correct different levels of bit error rates under different communicating conditions. From Section 5.2, we can see that the impacts of distance and horizontal angle $\varepsilon$ on the decoding rate are the most significant ones. Therefore, we first vary the distance from 2 m to 8 m in a long corridor with a speaker volume of 77 dB and 80 dB while keeping the horizontal angle $\varepsilon = 0^\circ$, then vary the horizontal angle $\varepsilon$ from $0^\circ$ to $90^\circ$ with a speaker volume of 77 dB and 80 dB while keeping the distance as 1 m. We evaluate the adaptability and transmission efficiency in terms of goodput, which is the ratio of the correctly decoded data bits (excluding the bits used for error-correction) to the total transmission time.



(a) Distance (m) (b) Horizontal angle (°)

Fig. 16. The goodput under different distances and angles.

TABLE 3
The Extra Overhead of SDB Scheme

| $n_p$ | 128 | | 256 | |
|---|---|---|---|---|
| $l_e$ (bytes) | 10 | 20 | 10 | 20 |
| Communication Overhead (bytes) | 36 | 56 | 52 | 72 |
| Storage at each Receiver (KB) | 2.6 | 5.1 | 5.1 | 10.2 |
| Generation Time of $K$ and $Hdr$ (ms) | 8.2 | 9.4 | 10.5 | 11.6 |
| Encryption Time of Each Block (ms) | 0.13 | 0.14 | 0.12 | 0.13 |
| Decryption Time of $K$ (ms) | 554.2 | 561.6 | 578.7 | 596.1 |
| Decryption Time of Each Block (ms) | 0.83 | 0.82 | 0.82 | 0.81 |

From Figs. 16a and 16b, we can observe that Dolphin supports up to 8 m signal capture distance and $90^\circ$ listening angle, and it achieves an average goodput of 240 bps at 2 m while keeping the smartphone towards the speaker. In addition, Dolphin can automatically adapt the data rate as the distance and horizontal angle changes. This is because a longer communicating distance and a larger horizontal angle lead to higher bit errors, thus more rateless symbols need to be received to recover the original bits. Compared with the OEC scheme in our preliminary version of this paper [1], the improved ROEC scheme has two advantages. First, the ROEC scheme enables receivers with different channel qualities to adapt the goodput without changing the encoding configurations at the sender, while the OEC scheme requires the sender to adjust the intra-symbol error correction parameter $n - k$ and the inter-symbol erasure correction parameter $m$ in order to optimize the goodput under different channel conditions. Second, the transmission efficiency in the ROEC scheme is higher than that in the OEC scheme. For example, the ROEC scheme achieves an average goodput of about 240 bps at 2 m, while the maximum goodput in the OEC scheme is about 200 bps at 2 m.

### 5.4 Extra Overhead of SDB Scheme

Compared with the original design of Dolphin, the SDB scheme has extra overheads in three aspects: 1) communication overhead of broadcast header, 2) storage overhead at each receiver, 3) encryption and decryption time.

Table 3 shows the extra overheads of our SDB scheme with different numbers of privileged users ($n_p$) and different lengths of bilinear map group elements (denoted as $l_e$). The length of each group element meeting standard security parameter is 20 bytes. The embedded broadcast information consists of broadcast header $\langle SI, Hdr \rangle$ and ciphertext $C_M$. The length of $C_M$ is the same as that of $M$, while the length of $SI$ is proportional to $n_p$, and the length of $Hdr$ is proportional to $l_e$. For example, when $n_p = 128, l_e = 10$ bytes, the communication overhead of the broadcast header is $128/8 + 10 \times 2 = 36$ bytes.

In addition, we evaluate the storage overhead at each receiver's smartphone. The storage overhead at the sender's PC is negligible, and each privileged user needs to store the public key $PK$ and their private key $SK_i$ on the smartphone.

The size of $SK_i$ of each receiver is the same as $l_e$, while the size of $PK$ increases as $n_p$ and $l_e$ increase. The size of $PK$ is very small compared with the storage capacity of a smartphone.

Moreover, we measure the generation time of $K$ and $Hdr$, the decryption time of $K$, the encryption time of $M$ and the decryption time of $C_M$. The sender first generates the session key $K$ and the header $Hdr$, then adopts AES to encrypt the secret message $M$ using $K$. After the receivers in $S$ extract the embedded broadcast information, they first calculate the session key $K$ according the broadcast header, then perform decryption to recover the secret message $M$. The decryption time of $K$ is relatively long, however we only need to calculate $K$ once every session. Since AES encrypts data block by block and each block data is 16 bytes, we measure the encryption and decryption time of each block, which corresponds to approximately 2 symbols in Dolphin. From Table 3, we can observe that the encryption and decryption time have little influence on the performance of our real-time encoding and decoding thanks to the highly efficient AES algorithm.

## 6 Conclusions

We presented and implemented Dolphin, a new paradigm of real-time unobtrusive dual-mode speaker-microphone communication atop any audio content generated on the fly. Dolphin overcomes the intrinsic limitations of screen-camera links and automatically adapts the data rate to various channel conditions. We also develop a secure data broadcasting scheme on Dolphin to ensure that transmitted data can only be decoded by designated privileged users in some special scenarios. We implemented Dolphin on off-the-shelf smartphones and evaluated it extensively under various environments and practical considerations. Dolphin has a great potential to be adopted as a complementary or joint dual-channel communication strategy with existing unobtrusive screen-camera systems to enhance the performance of communication under various practical settings.

## Acknowledgments

Qian Wang's research is supported in part by the National Natural Science Foundation of China (Grant Nos. U1636219, 61373167), and the Outstanding Youth Foundation of Hubei Province (Grant No. 2017CFA047). Kui Ren's research is supported in part by the US National Science Foundation under grant CNS-1421903. Lu Su's research is supported in part by the US National Science Foundation under grant CNS-1652503. Yanjiao Chen's research is supported in part by the National Natural Science Foundation of China under Grant No. 61702380, the Hubei Provincial Natural Science Foundation of China under Grant No. 2017CFB134, and the Hubei Provincial Technological Innovation Special Funding Major Projects under Grant No. 2017AAA125. A preliminary version of this work has appeared in the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom'16) [1] along with a demo paper [2].

## References

[1] Q. Wang, K. Ren, M. Zhou, T. Lei, D. Koutsonikolas, and L. Su, "Messages behind the sound: Real-time hidden acoustic signal capture with smartphones," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 29–41.

[2] Q. Wang, K. Ren, M. Zhou, T. Lei, D. Koutsonikolas, and L. Su, "Demo: Real-time hidden acoustic signal capture with smartphones," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netwo.*, 2016, pp. 493–494.

[3] [Online]. Available: https://www.sandvine.com/trends/global-internet-phenomena

[4] G. Woo, A. Lippman, and R. Raskar, "VRcodes: Unobtrusive and active visual codes for interaction by exploiting rolling shutter," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2012, pp. 59–64.

[5] W. Huang and W. H. Mow, "PiCode: 2D barcode with embedded picture and ViCode: 3D barcode with embedded video," in *Proc. ACM 19th Annu. Int. Conf. Mobile Comput. Netw.*, 2013, pp. 139–142.

[6] R. Carvalho, C.-H. Chu, and L.-J. Chen, "IVC: Imperceptible video communication," in *Proc. ACM HotMobile*, demo, 2014.

[7] A. Wang, Z. Li, C. Peng, G. Shen, G. Fang, and B. Zeng, "Inframe++: Achieve simultaneous screen-human viewing and hidden screen-camera communication," in *Proc. ACM 13th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2015, pp. 181–195.

[8] T. Li, C. An, X. Xiao, A. T. Campbell, and X. Zhou, "Real-time screen-camera communication behind any scene," in *Proc. ACM 13th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2015, pp. 197–211.

[9] M. Izz, Z. Li, H. Liu, Y. Chen, and F. Li, "Uber-in-light: Unobtrusive visible light communication leveraging complementary color channel," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, 2016, pp. 1–9.

[10] V. Nguyen, Y. Tang, A. Ashok, M. Gruteser, K. Dana, W. Hu, E. Wengrowski, and N. Mandayam, "High-rate flicker-free screen-camera communication with spatially adaptive embedding," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, 2016, pp. 1–9.

[11] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Proc. 25th Annu. Int. Conf. Advances Cryptology*, 2005, vol. 3621, pp. 258–275.

[12] W. Du, Z. Li, J. C. Liando, M. Li, W. Du, Z. Li, J. C. Liando, and M. Li, "From rateless to distanceless: Enabling sparse sensor network deployment in large areas," *IEEE/ACM Trans. Netw.*, vol. 24, no. 4, pp. 2498–2511, Aug. 2016.

[13] Z. Li, W. Du, Y. Zheng, M. Li, and D. Wu, "From rateless to hopless," *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 69–82, Feb. 2017.

[14] T. Hao, R. Zhou, and G. Xing, "COBRA: Color barcode streaming for smartphone systems," in *Proc. ACM 10th Int. Conf. Mobile Syst. Appl. Services*, 2012, pp. 85–98.

[15] W. Hu, H. Gu, and Q. Pu, "LightSync: Unsynchronized visual communication over screen-camera links," in *Proc. ACM 19th Annu. Int. Conf. Mobile Comput. Netw.*, 2013, pp. 15–26.

[16] A. Wang, S. Ma, C. Hu, J. Huai, C. Peng, and G. Shen, "Enhancing reliability to boost the throughput over screen-camera links," in *Proc. ACM 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 41–52.

[17] Q. Wang, M. Zhou, K. Ren, T. Lei, J. Li, and Z. Wang, "Rainbar: Robust application-driven visual communication using color barcodes," in *Proc. IEEE 35th Int. Conf. Distrib. Comput. Syst.*, 2015, pp. 537–546.

[18] B. Zhang, K. Ren, G. Xing, X. Fu, and C. Wang, "SBVLC: Secure barcode-based visible light communication for smartphones," *IEEE Trans. Mobile Comput.*, vol. 15, no. 2, pp. 432–446, Feb. 2016.

[19] W. Du, J. C. Liando, and M. Li, "Soft hint enabled adaptive visible light communication over screen-camera links," *IEEE Trans. Mobile Comput.*, vol. 16, no. 2, pp. 527–537, Feb. 2017.

[20] R. Nandakumar, K. K. Chintalapudi, V. Padmanabhan, and R. Venkatesan, "Dhwani: Secure peer-to-peer acoustic NFC," in *Proc. ACM SIGCOMM Conf. SIGCOMM*, 2013, pp. 63–74.

[21] B. Zhang, Q. Zhan, S. Chen, M. Li, K. Ren, C. Wang, and D. Ma, "Priwhisper: Enabling keyless secure acoustic communication for smartphones," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 33–45, Feb. 2014.

[22] H. Lee, T. H. Kim, J. W. Choi, and S. Choi, "Chirp signal-based aerial acoustic communication for smart devices," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 2407–2415.

[23] A. S. Nittala, X.-D. Yang, S. Bateman, E. Sharlin, and S. Greenberg, "Phoneear: Interactions for mobile devices that hear high-frequency sound-encoded data," in *Proc. 7th ACM SIGCHI Symp. Eng. Interactive Comput. Syst.*, 2015, pp. 174–179.

[24] H. S. Yun, K. Cho, and N. S. Kim, "Acoustic data transmission based on modulated complex lapped transform," *IEEE Signal Process. Lett.*, vol. 17, no. 1, pp. 67–70, Jan. 2010.

[25] G. E. Santagati, T. Melodia, L. Galluccio, and S. Palazzo, "Medium access control and rate adaptation for ultrasonic intrabody sensor networks," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1121–1134, Aug. 2015.

[26] S. Ka, T. H. Kim, J. Y. Ha, S. H. Lim, S. C. Shin, J. W. Choi, C. Kwak, and S. Choi, "Near-ultrasound communication for tv's 2nd screen services," in *Proc. ACM 22nd Annu. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 42–54.

[27] N. Roy, H. Hassanieh, and R. Roy Choudhury , "Backdoor: Making microphones hear inaudible sounds," in *Proc. ACM 15th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2017, pp. 2–14.

[28] P. Getreuer, C. Gnegy, R. F. Lyon, and R. A. Saurous, "Ultrasonic communication using consumer hardware," *IEEE Trans. Multimedia*, vol. 20, no. 6, pp. 1277–1290, Jun. 2018.

[29] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1673–1687, Dec. 1997.

[30] P. Bassia, I. Pitas, and N. Nikolaidis, "Robust audio watermarking in the time domain," *IEEE Trans. Multimedia*, vol. 3, no. 2, pp. 232–241, Jun. 2001.

[31] M. D. Swanson, B. Zhu, A. H. Tewfik, and L. Boney, "Robust audio watermarking using perceptual masking," *Elsevier Signal Process.*, vol. 66, no. 3, pp. 337–355, 1998.

[32] M. Arnold, "Audio watermarking: Features, applications, and algorithms," in *Proc. Int. Conf. Multimedia Expo. Latest Advances Fast Changing World Multimedia*, 2000, pp. 1013–1016.

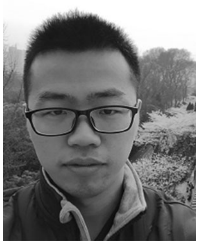
[33] X.-Y. Wang and H. Zhao, "A novel synchronization invariant audio watermarking scheme based on DWT and DCT," *IEEE Trans. Signal Process.*, vol. 54, no. 12, pp. 4835–4840, Dec. 2006.

[34] E. Zwicker and U. T. Zwicker, "Audio engineering and psychoacoustics: Matching signals to the final receiver, the human auditory system," *J. Audio Eng. Soc.*, vol. 39, no. 3, pp. 115–126, 1991.

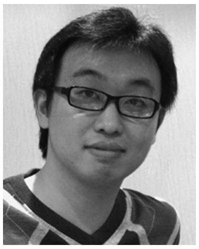
[35] S. Coleri, M. Ergen, A. Puri, and A. Bahai, "Channel estimation techniques based on pilot arrangement in OFDM systems," *IEEE Trans. Broadcast.*, vol. 48, no. 3, pp. 223–229, Sep. 2002.

[36] S. B. Wicker, *Reed-Solomon Codes and Their Applications*. Piscataway, NJ, USA: IEEE Press, 1994.

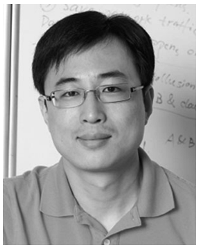
[37] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Boca Raton, FL, USa: CRC Press, 2014.



**Man Zhou** is working towards the MS degree in the School of Cyber Science and Engineering, Wuhan University, China. His research interests include mobile system and security. He was the recipient of the first prize in the "National Undergraduate Internet of Things Competition, China" in 2015, and the first prize in the "National Graduate Contest on Application, Design, and Innovation of Mobile-Terminal, China" in 2016 and 2017, respectively. He is a student member of the IEEE.



**Qian Wang** received the BS degree in electrical engineering from Wuhan University, China, in 2003, the MS degree in electrical engineering from the Shanghai Institute of Microsystem and Information Technology (SIMIT), Chinese Academy of Sciences, China, in 2006, and the PhD degree in electrical engineering from the Illinois Institute of Technology, in 2012. He is a professor with the School of Cyber Science and Engineering, Wuhan University. His research interests include AI security, data storage, search and computation outsourcing security and privacy, wireless systems security, big data security and privacy, and applied cryptography, etc. He is an expert under the National "1000 Young Talents Program" of China. He is a recipient of the IEEE Asia-Pacific Outstanding Young Researcher Award 2016. He is also a co-recipient of several Best Paper and Best Student Paper Awards from IEEE ICDCS'17, IEEE TrustCom'16, WAIM'14, and IEEE ICNP'11, etc. He serves as an associate editor for the *IEEE Transactions on Dependable and Secure Computing* (*TDSC*) and the *IEEE Transactions on Information Forensics and Security* (*TIFS*). He is a member of the IEEE and a member of the ACM.



**Kui Ren** received the PhD degree from the Worcester Polytechnic Institute. He is currently a professor of computer science and engineering and the director of the UbiSeC Lab, University at Buffalo, the State University of New York. He has published 200 papers in peer-reviewed journals and conferences. His current research interest spans cloud and outsourcing security, wireless and wearable systems security, and mobile sensing and crowdsourcing. He is an IEEE fellow, a member of the ACM, and a past board member of the Internet Privacy Task Force, State of Illinois. He received several Best Paper Awards, including IEEE ICDCS 2017, IWQoS 2017, and ICNP 2011. He received the IEEE CISTC Technical Recognition Award in 2017, the UB Exceptional Scholar Award for Sustained Achievement in 2016, the UB SEAS Senior Researcher of the Year Award in 2015, the Sigma Xi/IIT Research Excellence Award in 2012, and the NSF CAREER Award in 2011. He currently serves on the editorial boards of the *IEEE Transactions on Dependable and Secure Computing*, the *IEEE Transactions on Service Computing*, the *IEEE Transactions on Mobile Computing*, *IEEE Wireless Communications*, the *IEEE Internet of Things Journal*, and *SpingerBriefs on Cyber Security Systems and Networks*.

**Dimitrios Koutsonikolas** received the PhD degree in electrical and computer engineering from Purdue University, in 2010. He worked as a post-doctoral researcher with Purdue University from September to December 2010. He is currently an associate professor of computer science and engineering with the University at Buffalo, State University of New York. His research interests include broadly in experimental wireless networking and mobile computing. He received the NSF CAREER Award in 2016, the UB SEAS Early Career Researcher of the Year Award and the CSE Early Career Faculty Teaching Award in 2015, the UB SEAS Senior Teacher of the Year Award in 2017, and Best Paper Awards from SENSORCOMM 2007 and WCNC 2017. He currently serves on the editorial board of the *IEEE Transactions on Mobile Computing*. He is a senior member of the IEEE and ACM and a member of USENIX.

**Lu Su** received the MS degree in statistics and PhD degree in computer science from the University of Illinois at Urbana-Champaign, in 2012 and 2013, respectively. He is an assistant professor with the Department of Computer Science and Engineering, SUNY Buffalo. His research focuses on the general areas of mobile and crowd sensing systems, internet of things, and cyber-physical systems. He has also worked at the IBM T. J. Watson Research Center and National Center for Supercomputing Applications. He is the recipient of the NSF CAREER Award, the University at Buffalo Young Investigator Award, the ICCPS'17 Best Paper Award, and the ICDCS'17 Best Student Paper Award. He is a member of the ACM and the IEEE.

**Yanjiao Chen** received the BE degree in electronic engineering from Tsinghua University, in 2010, and the PhD degree in computer science and engineering from the Hong Kong University of Science and Technology, in 2015. She is currently a professor with Wuhan University, China. Her research interests include spectrum management for Femtocell networks, network economics, and Quality of Experience (QoE) of multimedia delivery/distribution. She is a member of the IEEE.