The Design and Implementation of a Steganographic Communication System over In-Band Acoustical Channels

TAO CHEN, City University of Hong Kong, China
LONGFEI SHANGGUAN, University of Pittsburgh, USA
ZHENJIANG LI*, City University of Hong Kong, China and CityU Shenzhen Research Institute, China
KYLE JAMIESON, Princeton University, USA

This paper presents SoundSticker, a system for steganographic, in-band data communication over an acoustic channel. In contrast with recent works that hide bits in inaudible frequency bands, SoundSticker embeds hidden bits in the audible sounds, making them more reliably survive audio codecs and bandpass filtering, while achieving a higher data rate and remaining imperceptible to a listener. The key observation behind SoundSticker is that the human ear is less sensitive to the audio phase changes than the frequency and amplitude changes, which leaves us an opportunity to alter the phase of an audio clip to convey hidden information. We take advantage of this opportunity and build an OFDM-based physical layer. To make this PHY-layer design work for a variety of end devices with heterogeneous computation resources, SoundSticker addresses multiple technical challenges including perceivable waveform artifacts caused by the phase-based modulation, bit rate adaptation without channel sounding and real-time preamble detection. Our prototype on both smartphones and ESP32 platforms demonstrates SoundSticker's superior performance against the state of the arts, while preserving excellent sound quality and remaining unaffected by common audio codecs like MP3 and AAC. Audio clips produced by SoundSticker can be found at https://soundsticker.github.io/.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing**.

Additional Key Words and Phrases: Acoustics, Mobile phones, Embedded boards, Microphones, Communications, Real-time systems

1 INTRODUCTION

Acoustic steganographic communication refers to the process of embedding bits in a sound clip (*e.g.*, a piece of music or a speech segment) and delivering them along with the original audio content through an acoustic channel in such a way that the embedded bits do not perceptibly distort the original sound clip. Unlike prior audio watermarking techniques [28, 38, 67] where the embedded bits do not survive acoustic transmission, acoustic steganographic communication ensures both the original audio content and the embedded bits are successfully received over an acoustic channel. With the wide deployment of speakers and microphones on wearables, mobiles, and cars, we envision acoustic steganography could enable a range of emerging and useful applications (details in §2), including geofenced connectivity, audio content authentication, device to device communication, etc.

*Corresponding author.

Authors' addresses: Tao Chen, Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Kowloon Tong, Hong Kong, China, tachen6-c@my.cityu.edu.hk; Longfei Shangguan, Department of Computer Science, University of Pittsburgh, 210 S Bouquet St, Pittsburgh, Pennsylvania, USA, longfei@pitt.edu; Zhenjiang Li, City University of Hong Kong, Hong Kong, China and CityU Shenzhen Research Institute, Shenzhen, China, zhenjiang.li@cityu.edu.hk; Kyle Jamieson, Department of Computer Science, Princeton University, 35 Olden St, Princeton, New Jersey, USA, kylej@cs.princeton.edu.

1550-4859/2023/5-ART1 $15.00
https://doi.org/10.1145/3587162

To enable such applications, current acoustic steganography designs [21, 22, 80] have explored the *out-of-hearing* bands (normally 18–21 kHz) and the frequency masking effect to deliver hidden bits. However, these systems suffer from multiple constraints, as listed below:

1) Hardware restriction. Processing out-of-hearing bands requires at least 44.1 kHz analog-to-digital (ADC) sampling rate, which is decided by the CPU's clock frequency. Such a rate is available on some high-end devices, *e.g.*, smartphones, while it is not viable for a body of embedded boards with low-end processing units usually (that are the more prevailing platforms in the context of Internet of Things (IoT) and can potentially leverage acoustic steganography as well), *e.g.*, 13 kHz sampling rate from an Ardunio Uno [65], 16 kHz from ESP32, etc.

2) Limited robustness. Bits can easily get lost due to common audio codecs in IoT applications (*e.g.*, speech recognition). For example, a codec like MP3 compression more aggressively compresses data on the upper edges of the audio frequency bands and low power in-band part, impacting robustness and reliability.

3) Limited data rate. Due to the ADC sampling rate constraints (even on smartphones), bandwidth of the received out-of-hearing band is normally limited or even not available (with low-end MCUs), leading to a low data rate[1], which is less sufficient to execute the connectivity management and the device-to-device communication.

4) Unpleasant acoustic noise. They may also cause unpleasant, high-pitched, perceptible noises, as the extra (and relatively strong) energy is explicitly added to the higher audible frequencies.

In this paper, we aim to advance the acoustic channel designs with standard speakers and microphones, inspired by our study of the human auditory system — the human ear is more sensitive to pitch (frequency) and loudness (amplitude) changes in an audio clip, and less sensitive to phase changes [45, 56]. When a sound propagates into the inner ear, its pitch determines the position of hair cells that respond, while loudness governs the response strength. So, leveraging frequency- or amplitude-based modulations, as adopted by prior designs [21, 22, 80], changes the hair cell response, resulting in perceptible interference. In contrast, human anatomy dictates that changes in phase do not affect the hair cell response [45]. Hence, we propose to instead modulate the phase of the audio signal in the audible (referred to as **in-band** in this paper) range of human hearing, yet surprisingly keeping the embedded bits fully imperceptible. Moreover, such an approach has three other highly desired benefits:

- First, both the in-band and out-of-hearing band can be received using a more advanced receiver (*e.g.*, on smartphones), while the former one has a wider bandwidth (*e.g.*, 0–17 kHz) than that of the latter one (*e.g.*, 18–21 kHz), which could lead to a higher data rate.
- Second, the audible band is within a low frequency range, which can thus be received even using low-end MCUs and its spectrum is still substantial (*e.g.*, the 16 kHz sampling rate of ESP32 achieves 0–8 kHz in-band spectrum). However, the out-of-hearing band cannot be received by such receivers usually, which will fail the existing acoustic channel designs. Thus, the in-band approach is compatible to a wider set of receivers for more use cases.
- Third, the embedded in-band information is tightly coupled with the original audio content and thus will survive an audio codec's compression.

To harvest these opportunities, we propose the following techniques and integrate them into a holistic stack, named *SoundSticker*. We note that SoundSticker is positioned as a crucial supplement to enrich and enhance the existing acoustic steganography family with above advantages through the in-band channel, instead of replacing the existing out-of-hearing-band designs.

1) At the transmitter side, although human ear is less sensitive to audio phase changes, a large and abrupt phase modulation may cause strong waveform artifacts (amplitude changes), which are still perceivable. We introduce a *window function* based mechanism to largely alleviate the negative perceptual effect of phase distortion.

[1]To overcome this issue, recent Dolphin design further combines part of the human perceivable bandwidth ($\geq$ 8 kHz) to increase the throughput, using energy difference keying. However, since human ear is more sensitive to the signal's intensity change [45], the included audible frequencies cannot be excessive, still limiting its overall data rate.

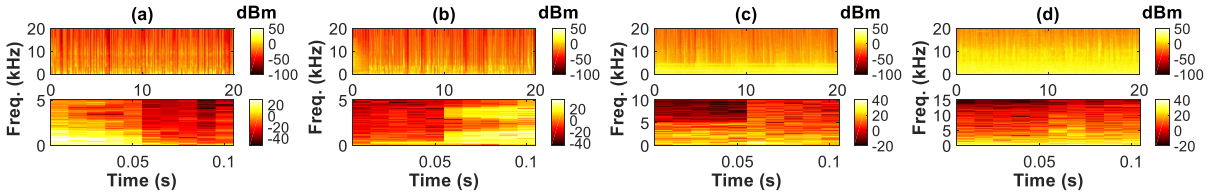

Fig. 1. Spectrogram of four typical types of common audio clips: (a): human speech; (b): news report; (c): instrumental music; (d): vocal music. For each figure in the first row, the figure below in the second row depicts one zoom-in snapshot.

2) The hidden data are encoded in the source audio, where the sound energy varies drastically with the audio content itself. We find that in this scenario the quality of received signals is dominated by the power distribution of the carrier (i.e., audio content) as opposed to the channel. This observation motivates us to design a power-aware rate selection mechanism to further enhance the link throughput.

3) At the receiver side, because SoundSticker uses differential phase-shift keying, we cannot detect its packet preamble with conventional cross-correlation directly. Without the preamble's position, the embedded bits cannot be decoded. To address this issue, we propose an effective and lightweight preamble search method, by leveraging audio phases' constructive and destructive patterns. When the preamble patterns appear, the bits can get decoded.

We integrate above transmitter and receiver designs into a SoundSticker prototype system on both smartphone and low-end ESP32 platforms. Results show the acoustic channel can achieve up to 306 bps goodput on the smartphone and ESP32 platforms in a typical office respectively, outperforming the state-of-the-art Dolphin [80] by 1.3× on smartphones (Dolphin is not compatible to the ESP32 platform). SoundSticker also preserves excellent sound quality and is insusceptible to standard audio codecs like MP3 and AAC. With SoundSticker's encoded bits, the transcript of the original audio content can still be recognized correctly by commercial speech-to-text tools, *e.g.*, Google [71], iFLY [73], HappyScribe [72] and Transcribe (Wreally) [74]. Evaluation further reveals SoundSticker can achieve promising goodput gains under various settings, *e.g.*, different transmission distances, noise levels, receiver moving speeds, and device diversity. Finally, as a case study, SoundSticker achieves nearly 100% success rate to avoid connections outside room boundaries on both platforms, which shows strong support for the geofenced connectivity. In summary, this paper makes the following contributions.

- We systematically study the audio channel's effect on acoustic steganographic communication, explore its characteristics, and design effective and tailored techniques to address the challenges in SoundSticker over in-band acoustic links.
- We engineer a functional prototype on two platforms, which entails a full PHY-layer stack, rate selection, and transceiver implementation. We conduct extensive field studies, micro-benchmarks and compare with the state-of-the-art in various settings.

2 APPLICATIONS

1) Geofenced connectivity. Acoustic waves propagate within human earshot and attenuate at room boundaries sharply [40]. Hence, they can be adopted to enable the geofenced connectivity inside the same room space[2]. For example, customers of a coffee shop are allowed to access its free Wi-Fi inside the store automatically, and meanwhile people from cross-wall restaurants are prevented to be free riders of this Wi-Fi service. Such a geofenced connectivity service can be also useful in other company or home scenarios, *e.g.*, allowing a public wireless access in certain areas only, like the lobby, meeting room, living room, etc., or limiting the data rate in these areas. In such applications, the acoustic steganography serves a side-channel to verify the room-area connectivity with the smartphone to provide the geofenced connectivity service for wireless, *e.g.*, Wi-Fi. Such geofenced connectivities could also be enabled with embedded devices such as Internet of Things (IoT) nodes, smartwatches, Airpods [4], or

[2]The geofenced connectivity is poorly served by the existing wireless, *e.g.*, Wi-Fi, Bluetooth, Zigbee, LoRa, etc.

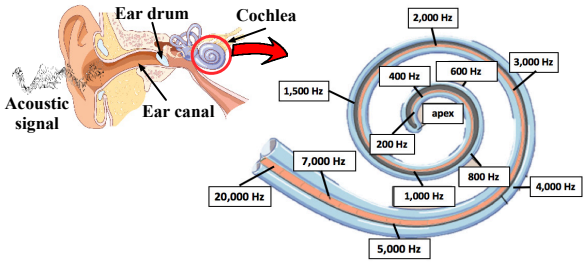

Fig. 2. The illustration of our human auditory system. Figure adapted from [39].

Fitbit [24]. As the acoustic steganography is only used for providing verification messages usually, the demand on the goodput is light, *e.g.*, tens of bits per second. And the goodput of SoundSticker (306 bps on smartphone and 55 bps on MCU) can support it smoothly.

With a wide adoption of speakers on smart devices [69], it becomes more frequent to hear audible sounds in our daily life, such as news reports, background music played in a room, airport and aircraft announcements, interactive messages (*e.g.*, "welcome", "connecting", etc.) sent from voice-controlled user interfaces like Alexa, *etc.* This observation inspires people to prioritize using the in-band acoustic channel upon these audible sounds to initialize the room-area connectivity. On the other hand, we also need to consider the possibility that audible sounds may not be always available, whereas a steganography message still needs to be transmitted. In such a case, the existing out-of-hearing band acoustic channel designs, such as [80], could serve as a remedy solution, not associated with an audible sound. As a result, SoundSticker is positioned as a crucial supplement to enrich and enhance the existing acoustic steganography family with unique advantages (§1) through the in-band channel, instead of replacing the existing out-of-the-hearing designs.

2) Audio content authentication. Audio adversarial attacks [10, 13] now can slightly modify the amplitude of an audio command (imperceptible to people) to fool the receiver's speech recognition function, so that the original command transcript can be recognized as any other targeted one. Similarly, speech synthesis techniques can also easily generate fake speech [41, 76] for any targeted person played in public. To prevent these attacks, a speaker can sign a transcript of the sound clip (or a hash thereof) digitally and deliver this signature over the acoustic-steganographic channel along with the audio content. Receivers or audiences can verify the digital signature and compare what they hear with the digitally signed transcript to authenticate the audio content.

In such applications, the audio transcript is first hashed into an output string (with a fixed length of n) using Secure Hash Algorithm (SHA), where n varies between 160 and 256 bits based on the SHA algorithm being adopted (SHA-1, SHA-256) [44]. The n-bit hashed string is then encrypted using the RSA algorithm, which does not change the string length [8]. In our current design, all n bits of encrypted data are carried by one SoundSticker packet with 1.2 s packet length (§3). As a result, the goodput requirement is around 133–213 bps, lower than the goodput that SoundSticker provides (306 bps).

3) Device to device communication: The steganographic links can be set up among nearby devices for the information sharing [61], including both smartphones and embedded devices. For instance, the covert communication among IoT devices in close proximity to avoid eavesdropping. In shopping malls, coupons are streamed all the time, and users can receive different ones after entering each shop for precise advertising. When a piece of music or an advertisement is played on TV, the useful side-channel information, *e.g.*, the music transcripts, URL to buy the advertised product online, etc., can be delivered from TV to different receivers, *e.g.*, smartphones, watches, Fitbits, etc. For side-channel information retrieval and device-to-device communication, the maximum achievable goodput of SoundSticker is 306 bps, which is equivalent to 38 English characters per second (8 bits per character). Such

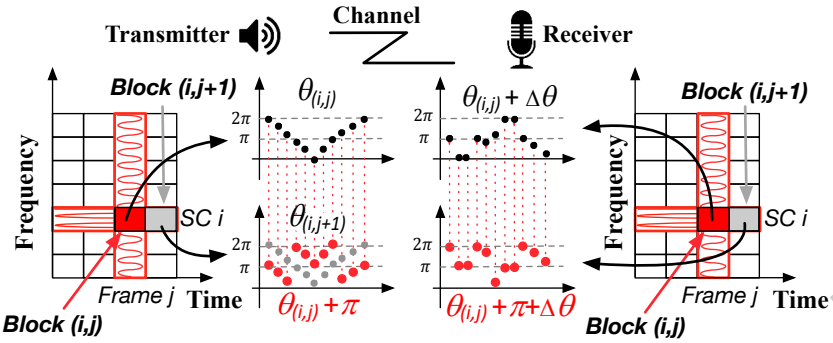

Fig. 3. OFDM operation (SC *i* means Subcarrier *i*) and DBPSK modulation in SoundSticker, which encodes a bit '1' by shifting $\theta_{i,j+1}$ (gray) to $\theta_{i,j}+\pi$ (red). $\Delta\theta$ is the phase shift introduced by the channel distortion and sampling time offset.

amount of data is sufficient for delivering music transcripts, URLs, copyright information, and control messages in real-time.

3 SYSTEM DESIGN

This section elaborates on the SoundSticker design at both the transmitter (§3.1) and receiver (§3.2) sides.

3.1 Transmitter Design

In the following part, we introduce three modules of the PHY-layer designs, including the modulator (§3.1.1), windowing (§3.1.2) and rate selection (§3.1.3).

3.1.1 Modulator. Acoustic wave (see Figure 1) is quasi-stationary within a short time (2–50 ms) [91]. After it enters the ear, our auditory system divides it into small segments and performs a time-frequency analysis on each segment similar to the Short-time Fourier Transform (STFT) [91]. Specifically, the acoustic wave first enters the outer ear and propagates along the ear canal down to the ear drum, making the ear drum vibrate. Vibration then moves into the inner ear *cochlea* and stimulates its hair cells to generate a neural response. Hair cells lie along the entire length of the cochlea, and each position responds to a specific range of vibration frequencies. In simplified terms, the cochlea can thus be viewed as a bank of *band-pass filters* (Figure 2), each passing frequencies within a *critical band*: two tones separated by more than a critical bandwidth are perceived independently.

1) Why human ears are not sensitive to phase changes? Each hair cell works like a combination of a transducer and ADC (analog to digital converter). It converts the continuous mechanical sound wave to the digital electrical signal for brain [59]. Because the hair cells have *low sampling rates and low-resolution ADCs* [12, 25], only the low-frequency sound (<1–2 kHz [57, 59]) can be well captured and reconstructed due to the Nyquist criteria. As the majority part of the audible band is higher than 2 kHz, the modulated phase information in the audible band is thus hardly perceived by human ears.

2) Phase-oriented modulation. SoundSticker employs Orthogonal Frequency Division Multiplexing (OFDM) to improve the spectrum utilization, based on which we adopt Differential Phase Shift Keying (DPSK) to modulate bits. In SoundSticker, we divide the sound wave into frames and select 1,500 audio samples, corresponding to the frame length L_{frame} of 34 ms. The overall bandwidth B_{whole} is 16.2 kHz and subcarrier bandwidth B_{sub} is 450 Hz, leading to 36 subcarriers. All these OFDM parameters are selected based on our study of the critical bands of our ears, which are detailed in §4.

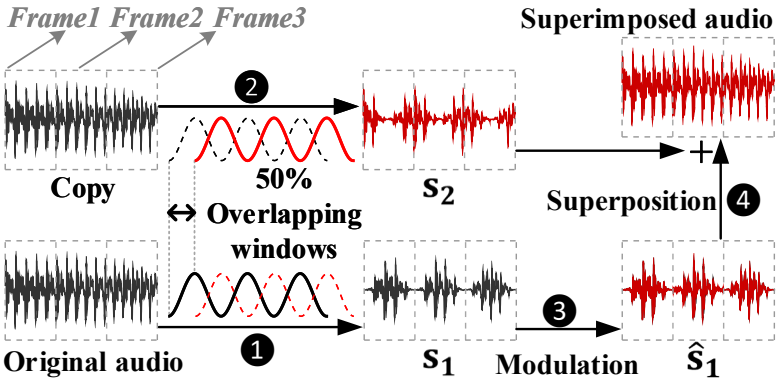

Fig. 4. An illustration of the "analysis and synthesis" two-stage operation for the waveform artifact suppression.

For each subcarrier i, let $\theta_{i,j}$ and $\theta_{i,j+1}$ be the phase values of the time-frequency blocks on the j^{th} and $(j+1)^{th}$ frames, respectively. Both $\theta_{i,j}$ and $\theta_{i,j+1}$ represent a group of phase values. With a 34 ms frame length and a 450 Hz sub-carrier bandwidth (§4), the total count of phase value in both $\theta_{i,j}$ and $\theta_{i,j+1}$ is 15. Figure 3 (left) shows an example of the phase values in both $\theta_{i,j}$ (black) and $\theta_{i,j+1}$ (gray).[3] In the slowest Differential Binary Phase-Shift Keying (DBPSK), we encode one bit by shifting all phase values in $\theta_{i,j+1}$ (gray) to a new value $\theta^*_{i,j+1}$ (red), such that $\theta^*_{i,j+1} = \theta_{i,j} + \theta_{shift}$, where $\theta_{shift} = 0$ means a bit '0', and $\theta_{shift} = \pi$ means a bit '1'.

Higher modulations adopt smaller θ_{shift} values, *e.g.*, $\frac{\pi}{2}$ for Differential Quadrature Phase-Shift Keying (DQPSK). We choose DPSK over other modulation schemes due to three considerations:

a) Preserving modulated information. Original phase values in each block are unknown by the receiver, but the signal phases transmitted over two consecutive blocks will experience a similar phase shift $\Delta\theta$ caused by the channel [14, 58]. Subtracting $\theta_{i,j}$ from $\theta_{i,j+1}$ thus naturally cancels out $\Delta\theta$, preserving the modulated information.

b) Canceling phase offset. Oscillator inaccuracies at both sender and receiver introduce sample time offsets (STO), which are exacerbated in our setting due to the relatively low sampling rate on mobile or IoT devices, resulting in relatively large phase shifts. Similarly, phase subtraction also cancels out this STO-induced phase shift.

c) Simplifying receiver design. DPSK does not require coherent detection and frequency synchronization, thus alleviating receiver design's complexity [30, 75].

3) Compared with prior phase modulations. Although there exist some prior works that have adopted phase-based modulations to build covert channels [14, 21, 22], all of them leverage the *frequency masking effect* to overlay the phase modulated information on a very narrow frequency band of an audio clip; thus yield a lower data rate. The frequency masking effect requires a high pitch tone residing in one frequency band to hide the data embedded in nearby low pitch frequency band, which rarely happens over the entire human auditory band. Beside, the consistently high pitch appears only in certain types of audios (*e.g.*, music), which severely limits the applicability of these covert channel designs. Instead of leveraging the frequency masking effect to overlay the modulated information on the source audio clip, SoundSticker directly modulates the phase of the source audio clip over a large, consecutive audible frequency band. As we experimentally demonstrate in §5, SoundSticker can achieve consistently high data rates over diverse types of audio clips.

3.1.2 Windowing. While the phase modulation generally ensures good imperceptibility of each encoded frame, it may introduce an abrupt phase change at the frame boundary, resulting in strong, human perceptible waveform artifacts (*i.e.*, amplitude variation). Figure 5 shows a concrete example, where the boundary of two frames is at 34

[3]The actual phase values in $\theta_{i,j}$ and $\theta_{i,j+1}$ are arbitrary due to the unrelated sound peaks, here we use the same pattern only for illustration.

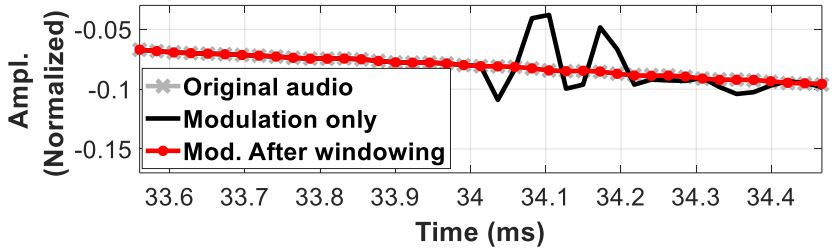

Fig. 5. Frame boundary amplitudes for the original audio clip, the audio clip after modulation, and the audio clip after modulation and windowing.

ms and we modulate the second frame using DPSK. We can see that the signal amplitude (black line) close to the frame boundary (34-34.2 ms) fluctuates significantly after applying DPSK modulation. To solve this problem, we employ an *analysis and synthesis* two-stage operation to suppress the waveform artifact on the frame boundary without sacrificing the audio quality.

1) Analysis stage. We make a copy of the original audio clip and apply a Hanning window[4] to the copy and itself, separately. These two Hanning windows overlap with each other by 50%. Due to the window's "bell" shape, the power on the frame boundary attenuates significantly, as shown in Figure 4. We denote these two audio clips (after windowing) as s_1 and s_2 separately. SoundSticker then modulates (③) each frame in s_1, yielding a new audio clip Θ_1.

2) Synthesis stage. In the synthesis stage, SoundSticker superimposes Θ_1 and s_2 frame by frame (④). Due to the 50% window overlap, the superposition of these two audio clips results in the energy on the boundary of the superimposed frame reverting to the same power level as in the original audio clip. The central part of each superimposed frame, on the other hand, stays consistently at a high level. Figure 5 shows the signal samples on the frame boundary of the superimposed audio clip (red line). We can see windowing successfully eliminates waveform artifacts while aligning the power of the superimposed audio clip to that of the original audio clip.

We note that this windowing operation will alter the phase of each modulated frame, and thus undermine the modulation result. However, by applying the same Hanning window on each received frame, the receiver could successfully remedy the phase shift introduced by the windowing operation on the transmitter and decode the embedded bits accordingly.

3.1.3 Rate selection. Higher SNRs support higher data rates, which are naturally determined by two factors — acoustic channel and source sound energy. To quantify this relationship in acoustic channels, we empirically measure the mapping between the received audio content power and the resulting Bit Error Rate (BER). We vary the received power over a high dynamic range $[-40, 32]$ dBm[5], and report results in Figure 6. When the received power is low (< -20 dBm), even DBPSK leads to a high BER. Within a moderate range ($[-20, 7]$) dBm, DBPSK is reliable, yet DQPSK is unreliable. When the received power is strong enough (> 7) dBm, DQPSK achieves a low BER as well. However, D8QPSK suffers high BER all the time.

Hence, SoundSticker can adopt DBPSK and DQPSK to encode bits, with the following bit rate selection rule. The rate is selected for each subcarrier based on its average power P_{recv} (as audio signals are converted to the frequency domain to encode the bits, P_{recv} can be easily computed from the frequency spectrum), divided into *High*, *Moderate*, and *Low* three levels as follows:

1) Problem. With this rule, one immediate question, however, is raised. As the receiver is not aware of the source sound energy for the incoming packet, the channel sounding is hardly to be conducted, so that P_{recv} is unknown

[4]We have tested 12 different types of window functions, among which Hanning window leads to the best performance as shown in §5.
[5]Measured from 20 different audio clips, including musics, news reports, speeches, etc.

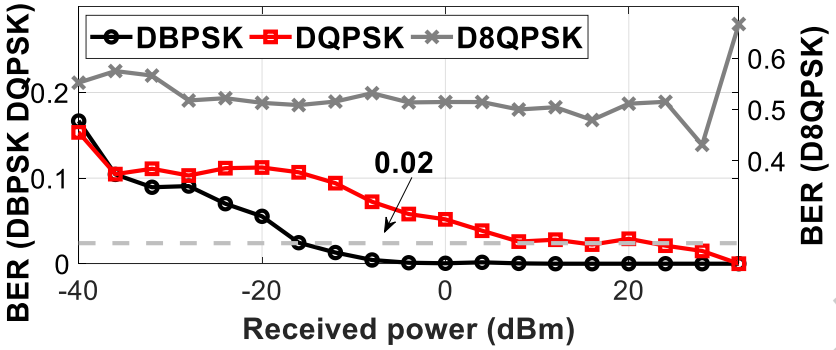

Fig. 6. Received power vs. BER under three modulations. The high BER of D8QPSK is plotted on a separated y-axis.

Level	*High*	*Moderate*	*Low*
Condition	> 7 dBm	[-20,7] dBm	< -20 dBm
Modulation	DQPSK	DBPSK	Ignore

Table 1. Bit rate selection in SoundSticker.

by the sender. We could, of course, always adopt the slowest modulation for all subcarriers, but it can be highly conservative. In SoundSticker, we instead let the sender locally select the rate on each subcarrier without feedback from the receiver, through an understanding of room acoustical channel characteristics. We observe P_{recv} might be highly dynamic over both time and frequency depending on the communication channel. It is however still dominated by the power distribution of audio content itself.

2) Opportunities for the sender-selected modulation. We record four typical types of audio clips that may appear in SoundSticker. Figure 1 depicts their spectrum (first row), from which we observe that the power distribution of the audio content itself seems highly uneven. To quantify such an observation, we investigate as follows:

- We first plot the CDF of the max-minus-min power differences across subcarriers from each time frame (*e.g.*, 34 ms as stated in §4) for these audio clips ("Original") in Figure 7(a). The result shows that the audio content itself has a highly diverse power distribution, up to 114 dB.
- Next, we transmit these audios over the air in various scenarios. Figure 7(a) further shows the max-minus-min differences of the received signal powers for the same audio frame are quite similar to each other, which are also close to that from original audios.
- This observation suggests that the uneven power distribution of the audio content itself could dominate the in-band received power diversity, which is much stronger than the frequency selectivity caused by the acoustic channel attenuation [52].

This observation implies that both the sender and receiver can see similar shapes of audio's power distribution, as the receiver receives what the sender plays, but their absolute values are different due to attenuation. In Figure 7(b), we quantify the attenuation by calculating the power ratios for the same time-frequency division block between the original audio files and the received ones. The *sharp* transitions of all CDFs imply the power's attenuation can be described by a scaling factor approximately. In Figure 7(b), when distance varies from 0.5 to 8m, scaling factors could vary from 0.38 to 0.67.

Considering the steganographic communication scenario, devices are normally within a reasonable distance (*e.g.*, several meters). Hence the audio power received by the receiving devices would not be extremely weak. To verify this, we vary the ear perceived loudness level of an in-air audio (measured by a sound pressure meter [70]) from 55

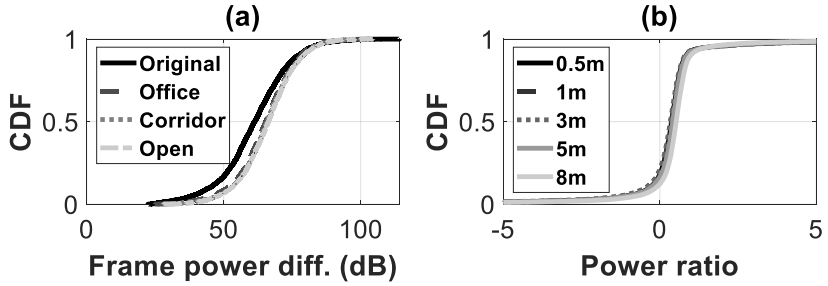

Fig. 7. (a) CDF of max-minus-min powers for original and received (over the air) audios in office, corridor and an indoor open space; (b) CDF of power ratios at different tx-rx distances.

Hearing (dBSPL)	55-60	60-65	65-70	70-75
Power ratio	**0.48**	**0.56**	**0.67**	**0.72**

Table 2. Power ratio between the received signal and the audio file signal in different hearing loudness levels.

to 75 dBSPL and calculate the power ratio between the received signal power and the signal power in the original audio file. The result is shown in Table 2. We find that even in the lowest hearing loudness category, the power ratio is close to 0.5, which inspires the following simply yet practical design:

- In each packet (bits are organized as packets for transmission in §4.2), the sender computes the mean power P_j for each subcarrier j, which then multiplies by a power ratio r (0.5 by default).
- The sender can thus use $r \times P_j$ to (conservatively) approximate the received power in this subcarrier for rate selection (Table 1), as sender and receiver observe similar audio power distributions.

3.2 Receiver Design

This component runs on the receiving device to detect the incoming signals and decode the hidden bits. We detail our design towards this goal in SoundSticker.

3.2.1 Preamble Detection. The preamble detection functionality scans incoming signals, detects the beginning of a packet, and synchronizes the frames prior to the decoding of the hidden bits.

1) Problem. A challenge here is that the packet preamble is also encoded by DPSK modulation and thus cannot be detected through the conventional cross-correlation [75] directly.

2) Proposed solution. Motivated by Schmidl-Cox algorithm [66], we introduce a sliding window that can hold two consecutive frames for phase subtraction, which moves to search for the preamble. Inside each window i, we calculate the phase shift of two consecutive blocks on every subcarrier j, namely, $\theta^{i,j}_{shift}$. We then pick the phase readings within the first 36 blocks (corresponding to the preamble, as shown in Figure 8(a)), and then correlate them with the phase shift values of the defined preamble:

$$d_i = \left| \sum_{j=1}^{M} e^{j(\theta^{i,j}_{shift} - \Theta^{j}_{shift})} \right|, \tag{1}$$

where M equals 540 (36 preamble blocks with 15 phase values for each) and Θ^{j}_{shift} represents the phase shift of the j^{th} bit in the preamble. This equation measures how likely the signal in the i^{th} sliding window is the preamble. The existence of a preamble will lead to a higher amplitude of the superimposed signal (as Figure 8(b) shows). In contrast, if the sliding window does not contain the preamble, the terms in the above equation cancel each other,

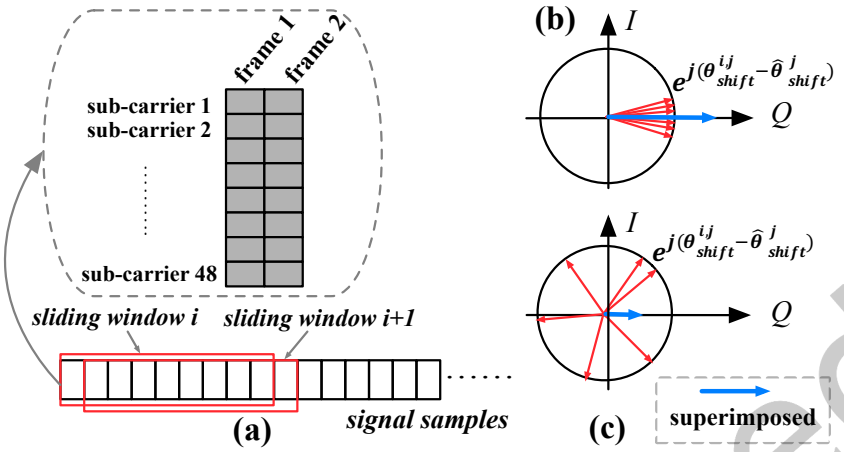

Fig. 8. (a) Preamble detection. (b) When preamble is in the current sliding window, signals are constructively combined. (c) Otherwise, they are destructively combined.

Search attempt(s)	1	2	3	4	5
Detection success	85%	93%	98%	99%	100%

Table 3. The cumulative distribution of the search attempt for preamble detection.

making the amplitude of the superimposed value smaller (as shown in Figure 8(c)). Hence the preamble is most likely to be located at the sliding window index that results in the highest amplitude of the superimposed signal.

A receiver may start receiving audios in the middle of one packet. If the search lasts for the whole packet duration (1.2 s), the preamble can be located, but this process is computationally intensive. To reduce the computation overhead, SoundSticker's sender transmits a bit sequence over the pilot subcarrier (the last subcarrier at 1.5 kHz). The sequence alternates between consecutive "1"s (for one packet) and "0"s (for the following packet). After energy detection has finished, the receiver first applies Eqn. (1) to the pilot subcarrier only. It can thus identify the frame boundaries and demodulate subsequent pilot bits. An appearance of a bit flip between two consecutive pilot bits indicates the incoming of a new packet, the receiver then performs correlation on the 36 subcarriers to detect preamble and aligns the packet. Two points are worth noting:

- The search for the pilot subcarrier is based only on 15 phase values in one block, the alignment is thus coarse. A preamble is still needed to tightly align the receiver's time to ensure decoding performance for subsequent packets.
- Bit flips may happen due to demodulation errors, while the preamble cannot be found in this case and the receiver thus waits for the next flip. As the pilot bits are modulated by DBPSK and the signal power at 1.5 kHz is generally strong, errors rarely happen.

We place a transmitter and a receiver one meter away from each other to evaluate the effectiveness of this algorithm. The sender transmits packets, and the receiver starts to record and decodes at a random time during the sender's playing. We then count the number of bit flips the receiver encounters before finding the packet preamble (termed as search attempts). The result is shown in Table 3. We observe that for more than 93% cases, the preamble can be found after the first two search attempts, with the maximum five attempts in our experiment.

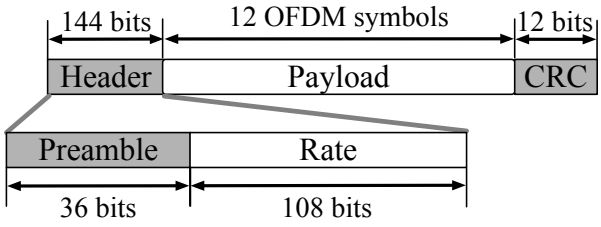

Fig. 9. Packet structure of SoundSticker.

3.2.2 Demodulation. Once the preamble is detected, the receiver locates the *rate* field in the packet header to apply their indicated setting to demodulate bits. In particular, for each consecutive block pair, the receiver subtracts their phases to obtain 15 phase shifts θ^i_{shift}, where $i \in [1, 15]$, and then applies a majority voting for the phase shift to demodulate the coded bits.

- For DBPSK, the phase shift 0 indicates a '0' bit, and the phase shift π indicates a '1' bit. If more than half of θ^i_{shift} are closer to $\frac{\pi}{2}$, the bit is demodulated as '1'; Otherwise, it is a bit '0'.
- The majority voting operation can be similarly applied to other modulations. For DQPSK, the phase shifts 0, $\frac{\pi}{2}$, π, $-\frac{\pi}{2}$ indicate bits '00', '01', '10' and '11', respectively.

The current SoundSticker design is primarily for the broadcast scenario because some transmitters (*e.g.*, television and loudspeaker) may not have microphone to get the receiver's acknowledgement. Hence, the packet re-transmission (due to the decoding error) is not considered explicitly in the current SoundSticker, while we plan to enable this mechanism in the future for the transmitters having such a receiving ability.

4 IMPLEMENTATION

We implement SoundSticker on the Android and the low-end system-on-chip (SoC) two types of platforms, which include all the components introduced in §3. Before the valuation, we first elaborate the key parameter settings of SoundSticker in our implementations.

4.1 OFDM Operation

SoundSticker employs OFDM to improve the throughput and we configure the key OFDM parameters based on the characteristics of sound waves and our auditory system.

Time division. With OFDM, we divide the sound wave into *frames*. To determine the frame length, in Figure 1, we zooms in each spectrum of the four audio clips in second row. Consistent with prior studies [91], we observe that each spectrum stays stable over 30–50 ms. We currently select 1,500 audio samples, corresponding to 34 ms, as the default length.

Frequency division. On the other hand, we choose the overall and subcarrier bandwidths with the following considerations:

1) Overall band B_{whole}: The voice or music spectrum is usually less than 17–18 kHz [80], and the signal energy above this range is too weak for communication. On the other hand, through our study in §3.1.1, we find that even small phase distortions at $<$1.5 kHz are easily perceivable. So, we select B_{whole} = 16.2 kHz (1.5–17.7 kHz).

2) Subcarrier band B_{sub}: Psychoacoustical studies [90] find that the critical bands of human auditory system increases with the frequency, as the ear is more sensitive to low frequency bands. Within the low-frequency bands, the critical bandwidth is within 320 to 550 Hz, and we thus select B_{sub} = 450 Hz, leading to 36 subcarriers for data delivery and one subcarrier as a pilot.

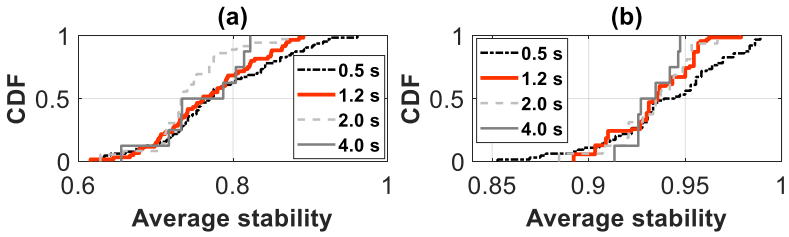

Fig. 10. Audio power temporal stability on (a) speeches and (b) musics.

4.2 Packet Design

All the message bits are transmitted in a form of packets in SoundSticker, which is designed as follows:

Packet structure. Each packet consists of three fields, including a header, a payload, and a cyclic redundancy check (CRC), shown in Figure 9. The *header* contains a 36-bit *preamble*, followed by a 108-bit *rate* indicating the bit rate on every subcarrier. The *payload* consists of 12 OFDM symbols, which is determined based on the packet length L_{pkt} as stated below. We finally append a 5 ms cyclic prefix (CP) to each OFDM symbol to eliminate inter-symbol interference (ISI). Each packet ends up with a 12-bit CRC. The audio clip is divided into multiple equal-length time slots (equivalent to the packet length 1.2 s as stated below). SoundSticker skips those "empty" slots where the average power of the audio content is lower than −30 dBm.

Packet length L_{pkt}. We select L_{pkt} as a valid time span to compute the average received power, *e.g.*, "coherence time", for the bit rate adaptation. To quantify this temporal stability, we segment two typical type of audio clips (speeches and musics) using time windows. Let t_{win} and t_{frame} be the window length and frame length (34 ms), respectively. Each window thus contains t_{win}/t_{frame} frames. We further denote $n_j(t)$ as the block count of the same received power level (high, moderate or low) that appears most in the j^{th} subcarrier within t_{win}, then the normalized $\overline{n}_j(t) = n_j(t)/(t_{win}/t_{frame})$ can thus describe the temporary power stability in the j^{th} subcarrier within window t_{win}. For instance, if one subcarrier stays at the same power level within time t, $\overline{n}_j(t) = 1$, which means it is very stable. We can further compute the average stability $\overline{n}(t) = \sum_{j=1}^{S} \overline{n}_j(t)/S$ across all subcarriers, where S is the total subcarrier amount.

We select the packet length as an appropriate value of window t_{win}. Figure 10 plots the CDF of $\overline{n}(t)$, where t_{win} varies from 500 ms to 4 s. We can see that as t_{win} increases, the stability's variance becomes smaller, while average stability also decreases. The degraded temporal stability implies a less valid time span ("coherence time") to compute the average received power for the bit rate adaptation. On the other hand, a short packet (small t_{win}) leads to a lower goodput (due to more packet header overhead), which hurts overall performance as well. Considering both aspects, we adopt packet length L_{pkt} = 1.2 s in current SoundSticker.

4.3 SoundSticker on MCU

We also implement the SoundSticker receiver on a low-end platform using an ESP-WROOM-32 DivKit (ESP32) micro-controller and an INMP441 microphone, which are connected by the digital I2S bus as shown in Figure 11(a).

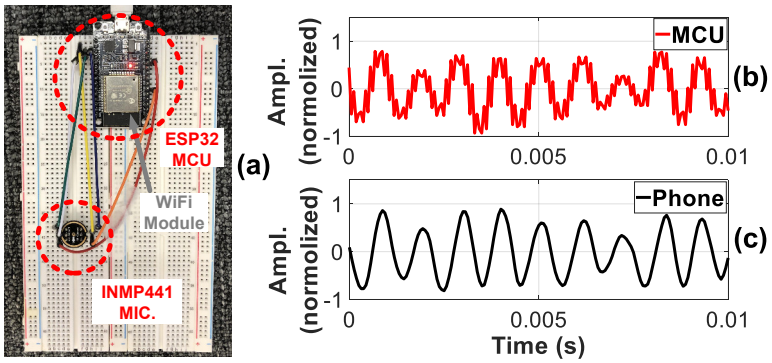

Fig. 11. (a) SoundSticker receiver on a low-end platform. A same audio is received by (b) low-end and (c) phone receivers.

Through our implementation, we find that some system settings should be tuned as follows to better fit this low-end platform.

Bit rate. Due to the low cost of hardware, we find that the quality of the received audio wave is much lower than that from a smartphone. For instance, Figure 11(b) shows that the received audio wave contains obvious noises [6], while the same audio can be received with a high quality using a smartphone as Figure 11(c) depicts. With such noises, we find that the decoding error rate is high if DQPSK is used. As a result, we only adopt DBPSK on this platform.

Parameters. The sampling rate on ESP32 is up to 16 kHz, leading to an effective frequency band at 0–8 kHz. With this hardware constraints in mind, we configure B_{whole} to 5,300 kHz (1,500–6,800 Hz) and B_{sub} to 310 Hz, which leads to 17 subcarriers in total. Moreover, we deliberately add an extra 1280 ms waiting slot on the packet and adjust the frame length as 1,024 samples to account for the MCU's limited computation capability. Each packet thus lasts for 3340 ms on this low-end platform. In practice, such parameters can be easily adapted to the type of transmitter-receiver pair automatically.

Processing overhead. On our MCU-based prototype, the average latency to decode the header, payload, and packet detection is 58 ms, 380 ms, and 1800 ms, respectively. The overall latency of packet reception is 2238 ms, mainly from the relatively computationally intensive correlation process of packet detection.

5 EVALUATION

We evaluate SoundSticker on different platforms, including a speaker (HiVi M200MKIII), multiple smartphones (Samsung Galaxy S7, Xiaomi Mi Note3, and Apple iPhone) and a low-end MCU receiver. Hidden bits are delivered over four types of audio content, including human speech, news report [7], instrumental [85] and vocal music [11, 51, 62], which covers the typical audio signals in our ambient environment with diverse energy distribution. All experiments are conducted in a typically office environment, as shown in Figure 12. In total, over 50 hours of audio data are used in the evaluation.

5.1 Field Study

We first conduct field studies to understand the system performance in real world settings. In particular we compare SoundSticker (ST) with Dolphin (DH) [80], a recent and representative out-of-hearing band design. Dolphin

[6]The noises come from the low bit depth (16 bit) [1] adopted by an MCU platform with the inherent constraints on CPU's clock frequency. Low bit depth results in inaccurate measurement of each sample in the ADC quantization [6, 16]. For comparison, a smartphone adopts 24 bits or 32 bits normally [9].

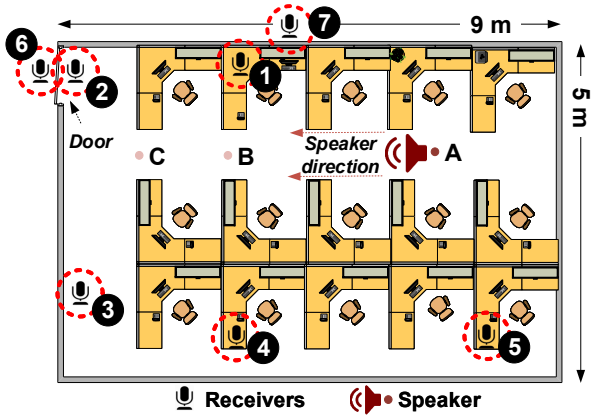

Fig. 12. Floorplan of the filed study.

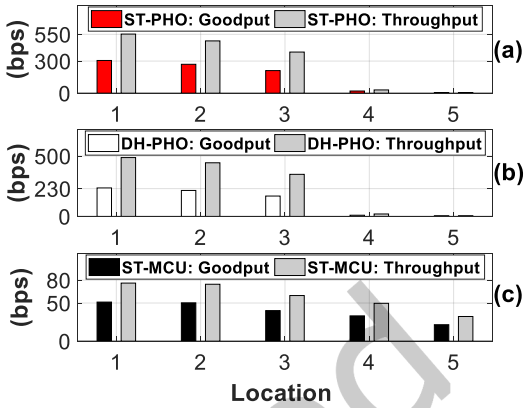

Fig. 13. Throughput and goodput at five positions in the office: (a) ST and (b) DH with phone receivers; (c) ST with MCU.

proposes an amplitude shift keying (ASK) and energy differential keying (EDK) two approaches to modulate data. It combines the upper edges of the audio band and part of the human perceivable band ($\geq$ 8KHz) to further increase the throughput.

5.1.1 Throughput and goodput. We first examine the transmission performance of ST and DH over their own acoustic channels. As shown in Figure 12, we place a speaker at position A and then investigate the the throughput and goodput using both smartphone and low-end MCU at all the position inside the office ("1" to "5" in Figure 12). We then tune the volume of speaker to 70 dBSPL as the default.

Higher throughput and goodput[7] are always desired, because 1) redundancy can be added to the packet payload, so that the message bits will have a higher chance to be decoded successfully; and 2) more bits can be delivered within one unit time in device-to-device communications. In particular, with a relatively clear line-of-sight path between each transmitter and receiver pair at positions "1" and "2", ST with phone ("ST-PHO") achieves 553 bps overall throughput and 306 bps goodput, 1.3$\times$ higher than DH with phone ("DH-PHO"). Their performance gap is primarily due to the advantages that the in-band spectrum provides a wider bandwidth and our rate selection design to transmit the hidden bits. With a moderate non-line-of-sight (NLOS) blocking at position "3", "ST-PHO" still outperforms "DH-PHO". With severe NLOS blocking at position "4" and "5", the throughput is decreased because we adopt a directional loudspeaker in the experiments, while positions "4" to "5" are far away from the speaker's directional beam. Moreover, the speaker is deployed at a height of about 0.5 m above the ground, so that the office cubicles can cause the strong NLoS paths to these positions.

On the other hand, with the low-end MCU receiver, ST with MCU ("ST-MCU")[8] can achieve good throughput and goodput performance at all the positions. At positions "1" and "2" with line-of-sight paths, the throughput and goodput are about 77 bps and 52 bps, respectively. At positions "3" to "5" with non-line-of-sight paths, the throughput slightly decreases to 60–33 bps and the goodput decreases to 40–22 bps. We observe that the low-end MCU platform can achieve better performance than the phone receivers at positions "3" to "5". This is because the in-band spectrum with the phone receiver (1.5–17.7 kHz) is much larger than that with a MCU receiver (1.5–6.8 kHz). The attenuation at the higher frequency range is relatively stronger, making it more vulnerable

[7]Goodput [31] measures the number of the bits carried in the payload (excluding the header and CRC overhead from *throughput*) delivered to the receiver per unit of time, which is accounted from the correctly decoded packets only.
[8]Because the received frequency band is only 0–8 kHz with the ESP32 MCU, DH is not compatible to such a low-end platform.

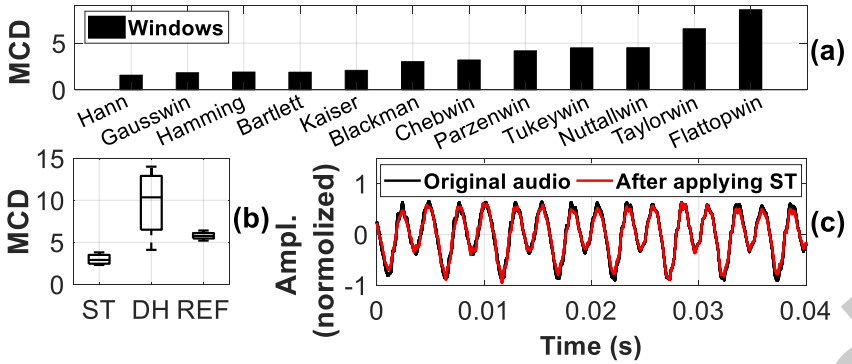

Fig. 14. (a) Audio qualities of SoundSticker using various window functions; (b) MCD of three encoding schemes (a lower MCD indicates a better sound quality); (c) Audio waveform comparison for SoundSticker.

to such poor channel conditions. To improve the data delivery at these challenging spots inside the room, we can use an omni-directional speaker and also deploy it at a higher place (to alleviate the NLOS blocking). The transmitter-receiver pair can be switched to the parameters of the MCU platform as well, if the phone's parameters keep incurring the decoding failure. We will explore these opportunities as a future work of this paper.

5.1.2 Sound quality. Next, we compare the sound quality between Dolphin and SoundSticker. We first calculate the Mel Cepstral Distortion (MCD)[9] of these two methods and show the result in Figure 14(b). A low MCD value indicates a better sound quality.

1) MCD comparison. In Figure 14(a), we examine 12 popular window functions and select the Hanning as the default window function in SoundSticker (§3.1.2) that leads to the best audio quality (the smallest MCD value). For comparison, we further synthesize human speech for a particular targeted person using a popular speech synthesis algorithm [15]. We then record the real speech from this targeted person and compute against the MCD value of the synthesized human speech. This MCD value is then used as the reference (*REF* in the figure). From Figure 14(b), we can see that SoundSticker achieves the best sound quality, with an average MCD of 3.0, which is even 2.5 smaller than the REF's MCD (5.5). In contrast, we see the MCD value of Dolphin is almost 3× higher than SoundSticker.

The result in Figure 14(b) demonstrates that SoundSticker achieves both higher goodput and better sound quality than Dolphin. We further plot the audio signal before and after applying SoundSticker in Figure 14(c). We see that SoundSticker's encoded audio does not include any amplitude artifacts.

2) Scoring the sound quality. While the MCD value reveals the sound quality, it does not reflect the imperceptibility of the encoded bits. Hence, in this experiment with the institutional review board (IRB) approval, we invite 60 volunteers (30 female and 30 male with diverse ages ranging from 20 to 51) to score the imperceptibility of the encoded audio clips. To do so, we ask each volunteer to listen to the source and SoundSticker's modulated audio clip randomly chosen from our audio dataset. The volunteer then selects the modulated audio from the two audio clips and give a score describing how she feels about the difference between these two sound clips using the criteria in Table 4. The score is adjusted to one if the volunteer fails to distinguish the modulated audio clip correctly, since it indicates that the volunteer thinks the modulated audio clip sounds better than the original audio.

Table 5 shows the results for four types of audio clips. We find that over 80% of the volunteers think there is no or just an occasional difference (but will not raise any alertness) between the modulated and the source audio

[9]MCD [15] measures the sound quality by comparing the distance between the target sound (the encoded audio) and the reference sound (the original sound) using $MCD = (10/ln(10)) \cdot \sqrt{2 \cdot \sum_{i=1}^{24} (mc_i^t - mc_i^e)^2}$ where mc_i^t and mc_i^e denote target and the estimated MCD, respectively.

Score	Explanation
1	There is no difference between the source and the modulated audio
2	The difference can be occasionally noticed, but it will not rise any alertness
3	The different can be frequently noticed, but it will not raise alertness
4	The difference can be occasionally noticed and will raise alertness
5	The difference can be frequently noticed and will raise alertness

Table 4. Criteria to assess the sound quality for scoring.

Score	Speech	News	Musical instrument	Vocal music
1	61.7%	73.3%	65.0%	88.3%
2	21.6%	16.7%	25.0%	8.3%
3	11.7%	8.3%	8.3%	3.4%
4	5.0%	1.7%	1.7%	0%
5	0%	0%	0%	0%
Better (ST/DH)	100%/0%	100%/0%	98%/2%	100%/0%

Table 5. Summary of the sound quality scoring results.

clip in all these four settings, demonstrating that in most cases, SoundSticker can successfully hide bits in audio clips without raising any human concerns. We also embed the same content in the same source audio clip using both Dolphin and SoundSticker. Each volunteer is then asked to find the modulated audio clip with better sound quality purely based on her acoustic perception. The result shows that almost all volunteers choose SoundSticker, indicating that SoundSticker does achieve better sound quality.

5.1.3 Impact of audio compression codecs. As discussed in §1, one of the primary advantages of SoundSticker is its resistance to audio codec. We experimentally validate this through MP3 and AAC compression. Specifically, we modulated an audio clip using both SoundSticker and Dolphin, and then conduct MP3 and AAC compression on the encoded audio clip, separately. The compression bitrate is 128 Kbps [55]. Figure 15(a-c) shows the spectrum of these two modulated audio clips before and after the MP3/AAC conversion. We observe the modulated part is entirely removed after MP3 and AAC conversion on Dolphin, which leads to zero packet reception rate (Figure 15(d)). In contrast, since SoundSticker modulates the in-band part of an audio clip for hidden bit encoding, the encoded information is well preserved after both MP3 and AAC compression. Accordingly, we achieve the same high packet reception rate. It should be noticed that MP3 and AAC compression would discard some low energy contents in the in-band frequency, it will not hurt the performance of SoundSticker due to its power-aware rate selection.

5.2 Microbenchmarks

We further conduct a set of microbenchmarks to fully understand the impacts of various factors on SoundSticker's performance. Due to the page limit, we focus on the smartphone receiver in this subsection.

Communication distance. In Figure 12, the receiver's positions already have different distances to the transmitter. In this experiment, we systematically investigate the factor of the communication distance by gradually increasing this distance from 0.5 to 8 m with line-of-sight paths, *e.g.*, a user can adjust her position inside the room to find a line-of-sight path to enjoy a higher goodput in the device-to-device communication mode. Figure 16(a) shows the average goodput at these distances. From the result, we can see that SoundSticker ("ST" in the figure) achieves consistently good goodput (around 310 bps) performance when the distance between the transmitter and the receiver is within 1 m. The goodput in these three settings all slightly drops as we place the receiver 3 m, 5 m, and then 8 m away, but SoundSticker still achieves above 220 bps goodput. On the other hand, Figure 16(a) further shows

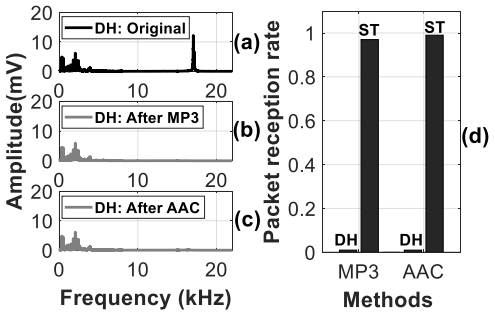

Fig. 15. Impact of MP3 and AAC conversions. (a) Spectrum of the audio clip encoded by Dolphin (DH) before conversion. The spectrum of the audio clip encoded by Dolphin after (b) MP3 and (c) AAC conversions. (d) Comparison of the packet reception rates.

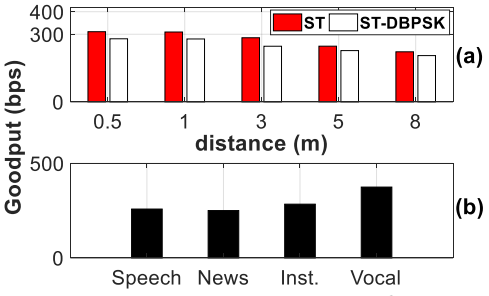

Fig. 16. Goodput of SoundSticker achieved (a) under different communication distances varied from 0.5 to 8 m with and without the rate selection; (b) by using four different types of audio contents.

the goodput of SoundSticker achieved by using the lowest modulation DBPSK only ("ST-DBPSK"). The result indicates that the rate selection design can increase the goodput by more than 10% than "ST-DBPSK".

Audio contents. We further evaluate the impact of different audio contents on SoundSticker. We encode hidden bits in four popular types of audio contents and play the modulated audio clips. Figure 16(b) shows the result. Due to the high energy across subcarriers, we observe that the instrumental and vocal musics can achieve higher goodput than the human speech and news report.

Immune to commercial speech-to-text translation systems. To further confirms the imperceptible of SoundSticker, we feed the modulated audio clips of human speeches and news reports to various neural network-based speech-to-text translation systems (G: Google speech-to-text system [71], F: iFLY [73], H: HappyScribe [72], and T: Transcribe(Wreally) [74]) to understand the impact of SoundSticker modulation on the audio content recognition accuracy. Figure 17(a) shows the character recognition accuracy. We can see all these four speech-to-text translation systems recognize each character of the source audio content on the modulated audio clip with a very high accuracy (>98%).

Impacts of receiver's movement, ambient noise and microphone diversity. Next, we study the impact of device movement on the packet decoding. We calculate BER when a user moves a receiver at different speeds: low (L-Speed) of 0.1 m/s, moderate (M-Speed) of 0.5 m/s and relatively high (H-Speed) of 1.5 m/s. We observe from Figure 17(b) that SoundSticker achieves consistently low BER (around 0.01 that is similar as the static setting) at a low moving speed. BER then grows to 0.03 at moderate moving speed and jumps to 0.15 at high moving speed setting due to Doppler frequency shift (DFS). One possible way to tackle this issue is to reserve some sub-carriers as anchors to detect (and then compensate) DFS. This brings a trade-off between goodput and robustness to mobility, which will be studied in the future.

We further examine the impact of ambient noise on the packet decoding. In particular, we add three kinds of background noise including a hum, music, and human speech to emulate different types of noisy environments. The speaker volume is 70 dBSPL and we then calculate the BER in these different settings. As shown in Figure 18(a), SoundSticker achieves similarly low BER in these three background noise settings. BER increases slightly from 0.01 to 0.03 as the noise level grows from 40 to 58 dBSPL (typical human conversation or TV show), indicating SoundSticker is robust to ambient noise in our daily lives. BER then jumps to 0.13 as the noise level grows up to 60 dBSPL.

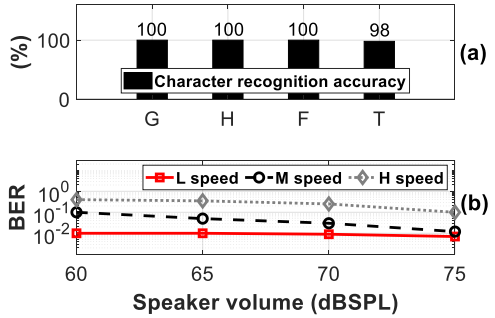

Fig. 17. (a) Character recognition accuracy on four audio-to-text translation systems; (b) Impact of receiver's moving speed (L: 0.1m/s; M: 0.5m/s; H: 1.5m/s) on BER vs. speaker sound volume (dBSPL).

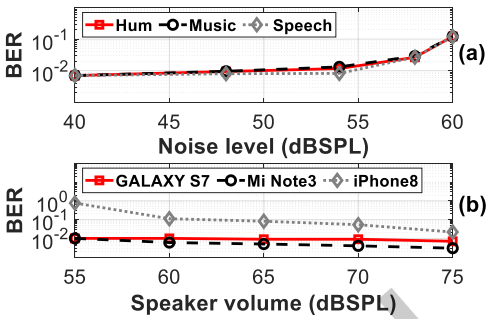

Fig. 18. (a) Impact of the background noise level (varying from 40 to 60 dBSPL) on SoundSticker's BER; (b) Device diversity (type of smartphones) on BER vs. speaker sound volume (dBSPL).

In the last experiment, we study the impact of device diversity on packet decoding. We receive the encoded audio clip using three different types of smartphone: a Samsung Galaxy S7, a Xiaomi Mi Note3, and an Apple iPhone 8. We then analyze their BERs and show the result in Figure 18(b). We observe there is a big BER gap between iPhone 8 and the remaining two devices, because the frequency response curve of iPhone 8 is worse than the other two devices. However, we also observe that this BER gap decreases as we gradually increase the speaker volume.

5.3 Case Study: Geofenced Connectivity

SoundSticker can enable geofenced connectivity through the in-band steganographic link as discussed in §2. In the future, SoundSticker could serve as a software-oriented package integrated into the wireless management toolkit at the access points (AP) or gateway side. It works only when room-area connectivity verification is enabled. In particular, after a device gets connected to a wireless network (*e.g.*, Wi-Fi, Zigbee, etc.), if the SoundSticker service is enabled, a "new connection" message (*e.g.*, containing the device ID, message type and message content) is generated and sent out by AP[10] over the acoustic channel (*e.g.*, background music). Only when the device is in the same room or space with the AP, it can receive the message and acknowledge it to the AP. After the verification is passed, the consequent traffic will get permitted for this device. The AP or gateway probes each verified device periodically if the re-verification is required.

Performance. In this case study, we examine the performance of such connectivity in the same office (Figure 12) with default setting. The connection is counted as successful as long as the bit error rate is lower than a threshold θ. We set θ to 10% and 6% for the smartphone and MCU, respectively. Such bit error rate can be easily corrected with redundancy. We repeat the experiment 100 times at each position and calculate the success rate of connectivity (SRC). The results are shown in Figure 19. When the receivers are placed outside the office at positions "6" (behind a wooden door) and "7" (behind the wall), the success rates are all zeros, for "ST-PHO", "DH-PHO" and "ST-MCU", which suggests that the acoustic channel can indeed avoid the connections cross room boundaries effectively. After the receivers enter the office at positions "1" and "2" (with a relatively clear line-of-sight path), "ST-PHO", "DH-PHO" and "ST-MCU" all achieve very high successful rates. The SRC of "DH-PHO" is slightly better than that of "ST-PHO". This is because the DQPSK modulation may cause slightly more packet decoding failures, while overall it can help improve throughput and goodput as shown in Figure 13. At NLOS positions "3"

[10]We note that smart devices and AP have been equipped or compatible with commercial speakers and microphone to enable SoundSticker.

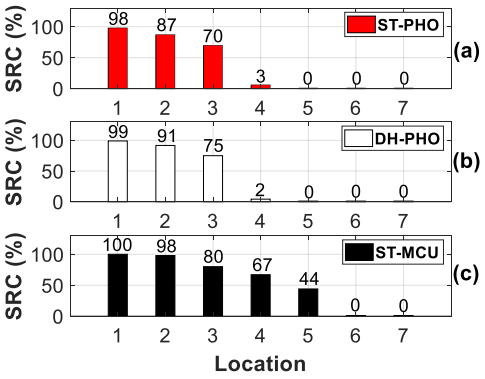

Fig. 19. Success Rate of Connectivity (SRC) cross locations: (a) ST and (b) DH with phone receivers; (c) ST with MCU.

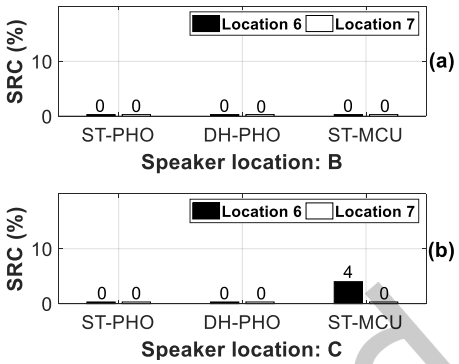

Fig. 20. Success Rate of Connectivity (SRC) by varying the speaker's position to (a) position "B" and (b) position "C".

to "5", the SRC is decreased, especially at positions "4" and "5", while the low-end MCU platform can achieve better performance than the phone receivers at these three positions, similar with Figure 13.

Varying the speaker's positions. In the experiments above, the speaker's default position "A" is relatively far away from the receiver's positions outside the room boundary, *e.g.*, positions "6" and "7". In Figure 20, we further examine the success rate of connectivity when the speaker is deployed closer to these behind-room-boundary positions. From the result, we can see that when the speaker is deployed at position "B" in Figure 12, the success rate remains 0% when receivers are both behind the wall and the door. Only when the speaker is placed closer to the door at position "C", the connectivity can be successful occasionally when the receiver is behind the door (position "6"), which can be further improved by lowering the transmission power when the speaker is in the vicinity of the door.

The experiments in Figures 19 and 20 indicate the efficacy to achieve the geofenced connectivity by using acoustic channels due to the sharp attenuation at the room boundaries of the acoustic signals. As unveiled in the above field study, the in-band design could outperform the out-of-hearing-band design in terms of the throughput, audio quality and the robustness. Therefore, the SoundSticker can enrich the existing geofenced connectivity channel family, which is prioritized to be adopted when the audio sound is available.

6 RELATED WORK

Over-the-air acoustic-steganographic transmissions. Early work [47, 48, 86] has explored acoustic channel for communication, *e.g.*, tone-based melody communication [47], near-field secure communication [52], fast device-to-device authentication [84], secure communication without key sharing [88]. However, most of these designs are human perceivable, so are not user-friendly and can also raise security concerns. Later, researchers study hidden bit encoding in audio clips [42, 43, 54]. To minimize the human perception, these hidden bits are typically encoded on the out-of-hearing band. But, as the embedded bits are loosely coupled with the original audio content in the frequency domain, the hidden bits can easily get lost due to common audio processing like MP3 conversion and compression. Furthermore, the limited sampling rate of ADC on commercial off-the-shelf devices (*e.g.*, smartphone) renders the maximum working bandwidth to 4kHz only (18-22kHz out-of-hearing-band), which severely limits the maximum achievable throughput of these designs.

The state-of-the-art works [49, 80, 81] divide the limited out-of-hearing band into multiple orthogonal subcarriers, with the goal of better utilizing the available out-of-hearing band. However, the throughput of these systems is

still limited due to the limited bandwidth. Dolphin [80] further combines the out-of-hearing band with part of the human perceivable band ($\geq$ 8 kHz bandwidth) to increase the throughput, using amplitude-shift keying and/or energy difference keying. However, since human ear is more sensitive to the signal's intensity change [45], the included audible frequencies cannot be excessive, which still limits its goodput. Moreover, the extra energy in these higher audible frequencies can further cause the perceptible interference. On the other hand, the out-of-hearing band cannot be received on the low-end platforms, due to the limited ADC sampling rate, which limits its usage on a wider set of receivers.

SoundSticker employs audio wave's phase to modulate the hidden bits. As the human ear is less imperceptible to the phase change of audio clips, some prior studies [14, 21, 22, 26, 50, 58] have explored phase based modulations to encode the hidden bits. For example, [58] applies DPSK modulation to adjacent subcarriers. [50] takes a single tone as the carrier and adopts DPSK to modulate bits. [14] introduces a synchronization frame to assists hidden bits decoding from the following frames. [26] encodes bits in Modulated Complex Lapped Transform (MCLT) domain. However, these works only conduct an initial exploration of this opportunity, and achieve less than 200 bps throughput over short links (2-4 m). In contrast, SoundSticker is a completely functional design consists of a full OFDM stack, PHY-layer rate selection and transmitter/receiver implementation. The prototype shows a working range within eight meters with much higher goodput. Two recent works [21, 22] harness the frequency masking techniques to embed hidden bits in the human audible band. However, they require a high pitch tone residing in one frequency band to hide the data encoded in nearby, low pitch frequency band, which appears only in certain types of audios (*e.g.*, music).

Digital watermarking and steganography bits. The SoundSticker design is also related to the audio watermarking [19, 67], which embeds hidden bits into audio files for copyright protection, while digital steganography [18] transmits secure messages through VoIP networks. These two techniques adopt similar modulation schemes that are broadly categorized into temporal methods [3, 23, 68], frequency methods [17, 27, 32, 34] and codec methods [29, 35, 53]. Relevant to SoundSticker, phase coding [19] in frequency domain commonly uses their absolute phase values [28, 38]. However, when audio files are played through a speaker, watermarking (embedded bits) cannot be decoded by receiver, because the audio's absolute phase values will change after over-the-air transmissions. The modulated phase information is thus destroyed. There are some initial attempts, like [5], to investigate this issue from the theoretical perspective using simulations. To our best knowledge, phase coding still cannot survive after the over-the-air transmission in practice [80]. Inspired by the observation explored in the information hiding domain that the human ear is less sensitive to the signal's phase changes, SoundSticker proposes novel ways to preserve the hidden bits after the over-the-air transmission, improves the data rate and minimizes the perceptible interference.

Microphone non-linearity. Recent studies [63, 64, 89] successfully realize a series of inaudible attacks by harnessing the non-linearities of the microphone's diaphragm and the receiver's power amplifier [63]. The speaker transmits tones at high (40–50 kHz) frequencies above the range of human hearing, while the receiver's non-linearities alias these high frequencies into the audible band so that receiver can detect them. LipRead uses this mechanism to silently inject commands to voice assistant products [89]. These designs can also enable an inaudible communication channel to microphone receivers and thus be used for information broadcast in shopping malls or serve as an alternate channel to Internet of Things devices to reduce wireless traffic. However, these systems, require dedicated speakers being able to transmit signals in the ultrasonic range, adding an additional hardware requirement, while SoundSticker is built with standard speakers that send signals only over audible frequencies.

Other related communication modality. Covert channels have also been studied through a dedicated constellation design [20] and Inertial measurement unit (IMU) sensors on smart phones [60]. Screen-camera based communications [36, 77–79, 87] can also deliver the hidden bits using videos [33, 37, 78]. Compared with the screen-camera

communications, acoustic based designs can avoid two major limitations: the line-of-sight requirement path and the rigid alignment between the sender and receiver [46, 82, 83].

7 CONCLUSION

This paper presents SoundSticker to send hidden bits over in-band acoustic steganographic links. Due to the fact that the audible band owns a substantial spectrum and the human ear is less sensitive to the audio phase changes than pitch and loudness changes, we modulate the phase of the original audio signal convey plenty of encoded bits imperceptibly. Moreover, the audible band is within a low frequency range, which can thus be received using even low-end MCUs to benefit a wider set of receivers. We propose novel and effective designs, and make significant engineering efforts to achieve both good goodput and undiminished audio quality. We develop the prototype of SoundSticker on two different platforms, which achieves promising performance and outperforms the state-of-the-art design. Finally, we release the audio samples of SoundSticker through [2].

ACKNOWLEDGEMENT

This work is sponsored by the project JCYJ20190808183203749 supported by the Science Technology and Innovation Committee of Shenzhen Municipality.

REFERENCES

[1] 2022. Inter-IC Sound (I2S) Bit Depth. website.
[2] 1released audio 2022. Audio clip samples generated by SoundSticker system. https://soundsticker.github.io/.
[3] Mohamed A Ahmed, Miss Laiha Mat Kiah, BB Zaidan, and AA Zaidan. 2010. A novel embedding method to increase capacity and robustness of low-bit encoding audio steganography technique using noise gate software logic algorithm. *Journal of Applied Sciences* (2010).
[4] Airpods 2022. Airpods. https://www.apple.com/.
[5] Michael Arnold, Xiao-Ming Chen, Peter Baum, Ulrich Gries, and Gwenael Doerr. 2014. A phase-based audio watermarking system robust to acoustic path propagation. *IEEE Transactions on Information Forensics and Security* (2014).
[6] Audio bit depth 2022. Audio bit depth. website.
[7] BBC News 2022. BBC News. https://www.bbc.co.uk/programmes.
[8] Mihir Bellare and Phillip Rogaway. 1996. The exact security of digital signatures-How to sign with RSA and Rabin. In *Proc. of EUROCRYPT*.
[9] bit depth 2022. Smart Phone Sample Rate and Bit Depth. website.
[10] Nicholas Carlini and David Wagner. 2018. Audio adversarial examples: Targeted attacks on speech-to-text. In *IEEE Deep Learning and Security workshop*.
[11] Carole King - I Feel the Earth Move 2022. Carole King - I Feel the Earth Move. https://www.amazon.com/I-Feel-the-Earth-Move.
[12] Nuno Borges Carvalho, Alessandro Cidronali, and Roberto Gómez-García. 2014. *White space communication technologies*. Cambridge University Press.
[13] Tao Chen, Longfei Shangguan, Zhenjiang Li, and Kyle Jamieson. 2020. Metamorph: Injecting Inaudible Commands into Over-the-air Voice Controlled Systems. In *Proceedings of NDSS*.
[14] Kiho Cho, Jae Choi, and Nam Soo Kim. 2015. An acoustic data transmission system based on audio data hiding: method and performance evaluation. *EURASIP Journal on Audio, Speech, and Music Processing* (2015).
[15] Srinivas Desai, E Veera Raghavendra, B Yegnanarayana, Alan W Black, and Kishore Prahallad. 2009. Voice conversion using artificial neural networks. In *Proceedings of IEEE ICASSP*.
[16] Digital Audio Basics 2022. Digital Audio Basics: Sample Rate and Bit Depth. website.
[17] Fatiha Djebbar, Beghdad Ayad, Karim Abed-Meraim, and Habib Hamam. 2013. Unified phase and magnitude speech spectra data hiding algorithm. *Security and Communication Networks* (2013).
[18] Fatiha Djebbar, Beghdad Ayad, Karim Abed Meraim, and Habib Hamam. 2012. Comparative study of digital audio steganography techniques. *EURASIP Journal on Audio, Speech, and Music Processing* (2012).
[19] Xiaoxiao Dong, Mark F Bocko, and Zeljko Ignjatovic. 2004. Data hiding via phase manipulation of audio signals. In *Proceedings of IEEE ICASSP*.
[20] Aveek Dutta, Dola Saha, Dirk Grunwald, and Douglas Sicker. 2012. Secret agent radio: Covert communication through dirty constellations. In *International Workshop on Information Hiding*.

[21] Manuel Eichelberger, Simon Tanner, Gabriel Voirol, and Roger Wattenhofer. 2019. Imperceptible Audio Communication. In *Proceedings of IEEE ICASSP*.
[22] Manuel Eichelberger, Simon Tanner, Gabriel Voirol, and Roger Wattenhofer. 2019. Receiving Data Hidden in Music. In *Proceedings of ACM HotMobile*.
[23] Yousof Erfani and Shadi Siahpoush. 2009. Robust audio watermarking using improved TS echo hiding. *Digital Signal Processing* (2009).
[24] Fitbit 2022. Fitbit. https://www.fitbit.com/.
[25] Harvey Fletcher. 1953. *Speech and hearing in communication*. D. van Nostrand.
[26] Roman Frigg, Giorgio Corbellini, Stefan Mangold, and Thomas R Gross. 2014. Acoustic data transmission to collaborating smartphones—An experimental study. In *Proceedings of IEEE WONS*.
[27] Litao Gang, Ali N Akansu, and Mahalingam Ramkumar. 2001. MP3 resistant oblivious steganography. In *Proceedings of IEEE ICASSP*.
[28] Jose Juan Garcia-Hernandez, Ramon Parra-Michel, Claudia Feregrino-Uribe, and Rene Cumplido. 2013. High payload data-hiding in audio signals based on a modified OFDM approach. *Expert Systems with Applications* (2013).
[29] Bernd Geiser and Peter Vary. 2008. High rate data hiding in ACELP speech codecs. In *Proceedings of IEEE ICASSP*.
[30] Andrea Goldsmith. 2005. *Wireless communications*. Cambridge university press.
[31] goodput 2010. Goodput. https://en.wikipedia.org/wiki/Goodput.
[32] Kaliappan Gopalan and Stanley Wenndt. 2004. Audio steganography for covert data transmission by imperceptible tone insertion. In *Proceedings of CSA*.
[33] Tian Hao, Ruogu Zhou, and Guoliang Xing. 2012. COBRA: color barcode streaming for smartphone systems. In *Proceedings of ACM MobiSys*.
[34] S Hernandez-Garay, Ruben Vazquez-Medina, L Nino de Rivera, and V Ponomaryov. 2008. Steganographic communication channel using audio signals. In *Proceedings of IEEE MMET*.
[35] Konrad Hofbauer and Gernot Kubin. 2006. High-rate data embedding in unvoiced speech. In *Proceedings of INTERSPEECH*.
[36] Wenjun Hu, Hao Gu, and Qifan Pu. 2013. Lightsync: Unsynchronized visual communication over screen-camera links. In *Proceedings of ACM MobiCom*.
[37] Wenjun Hu, Jingshu Mao, Zihui Huang, Yiqing Xue, Junfeng She, Kaigui Bian, and Guobin Shen. 2014. Strata: layered coding for scalable visual communication. In *Proceedings of ACM MobiCom*.
[38] Guang Hua, Jiwu Huang, Yun Q Shi, Jonathan Goh, and Vrizlynn LL Thing. 2016. Twenty years of digital audio watermarking - a comprehensive review. *Elsevier Signal Processing* (2016).
[39] humanAuditorySystem 2010. Video of how the ear works. https://www.hearinglink.org/your-hearing/.
[40] Peter A Iannucci, Ravi Netravali, Ameesh K Goyal, and Hari Balakrishnan. 2015. Room-area networks. In *Proceedings of ACM HotNet*.
[41] Zeyu Jin, Gautham J Mysore, Stephen Diverdi, Jingwan Lu, and Adam Finkelstein. 2017. VoCo: text-based insertion and replacement in audio narration. *ACM Transactions on Graphics* (2017).
[42] Soonwon Ka, Tae Hyun Kim, Jae Yeol Ha, Sun Hong Lim, Su Cheol Shin, Jun Won Choi, Chulyoung Kwak, and Sunghyun Choi. 2016. Near-ultrasound communication for TV's 2nd screen services. In *Proceedings of ACM MobiCom*.
[43] Hyewon Lee, Tae Hyun Kim, Jun Won Choi, and Sunghyun Choi. 2015. Chirp signal-based aerial acoustic communication for smart devices. In *Proceedings of IEEE INFOCOM*.
[44] Roar Lien, Tim Grembowski, and Kris Gaj. 2004. A 1 Gbit/s partially unrolled architecture of hash functions SHA-1 and SHA-512. In *Proceedings of CT-RSA*.
[45] Yiqing Lin and Waleed H Abdulla. 2015. Principles of Psychoacoustics. In *Audio Watermark*. Springer.
[46] Manni Liu, Linsong Cheng, Kun Qian, Jiliang Wang, Jin Wang, and Yunhao Liu. 2020. Indoor acoustic localization: A survey. *Human-centric Computing and Information Sciences* (2020).
[47] Anil Madhavapeddy, Richard Sharp, David Scott, and Alastair Tse. 2005. Audio networking: the forgotten wireless technology. *IEEE Pervasive Computing* (2005).
[48] Hosei Matsuoka, Yusuke Nakashima, and Takeshi Yoshimura. 2006. Acoustic communication system using mobile terminal microphones. *NTT DoCoMo Tech. J* (2006).
[49] Hosei Matsuoka, Yusuke Nakashima, Takeshi Yoshimura, and Toshiro Kawahara. 2008. Acoustic OFDM: Embedding high bit-rate data in audio. In *Proceedings of MMM*.
[50] Keiichi Mizutani, Naoto Wakatsuki, and Koichi Mizutani. 2007. Acoustic communication in air using differential biphase shift keying with influence of impulse response and background noise. *Japanese Journal of Applied Physics* (2007).
[51] My Chemical Romance - The World Is Ugly 2022. My Chemical Romance - The World Is Ugly. https://www.amazon.com/The-World-Is-Ugly.
[52] Rajalakshmi Nandakumar, Krishna Kant Chintalapudi, Venkat Padmanabhan, and Ramarathnam Venkatesan. 2013. Dhwani: secure peer-to-peer acoustic NFC. In *Proceedings of ACM SIGCOMM*.
[53] Akira Nishimura. 2008. Data hiding for audio signals that are robust with respect to air transmission and a speech codec. In *Proceedings of IEEE IIH-MSP*.

[54] Aditya Shekhar Nittala, Xing-Dong Yang, Scott Bateman, Ehud Sharlin, and Saul Greenberg. 2015. PhoneEar: interactions for mobile devices that hear high-frequency sound-encoded data. In *Proceedings of ACM SIGCHI*.
[55] online-convert 2022. audio online convert. https://audio.online-convert.com/.
[56] Kuldip K Paliwal and Leigh D Alsteris. 2005. On the usefulness of STFT phase spectrum in human listening tests. *Speech Communication* (2005).
[57] AR Palmer and IJ Russell. 1986. Phase-locking in the cochlear nerve of the guinea-pig and its relation to the receptor potential of inner hair-cells. *Hearing research* (1986).
[58] Nikhil Parab, Mark Nathan, and KT Talele. 2011. Audio steganography using differential phase encoding. In *Technology Systems and Management*. Springer.
[59] Dale Purves, George J Augustine, David Fitzpatrick, WC Hall, AS LaMantia, JO McNamara, and L White. 2011. Neuroscience, 5th edition. *Sinauer Associates* (2011).
[60] Wen Qi, Wanfu Ding, Xinyu Wang, Yonghang Jiang, Yichen Xu, Jianping Wang, and Kejie Lu. 2018. Construction and mitigation of user-behavior-based covert channels on smartphones. *IEEE Transactions on Mobile Computing* (2018).
[61] Kun Qian, Yumeng Lu, Zheng Yang, Kai Zhang, Kehong Huang, Xinjun Cai, and Yunhao Liu. 2021. {AIRCODE}: Hidden Screen-Camera Communication on an Invisible and Inaudible Dual Channel. In *Proceedings of USENIX NSDI*.
[62] Queen - The Show Must Go On 2022. Queen - The Show Must Go On. https://www.amazon.com/Show-Must-Go-Remastered.
[63] Nirupam Roy, Haitham Hassanieh, and Romit Roy Choudhury. 2017. Backdoor: Making microphones hear inaudible sounds. In *Proceedings of ACM MobiSys*.
[64] Nirupam Roy, Sheng Shen, Haitham Hassanieh, and Romit Roy Choudhury. 2018. Inaudible Voice Commands: The Long-Range Attack and Defense. In *Proceedings of USENIX NSDI*.
[65] sampling rate of adc 2010. Fast sampling ADC. https://forum.arduino.cc/t/fast-sampling-adc/64877.
[66] Timothy M Schmidl and Donald C Cox. 1997. Robust frequency and timing synchronization for OFDM. *IEEE transactions on communications* (1997).
[67] Juergen Seitz. 2005. *Digital watermarking for digital media*. IGI Global.
[68] Sajad Shirali-Shahreza and Mohammad Shirali-Shahreza. 2008. Steganography in silence intervals of speech. In *Proceedings of IEEE IIH-MSP*.
[69] Smart Speaker Market 2022. Smart Speaker Market: USD 34.24 billion in 2028. website.
[70] soundmeter 2010. Abc Apps. Sound Meter. https://play.google.com/store/apps/.
[71] Speech Recognition Tool 2022. Google speech-to-text. https://cloud.google.com/speech-to-text/.
[72] Speech Recognition Tool 2022. Happy Scribe. https://www.happyscribe.co/.
[73] Speech Recognition Tool 2022. iflyrec. https://www.iflyrec.com/.
[74] Speech Recognition Tool 2022. Transcribe (wreally). https://transcribe.wreally.com/.
[75] David Tse and Pramod Viswanath. 2005. *Fundamentals of wireless communication*. Cambridge university press.
[76] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).
[77] Anran Wang, Zhuoran Li, Chunyi Peng, Guobin Shen, Gan Fang, and Bing Zeng. 2015. Inframe++: Achieve simultaneous screen-human viewing and hidden screen-camera communication. In *Proceedings of ACM MobiSys*.
[78] Anran Wang, Shuai Ma, Chunming Hu, Jinpeng Huai, Chunyi Peng, and Guobin Shen. 2014. Enhancing reliability to boost the throughput over screen-camera links. In *Proceedings of ACM MobiCom*.
[79] Anran Wang, Chunyi Peng, Ouyang Zhang, Guobin Shen, and Bing Zeng. 2014. Inframe: Multiflexing full-frame visible communication channel for humans and devices. In *Proceedings of ACM HotNets*.
[80] Qian Wang, Kui Ren, Man Zhou, Tao Lei, Dimitrios Koutsonikolas, and Lu Su. 2016. Messages behind the sound: real-time hidden acoustic signal capture with smartphones. In *Proceedings of ACM MobiCom*.
[81] Shuai Wang. 2011. Embedding data in an audio signal, using acoustic OFDM. https://diva-portal.org/.
[82] Weiguo Wang, Jinming Li, Yuan He, and Yunhao Liu. 2022. Localizing Multiple Acoustic Sources With a Single Microphone Array. *IEEE Transactions on Mobile Computing* (2022).
[83] Yanwen Wang, Jiaxing Shen, and Yuanqing Zheng. 2020. Push the limit of acoustic gesture recognition. *IEEE Transactions on Mobile Computing* (2020).
[84] Pengjin Xie, Jingchao Feng, Zhichao Cao, and Jiliang Wang. 2017. GeneWave: Fast authentication and key agreement on commodity mobile devices. In *Proceedings of IEEE ICNP*.
[85] Young Mountain - This Will Destroy You 2022. Young Mountain - This Will Destroy You. https://www.amazon.com/Young-Mountain-This-Will-Destroy.
[86] Hwan Sik Yun, Kiho Cho, and Nam Soo Kim. 2010. Acoustic data transmission based on modulated complex lapped transform. *IEEE Signal Processing Letters* (2010).

[87] Tong Zhan, Wenzhong Li, Xu Chen, and Sanglu Lu. 2018. Capturing the shifting shapes: Enabling efficient screen-camera communication with a pattern-based dynamic barcode. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (2018).
[88] Bingsheng Zhang, Qin Zhan, Si Chen, Muyuan Li, Kui Ren, Cong Wang, and Di Ma. 2014. PriWhisper: Enabling Keyless Secure Acoustic Communication for Smartphones. *IEEE internet of things journal* (2014).
[89] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2017. DolphinAttack: Inaudible voice commands. In *Proceedings of ACM CCS*.
[90] Eberhard Zwicker. 1961. Subdivision of the audible frequency range into critical bands (Frequenzgruppen). *The Journal of the Acoustical Society of America* (1961).
[91] Eberhard Zwicker and Hugo Fastl. 2013. *Psychoacoustics: Facts and models*. Springer Science & Business Media.