Gyrophone: Recognizing Speech From Gyroscope Signals

Yan Michalevsky  Dan Boneh
Computer Science Department
Stanford University

Gabi Nakibly
National Research & Simulation Center
Rafael Ltd.

Abstract
We show that the MEMS gyroscopes found on modern smart phones are sufficiently sensitive to measure acoustic signals in the vicinity of the phone. The resulting signals contain only very low-frequency information (<200Hz). Nevertheless we show, using signal processing and machine learning, that this information is sufficient to identify speaker information and even parse speech. Since iOS and Android require no special permissions to access the gyro, our results show that apps and active web content that cannot access the microphone can nevertheless eavesdrop on speech in the vicinity of the phone.

1 Introduction
Modern smartphones and mobile devices have many sensors that enable rich user experience. Being generally put to good use, they can sometimes unintentionally expose information the user does not want to share. While the privacy risks associated with some sensors like a microphone (eavesdropping), camera or GPS (tracking) are obvious and well understood, some of the risks remained under the radar for users and application developers. In particular, access to motion sensors such as gyroscope and accelerometer is unmitigated by mobile operating systems. Namely, every application installed on a phone and every web page browsed over it can measure and record these sensors without the user being aware of it.

Recently, a few research works pointed out unintended information leaks using motion sensors. In Ref. [34] the authors suggest a method for user identification from gait patterns obtained from a mobile device’s accelerometers. The feasibility of keystroke inference from nearby keyboards using accelerometers has been shown in [35]. In [21], the authors demonstrate the possibility of keystroke inference on a mobile device using accelerometers and mention the potential of using gyroscope measurements as well, while another study [19] points to the benefits of exploiting the gyroscope.

All of the above work focused on exploitation of motion events obtained from the sensors, utilizing the expected kinetic response of accelerometers and gyroscopes. In this paper we reveal a new way to extract information from gyroscope measurements. We show that gyroscopes are sufficiently sensitive to measure acoustic vibrations. This leads to the possibility of recovering speech from gyroscope readings, namely using the gyroscope as a crude microphone. We show that the sampling rate of the gyroscope is up to 200 Hz which covers some of the audible range. This raises the possibility of eavesdropping on speech in the vicinity of a phone without access to the real microphone.

As the sampling rate of the gyroscope is limited, one cannot fully reconstruct a comprehensible speech from measurements of a single gyroscope. Therefore, we resort to automatic speech recognition. We extract features from the gyroscope measurements using various signal processing methods and train machine learning algorithms for recognition. We achieve about 50% success rate for speaker identification from a set of 10 speakers. We also show that while limiting ourselves to a small vocabulary consisting solely of digit pronunciations (“one”, “two”, “three”, ...) and achieve speech recognition success rate of 65% for the speaker dependent case and up to 26% recognition rate for the speaker independent case. This capability allows an attacker to substantially leak information about numbers spoken over or next to a phone (i.e. credit card numbers, social security numbers and the like).

We also consider the setting of a conference room where two or more people are carrying smartphones or tablets. This setting allows an attacker to gain simultaneous measurements of speech from several gyroscopes. We show that by combining the signals from two or more phones we can increase the effective sampling rate of the acoustic signal while achieving better speech recognition rates. In our experiments we achieved 77% successful recognition rate in the speaker dependent case based on the digits vocabulary.

The paper structure is as follows: in Section 2 we provide a brief description of how a MEMS gyroscope works and present initial investigation of its properties as a microphone. In Section 3 we discuss speech analysis and describe our algorithms for speaker and speech recognition. In Section 4 we suggest a method for audio signal recovery using samples from multiple devices. In Section 5 we discuss more directions for exploitation of gyroscopes’ acoustic sensitivity. Finally, in Section 6 we discuss mitigation measures of this unexpected threat. In
particular, we argue that restricting the sampling rate is an effective and backwards compatible solution.

2 Gyroscope as a microphone

In this section we explain how MEMS gyroscopes operate and present an initial investigation of their susceptibility to acoustic signals.

2.1 How does a MEMS gyroscope work?

Standard-size (non-MEMS) gyroscopes are usually composed of a spinning wheel on an axle that is free to assume any orientation. Based on the principles of angular momentum the wheel resists to changes in orientation, thereby allowing to measure those changes. Nonetheless, all MEMS gyro take advantage of a different physical phenomenon – the Coriolis force. It is a fictitious force (d’Alembert force) that appears to act on an object while viewing it from a rotating reference frame (much like the centrifugal force). The Coriolis force acts in a direction perpendicular to the rotation axis of the reference frame and to the velocity of the viewed object. The Coriolis force is calculated by \( F = 2m\vec{v} \times \vec{\omega} \) where \( m \) and \( \vec{v} \) denote the object’s mass and velocity, respectively, and \( \vec{\omega} \) denotes the angular rate of the reference frame.

Generally speaking, MEMS gyro measure their angular rate (\( \vec{\omega} \)) by sensing the magnitude of the Coriolis force acting on a moving proof mass within the gyro. Usually the moving proof mass constantly vibrates within the gyro. Its vibration frequency is also called the resonance frequency of the gyro. The Coriolis force is sensed by measuring its resulting vibration, which is orthogonal to the primary vibration movement. Some gyroscope designs use a single mass to measure the angular rate of different axes, while others use multiple masses. Such a general design is commonly called vibrating structure gyroscope.

There are two primary vendors of MEMS gyroscopes for mobile devices: STMicroelectronics [15] and InvenSense [7]. According to a recent survey [18] STMicroelectronics dominates with 80% market share. Teardown analyses show that this vendor’s gyros can be found in Apple’s iPhones and iPads [17, 8] and also in the latest generations of Samsung’s Galaxy-line phones [5, 6]. The second vendor, InvenSense, has the remaining 20% market share [18]. InvenSense gyros can be found in Google’s latest generations of Nexus-line phones and tablets [14, 13] as well as in Galaxy-line tablets [4, 3]. These two vendors’ gyroscopes have different mechanical designs, but are both noticeably influenced by acoustic noise.

2.1.1 STMicroelectronics

The design of STMicroelectronics 3-axis gyros is based on a single driving (vibrating) mass (shown in Figure 1). The driving mass consists of 4 parts \( M_1, M_2, M_3 \) and \( M_4 \) (Figure 1(b)). They move inward and outward simultaneously at a certain frequency\(^1\) in the horizontal plane. As shown in Figure 1(b), when an angular rate is applied on the Z-axis, due to the Coriolis effect, \( M_2 \) and \( M_4 \) will move in the same horizontal plane in opposite directions as shown by the red and yellow arrows. When an angular rate is applied on the X-axis, then \( M_1 \) and \( M_3 \) will move in opposite directions up and down out of the plane due to the Coriolis effect. When an angular rate is applied to the Y-axis, then \( M_2 \) and \( M_4 \) will move in opposite directions up and down out of the plane. The movement of the driving mass causes a capacitance change relative to stationary plates surrounding it. This change is sensed and translated into the measurement signal.

2.1.2 InvenSense

InvenSense’s gyro design is based on the three separate driving (vibrating) masses\(^2\); each senses angular rate at a different axis (shown in Figure 2(a)). Each mass is a coupled dual-mass that move in opposite directions. The masses that sense the X and Y axes are driven out-of-plane (see Figure 2(b)), while the Z-axis mass is driven in-plane. As in the STMicroelectronics design the movement due to the Coriolis force is measures by capacitance changes.

2.2 Acoustic Effects

It is a well known fact in the MEMS community that MEMS gyros are susceptible to acoustic noise which degrades their accuracy [22, 24, 25]. An acoustic signal affects the gyroscope measurement by making the driving mass vibrate in the sensing axis (the axis which senses the Coriolis force). The acoustic signal can be transferred to the driving mass in one of two ways. First, it may induce mechanical vibrations to the gyro package. Additionally, the acoustic signal can travel through the gyroscope packaging and directly affect the driving mass in case it is suspended in air. The acoustic noise has the most substantial effect when it is near the resonance frequency of the vibrating mass. Such effects in some cases can render the gyro’s measurements useless or even saturated. Therefore to reduce the noise effects vendors manufacture gyros with a high resonance frequency (above

\(^1\)It is indicated in [1] that STMicroelectronics uses a driving frequency of over 20 KHz.
\(^2\)According to [43] the driving frequency of the masses is between 25 KHz and 30 KHz.
20 KHz) where acoustic signals are minimal. Nonetheless, in our experiments we found that acoustic signals at frequencies much lower than the resonance frequency still have a measurable effect on a gyro’s measurements, allowing one to reconstruct the acoustic signal.

2.3 Characteristics of a gyro as a microphone

Due to the gyro’s acoustic susceptibility one can treat gyroscope readings as if they were audio samples coming from a microphone. Note that the frequency of an audible signal is higher than 20 Hz, while in common cases the frequency of change of mobile device’s angular velocity is lower than 20 cycles per second. Therefore, one can high-pass-filter the gyroscope readings in order to retain only the effects of an audio signal even if the mobile device is moving about. Nonetheless, it should be noted that this filtering may result in some loss of acoustic information since some aliased frequencies may be filtered out (see Section 2.3.2). In the following we explore the gyroscope characteristics from a standpoint of an acoustic sensor, i.e. a microphone. In this section we exemplify these characteristics by experimenting with Galaxy S III which has an STMicroelectronics gyro [6].

2.3.1 Sampling

**Sampling resolution** is measured by the number of bits per sample. More bits allow us to sample the signal more accurately at any given time. All the latest generations of gyroscopes have a sample resolution of 16 bits [9, 12]. This is comparable to a microphone’s sampling resolution used in most audio applications.

**Sampling frequency** is the rate at which a signal is sampled. According to the Nyquist sampling theorem a sampling frequency $f$ enables us to reconstruct signals at frequencies of up to $f/2$. Hence, a higher sampling frequency allows us to more accurately reconstruct the audio signal. In most mobile devices and operating systems an application is able to sample the output of a microphone at up to 44.1 KHz. A telephone system (POTS) samples an audio signal at 8000 Hz. However, STMicroelectronics’ gyroscope hardware supports sampling frequencies of up to 800 Hz [9], while InvenSense gyros’ hardware support sampling frequency up to 8000 Hz [12]. Moreover, all mobile operating systems bound the sampling frequency even further – up to 200 Hz – to limit power consumption. On top of that, it appears that some browser toolkits limit the sampling frequency even further. Table 1 summarizes the results of our experi-
might be masked by the noise at low frequencies, i.e., under 20 Hz.

phenomenon where for a sinusoid of frequency \( f \) sampled with frequency \( f_s \), the resulting samples are indistinguishable from those of another sinusoid of frequency \( |f - N \cdot f_s| \), for any integer \( N \). The values corresponding to \( N \neq 0 \) are called images or aliases of frequency \( f \). An undesirable phenomenon in general, here aliasing allows us to sense audio signals having frequencies which are higher than 100 Hz, thereby extracting more information from the gyroscope readings. This is illustrated in Figure 3.

Using the gyro, we recorded a single 280 Hz tone. Figure 3(a) depicts the recorded signal in the frequency domain (x-axis) over time (y-axis). A lighter shade in the spectrogram indicates a stronger signal at the corresponding frequency and time values. It can be clearly seen that there is a strong signal sensed at frequency 80 Hz starting around 1.5 sec. This is an alias of the 280 Hz-tone. Note that the aliased tone is indistinguishable from an actual tone at the aliased frequency. Figure 3(b) depicts a recording of multiple short tones between 130 Hz and 200 Hz. Again, a strong signal can be seen at the aliased frequencies corresponding to 130 - 170 Hz. We also observe some weaker aliases that do not correspond to the base frequencies of the recorded tones, and perhaps correspond to their harmonics. Figure 3(c) depicts the recording of a chirp in the range of 420 - 480 Hz. The aliased chirp is detectable in the range of 20 - 80 Hz; however it is a rather weak signal.

### 2.3.2 Aliasing

As noted above, the sampling frequency of a gyro is uniform and can be at most 200 Hz. This allows us to directly sense audio signals of up to 100 Hz. Aliasing is a phenomenon where for a sinusoid of frequency \( f \), sampled with frequency \( f_s \), the resulting samples are indistinguishable from those of another sinusoid of frequency \( |f - N \cdot f_s| \), for any integer \( N \). The values corresponding to \( N \neq 0 \) are called images or aliases of frequency \( f \). An undesirable phenomenon in general, here aliasing allows us to sense audio signals having frequencies which are higher than 100 Hz, thereby extracting more information from the gyroscope readings. This is illustrated in Figure 3.

Using the gyro, we recorded a single 280 Hz tone. Figure 3(a) depicts the recorded signal in the frequency domain (x-axis) over time (y-axis). A lighter shade in the spectrogram indicates a stronger signal at the corresponding frequency and time values. It can be clearly seen that there is a strong signal sensed at frequency 80 Hz starting around 1.5 sec. This is an alias of the 280 Hz-tone. Note that the aliased tone is indistinguishable from an actual tone at the aliased frequency. Figure 3(b) depicts a recording of multiple short tones between 130 Hz and 200 Hz. Again, a strong signal can be seen at the aliased frequencies corresponding to 130 - 170 Hz. We also observe some weaker aliases that do not correspond to the base frequencies of the recorded tones, and perhaps correspond to their harmonics. Figure 3(c) depicts the recording of a chirp in the range of 420 - 480 Hz. The aliased chirp is detectable in the range of 20 - 80 Hz; however it is a rather weak signal.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Sampling Freq. [Hz]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>200</td>
</tr>
<tr>
<td>Chrome</td>
<td>25</td>
</tr>
<tr>
<td>Firefox</td>
<td>200</td>
</tr>
<tr>
<td>Opera</td>
<td>20</td>
</tr>
<tr>
<td>iOS 7</td>
<td>100 [2]</td>
</tr>
<tr>
<td>Safari</td>
<td>20</td>
</tr>
<tr>
<td>Chrome</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 1: Maximum sampling frequencies on different platforms

ments measuring the maximum sampling frequencies allowed in the latest versions of Android and iOS both for application and for web application running on common browsers. The code we used to sample the gyro via a web page can be found in Appendix B. The results indicate that a Gecko based browser does not limit the sampling frequency beyond the limit imposed by the operating system, while WebKit and Blink based browsers does impose stricter limits on it.

### 2.3.3 Self noise

The self noise characteristic of a microphone indicates what is the quietest sound, in decibels, a microphone can pick up, i.e. the sound that is just over its self noise. To measure the gyroscope’s self noise we played 80 Hz tones for 10 seconds at different volumes while measuring it using a decibel meter. Each tone was recorded by the Galaxy S III gyroscope. While analyzing the gyro recordings we realized that the gyro readings have a noticeable increase in amplitude when playing tones with volume of 75 dB or higher which is comparable to the volume of a loud conversation. Moreover, a FFT plot of the gyro recordings gives a noticeable peak at the tone’s frequency when playing tone with a volume as low as 57 dB which is below the sound level of a normal conversation. These findings indicate that a gyro can pick up audio signals which are lower than 100 HZ during most conversations made over or next to the phone. To test the self noise of the gyro for aliased tones we played 150 Hz and 250 Hz tones. The lowest level of sound the gyro picked up was 67 dB and 77 dB, respectively. These are much higher values that are comparable to a loud conversation.

### 2.3.4 Directionality

We now measure how the angle at which the audio signal hits the phone affects the gyro. For this experiment we played an 80 Hz tone at the same volume three times. The tone was recorded at each time by the Galaxy S III gyro while the phone rested at a different orientation. For each recording we calculated the FFT magnitude at 80 Hz. Table 2 summarizes the results.

It is obvious from the table that for each direction the audio hit the gyro, there is at least one axis whose readings are dominant by an order of magnitude compared to the rest. This can be explained by STMicroelectronics gyroscope design as depicted in Figure 1. When the signal travels in parallel to the phone’s x or y axes, the sound pressure vibrates mostly masses laid along the respective axis, i.e. \( M_2 \) and \( M_4 \) for x axis and \( M_1 \) and \( M_3 \).
Figure 3: Example of aliasing on a mobile device. Nexus 4 (a,c) and Galaxy SII (b).

<table>
<thead>
<tr>
<th>Tone direction:</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recording direction:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amplitude:</td>
<td>0.002</td>
<td>0.012</td>
<td>0.0024</td>
<td>0.01</td>
<td>0.007</td>
<td>0.004</td>
<td>0.007</td>
<td>0.0036</td>
<td>0.0003</td>
</tr>
</tbody>
</table>

Table 2: Sensed amplitude for every direction of a tone played at different orientations relative to the phone. For each orientation the dominant sensed directions are emphasized.

Figure 4: Coordinate system of Android and iOS.

for the y axis; therefore, the gyro primarily senses a rotation at the y or x axes, respectively (see Section 2.1.1). When the signal travels in parallel to the phone’s z axis then the sound pressure vibrates all the 4 masses up and down, hence the gyro primarily senses a rotation at both x and y axes.

These findings indicate that the gyro is an omnidirectional audio sensor allowing it to pick up audio signal from every direction.

3 Speech analysis based on a single gyroscope

In this section we show that the acoustic signal measured by a single gyroscope is sufficient to extract information about the speech signal, such as speaker characteristics and identity, and even recognize the spoken words or phrases. We do so by leveraging the fact that aliasing causes information leaks from higher frequency bands into the sub-Nyquist range.

Since the fundamentals of human voices are roughly in the range of 80 – 1100 Hz [20], we can capture a large fraction of the interesting frequencies, considering the results we observe in 2.3.2. Although we do not delve into comparing performance for different types of speakers, one might expect that given a stronger gyroscope response for low frequencies, typical adult male speech (Bass, Baritone, Tenor) could be better analyzed than typical female or child speech (Alto, Mezzo-Soprano, Soprano) 5, however our tests show that this is not necessarily the case.

The signal recording, as captured by the gyroscope, is not comprehensible to a human ear, and exhibits a mixture of low frequencies and aliases of frequencies beyond the Nyquist sampling frequency (which is 1/2 the sampling rate of the Gyroscope, i.e. 100 Hz). While the signal recorded by a single device does not resemble speech, it is possible to train a machine to transcribe the signal with significant success.

Speech recognition tasks can be classified into several types according to the setup. Speech recognition can handle fluent speech or isolated words (or phrases); operate on a closed set of words (finite dictionary) or an open set6; It can also be speaker dependent (in which case the recognizer is trained per speaker) or speaker in-

5For more information about vocal range see http://www.wikipedia.org/wiki/Vocal_range
6For example by identifying phonemes and combining them to words.
dependent (in which case the recognizer is expected to identify phrases pronounced by different speakers and possibly ones that were not encountered in the training set). Additionally, speech analysis may be also used to identify the speaker.

We focused on speaker identification (including gender identification of the speaker) and isolated words recognition while attempting both speaker independent and speaker dependent recognition. Although we do not demonstrate fluent speech transcription, we suggest that successful isolated words recognition could be fairly easily transformed into a transcription algorithm by incorporating word slicing and HMM [40]. We did not aim to implement a state-of-the-art speech recognition algorithm, nor to thoroughly evaluate or do a comparative analysis of the classification tests. Instead, we tried to indicate the potential risk by showing significant success rates of our speech analysis algorithms compared to randomly guessing. This section describes speech analysis techniques that are common in practice, our approach, and suggestions for further improvements upon it.

3.1 Speech processing: features and algorithms

3.1.1 Features

It is common for various feature extraction methods to view speech as a process that is stationary for short time windows. Therefore speech processing usually involves segmentation of the signal to short (10 – 30 ms) overlapping or non-overlapping windows and operation on them. This results in a time-series of features that characterize the time-dependent behavior of the signal. If we are interested in time-independent properties we shall use spectral features or the statistics of those time-series (such as mean, variance, skewness and kurtosis).

Mel-Frequency Cepstral Coefficients (MFCC) are widely used features in audio and speech processing applications. The Mel-scale basically compensates for the non-linear frequency response of the human ear. The Cepstrum transformation is an attempt to separate the excitation signal originated by air passing through the vocal tract from the effect of the vocal tract (acting as a filter shaping that excitation signal). The latter is more important for the analysis of the vocal signal. It is also common to take the first and second derivatives of the MFCC as additional features, indicative of temporal changes.

Short Time Fourier Transform (STFT) is essentially a spectrogram of the signal. Windowing is applied to short overlapping segments of the signal and FFT is computed. The result captures both spectral and time-dependent features of the signal.

3.1.2 Classifiers

Support Vector Machine (SVM) is a general binary classifier, trained to distinguish to groups. We use SVM to distinguish male and female speakers. Multi-class SVMs can be constructed using multiple binary SVMs, to distinguish between multiple groups. We used a multi-class SVM to distinguish between multiple speakers, and to recognize words from a limited dictionary.

Gaussian Mixture Model (GMM) has been successfully used for speaker identification [41]. We can train a GMM for each group in the training stage. In the testing stage we can obtain a match score for the sample using each one of the GMMs and classify the sample according to the group corresponding to the GMM that yields the maximum score.

Dynamic Time Warping (DTW) is a time-series matching and alignment technique [37]. It can be used to match time-dependent features in presence of misalignment or when the series are of different lengths. One of the challenges in word recognition is that the samples may differ in length, resulting in different number of segments used to extract features.

3.2 Speaker identification algorithm

Prior to processing we converted the gyroscope recordings to audio files in WAV format while upsampling them to 8 KHz. We applied silence removal to include only relevant information and minimize noise. The silence removal algorithm was based on the implementation in [29], which classifies the speech into voiced and unvoiced segments (filtering out the unvoiced) according to dynamically set thresholds for Short-Time Energy and Spectral Centroid features computed on short segments of the speech signal. Note that the gyroscope’s zero-offset yields particularly noisy recordings even during unvoiced segments.

We used statistical features based on the first 13 MFCC computed on 40 sub-bands. For each MFCC we computed the mean and standard deviation. Those features reflect the spectral properties which are independent of the pronounced word. We also use delta-MFCC (the derivatives of the MFCC), RMS Energy and

\[ \text{Approximated as logarithmic by the Mel-scale} \]
Spectral Centroid statistical features. We used MIRToolbox [32] for the feature computation. It is important to note that while MFCC have a physical meaning for real speech signal, in our case of a narrow-band aliased signal, MFCC don’t necessarily have an advantage, and were used partially because of availability in MIRToolbox. We attempted to identify the gender of the speaker, distinguish between different speakers of the same gender and distinguish between different speakers in a mixed set of male and female speakers. For gender identification we used a binary SVM, and for speaker identification we used multi-class SVM and GMM. We also attempted gender and speaker recognition using DTW with STFT features. All STFT features were computed with a window of 512 samples which, for sampling rate of 8 KHz, corresponds to 64 ms.

3.3 Speech recognition algorithm

The preprocessing stage for speech recognition is the same as for speaker identification. Silence removal is particularly important here, as the noisy unvoiced segments can confuse the algorithm, by increasing similarity with irrelevant samples. For word recognition, we are less interested in the spectral statistical features, but rather in the development of the features in time, and therefore suitable features could be obtained by taking the full spectrogram. In the classification stage we extract the same features for a sample \( y \). For each possible label \( l \) we obtain a similarity score of the \( y \) with each sample \( X^l \) corresponding to that guess in the training set. Let us denote this similarity function by \( D(y, X^l) \). Since different samples of the same word can differ in length, we use DTW. We sum the similarities to obtain a total score for that guess

\[
S^l = \sum_i D(y, X^l_i)
\]

After obtaining a total score for all possible words, the sample is classified according to the maximum total score

\[
C(y) = \arg\max_l S^l
\]

3.4 Experiment setup

Our setup consisted of a set of loudspeakers that included a sub-woofer and two tweeters (depicted in Figure 5). The sub-woofer was particularly important for experimenting with low-frequency tones below 200 Hz. The playback was done at volume of approximately 75 dB to obtain as high SNR as possible for our experiments. This means that for more restrictive attack scenarios (farther source, lower volume) there will be a need to handle low SNR, perhaps by filtering out the noise or applying some other preprocessing for emphasizing the speech signal.  

3.4.1 Data

Due to the low sampling frequency of the gyro, a recognition of speaker-independent general speech would be an ambitious long-term task. Therefore, in this work we set out to recognize speech of a limited dictionary, the recognition of which would still leak substantial private information. For this work we chose to focus on the digits dictionary, which includes the words: zero, one, two..., nine, and “oh”. Recognition of such words would enable an attacker to eavesdrop on private information, such as credit card numbers, telephone numbers, social security numbers and the like. This information may be eavesdropped when the victim speaks over or next to the phone.

In our experiments, we use the following corpus of audio signals on which we tested our recognition algorithms.

**TIDIGITS** This is a subset of a corpus published in [33]. It includes speech of isolated digits, i.e., 11 words per speaker where each speaker recorded each word twice. There are 10 speakers (5 female and 5 male). In total, there are \( 10 \times 11 \times 2 = 220 \) recordings. The corpus is digitized at 20 kHz.

3.4.2 Mobile devices

We primarily conducted our experiments using the following mobile devices:

---

9We tried recording in an anechoic chamber, but it didn’t seem to provide better recognition results compared to a regular room. We therefore did not proceed with the anechoic chamber experiments. Yet, further testing is needed to understand whether we can benefit significantly from an anechoic environment.

2. Nexus 7 tablet which according to a teardown analysis [14] is equipped with an InverSense MPU-6050 gyroscope and accelerometer.


3.5 Sphinx

We first try to recognize digit pronunciations using general-purpose speech recognition software. We used Sphinx-4 [47] – a well-known open-source speech recognizer and trainer developed in Carnegie Mellon University. Our aim for Sphinx is to recognize gyro-recordings of the TIDIGITS corpus. As a first step, in order to test the waters, instead of using actual gyro recordings we downsampled the recordings of the TIDITS corpus to 200 Hz; then we trained Sphinx based on the modified recordings. The aim of this experiment is to understand whether Sphinx detects any useful information from the sub-100 Hz band of human speech. Sphinx had a reasonable success rate, recognizing about 40% of pronunciations.

Encouraged by the above experiment we then recorded the TIDIGITS corpus using a gyro – both for Galaxy S III and Nexus 4. Since Sphinx accepts recording in WAV format we had to convert the raw gyro recordings. Note that at this point for each gyro recording we had 3 WAV files, one for each gyro axis. The final stage is silence removal. Then we trained Sphinx to create a model based on a training subset of the TIDIGITS, and tested it using the complement of this subset.

The recognition rates for either axes and either Nexus 4 or Galaxy S III were rather poor: 14% on average. This presents only marginal improvement over the expected success of a random guess which would be 9%.

This poor result can be explained by the fact that Sphinx’s recognition algorithms are geared towards standard speech recognition tasks where most of the voice-band is present and is less suited to speech with very low sampling frequency.

3.6 Custom recognition algorithms

In this section we present the results obtained using our custom algorithm. Based on the TIDIGITS corpus we randomly performed a 10-fold cross-validation. We refer mainly to the results obtained using Nexus 4 gyroscope readings in our discussion. We also included in the tables some results obtained using a Galaxy III device, for comparison.

Results for gender identification are presented in Table 3. As we see, using DTW scoring for STFT features yielded a much better success rate.

Results for speaker identification are presented in Table 4. Since the results for a mixed female-male set of speakers may be partially attributed to successful gender identification, we tested classification for speakers of the same gender. In this setup we have 5 different speakers. The improved classification rate (except for DTW for female speaker set) can be partially attributed to a smaller number of speakers.

The results for speaker-independent isolated word recognition are summarized in Table 5. We had correct classification rate of ~10% using multi-class SVM and GMM trained with MFCC statistical features, which is almost equivalent to a random guess. Using DTW with STFT features we got 23% correct classification for male speakers, 26% for female speakers and 17% for a mixed set of both female and male speakers. The confusion matrix in Figure 6, corresponding to the mixed speaker-set recorded on a Nexus 4, explains the not so high recognition rate, exhibiting many false positives for the words “6” and “9”. At the same time the recognition rate for
For a speaker-dependent case one may expect to get better recognition results. We recorded a set of 44 digit pronunciations, where each digit was pronounced 4 times. We tested the performance of our classifiers using “leave-one-out” cross-validation. The results are presented in Table 6, and as we expected exhibit an improvement compared to the speaker independent recognition\(^\text{10}\) (except for GMM performance that is equivalent to randomly guessing). The confusion matrix corresponding to speaker-dependent word recognition using DTW is presented in Figure 7.

DTW method outperforms SVM and GMM in most cases. One would expect that DTW would perform better for word recognition since the changing in time of the spectral features is taken into account. While true for Nexus 4 devices it did not hold for measurements taken with Galaxy III. possible explanation to that is that the low-pass filtering on the Galaxy III device renders all methods quite ineffective resulting in a success rate equivalent to a random guess. For gender and speaker identification, we would expect statistical spectral features based methods (SVM and GMM) to perform at least as good as DTW. It is only true for the Galaxy S III mixed speaker set and gender identification cases, but not for the other experiments. Specifically for gender identification, capturing the temporal development of the spectral feature wouldn’t seem like a clear advantage and is therefore somewhat surprising. One comparative study that supports the advantage of DTW over SVM for speaker recognition is [48]. It doesn’t explain though why it outperforms GMM which is a well established method for speaker identification. More experimentation is required to confirm whether this phenomenon is consistent and whether it is related to capturing the high frequencies.

### 3.7 Further improvement

We suggest several possible future improvements on our recognition algorithms. Phoneme recognition instead of whole words, in combination with an HMM could improve the recognition results. This could be more suitable since different pronunciations have different lengths, while an HMM could introduce a better probabilistic recognition of the words. Pre-filtering of the signal could be beneficial and reduce irrelevant noise. It is not clear which frequencies should be filtered and therefore some experimentation is needed to determine it.

\(^\text{10}\)It is the place to mention that a larger training set for speaker independent word recognition is likely to yield better results. For our tests we used relatively small training and evaluation sets.
For our experiments, we used samples recorded by the gyroscope for training. For speaker-dependent speech recognition we can imagine it may be easier to obtain regular speech samples for a particular speaker than a transcribed recording of gyroscope samples. Even for speaker independent speech recognition, it would be easier to use existing audio corpora for training a speech recognition engine than to produce gyroscope recordings for a large set of words. For that purpose it would be interesting to test how well the recognition can perform when the training set is based on normal audio recordings, downsampled to 200 Hz to simulate a gyroscope recording.

Another possible improvement is to leverage the 3-axis recordings. It is obvious that the three recordings are correlated while the noise of gyro readings is not. Hence, one may take advantage of this to get a composed signal of the three axes to get a better signal-to-noise ratio.

While we suggested that the signal components related to speech, and those related to motion lie in separate frequency bands, the performance of speech analysis in the presence of such noise is yet to be evaluated.

### 4 Reconstruction using multiple devices

In this section we suggest that isolated word recognition can be improved if we sample the gyroscopes of multiple devices that are in close proximity, such that they exhibit a similar response to the acoustic signals around them. This can happen for instance in a conference room where two mobile devices are running malicious applications or, having a browser supporting high-rate sampling of the gyroscope, are tricked into browsing to a malicious website.

We do not refer here to the possibility of using several different gyroscopic readings to effectively obtain a larger feature vector, or have the classification algorithm take into account the score obtained for all readings. While such methods to exploit the presence of more than one acoustic side-channel may prove very efficient we leave them outside the scope of this study. It also makes sense to look into existing methods for enhancing speech recognition using multiple microphones, covered in signal processing and machine learning literature (e.g., [23]).

Instead, we look at the possibility of obtaining an enhanced signal by using all of the samples for reconstruction, thus effectively obtaining higher sampling rate. Moreover, we hint at the more ambitious task of reconstructing a signal adequate enough to be comprehensible by a human listener, in a case where we gain access to readings from several compromised devices. While there are several practical obstacles to it, we outline the idea, and demonstrate how partial implementation of it facilitates the automatic speech recognition task.

We can look at our system as an array of time-interleaved data converters (interleaved ADCs). Interleaved ADCs are multiple sampling devices where each samples the signal with a sub-Nyquist frequency. While the ADCs should ideally have time offsets corresponding to a uniform sampling grid (which would allow to simply interleave the samples and reconstruct according to the Whittaker-Shannon interpolation formula [44]), usually there will be small time skews. Also, DC offsets and different input gains can affect the result and must all be compensated.

This problem is studied in a context of analog design and motivated by the need to sample high-frequency signals using low-cost and energy-efficient low-frequency A/D converters. While many papers on the subject exist, such as [27], the proposed algorithms are usually very hardware centric, oriented towards real-time processing at high-speed, and mostly capable of compensating for very small skews. Some of them require one ADC that samples the signal above the Nyquist rate, which is not available in our case. At the same time, we do not aim for a very efficient, real-time algorithm. Utilizing recordings from multiple devices implies offline processing of the recordings, and we can afford a long run-time for the task.

The ADCs in our case have the same sampling rate $F_s = 1/T = 200$. We assume the time-skews between them are random in the range $[0, T_Q]$ where for $N$ ADCs $T_Q = \frac{T}{N}$ is the Nyquist sampling period. Being located at different distances from the acoustic source they are likely to exhibit considerably different input gains, and possibly have some DC offset. [26] provides background for understanding the problems arising in this configuration and covers some possible solutions.

#### 4.1 Reconstruction algorithm

##### 4.1.1 Signal offset correction

To correct a constant offset we can take the mean of the Gyro samples and compare it to 0 to get the constant offset. It is essentially a simple DC component removal.

##### 4.1.2 Gain mismatch correction

Gain mismatch correction is crucial for a successful signal reconstruction. We correct the gain by normalizing the signal to have standard deviation equal to 1. In case we are provided with some reference signal with a known peak, we can adjust the gains of the recordings so that the amplitude at this peak is equal for all of them.
4.1.3 Time mismatch correction

While gyroscope motion events are provided with precise timestamps set by the hardware, which theoretically could have been used for aligning the recordings, in practice, we cannot rely on the clocks of the mobile devices to be synchronized. Even if we take the trouble of synchronizing the mobile device clock via NTP, or even better, a GPS clock, the delays introduced by the network, operating system and further clock-drift will stand in the way of having clock accuracy on the order of a millisecond\textsuperscript{11}. While not enough by itself, such synchronization is still useful for coarse alignment of the samples.

El-Manar describes foreground and background time-mismatch calibration techniques in his thesis [27]. Foreground calibration means there is a known signal used to synchronized all the ADCs. While for the purpose of testing we can align the recordings by maximizing the cross-correlation with a known signal, played before we start recording, in an actual attack scenario we probably won’t be able to use such a marker\textsuperscript{12}. Nevertheless, in our tests we attempted aligning using a reference signal as well. It did not exhibit a clear advantage over obtaining coarse alignment by finding the maximum of the cross-correlation between the signals. One can also exhaustively search a certain range of possible offsets, choosing the one that results in a reconstruction of a sensible audio signal.

Since this only yields alignment on the order of a sampling period of a single gyroscope ($T$), we still need to find the more precise time-skews in the range $[0, T]$. We can scan a range of possible time-skews, choosing the one that yields a sensible audio signal. We can think of an automated evaluation of the result by a speech recognition engine or scoring according to features that would indicate human speech, suggesting a successful reconstruction.

This scanning is obviously time consuming. If we have $n$ sources, we set one of the time skews (arbitrary) to 0, and have $n - 1$ degrees of freedom to play with, and the complexity grows exponentially with the number of sources. Nevertheless, in an attack scenario, it is not impossible to manually scan all possibilities looking for the best signal reconstruction, provided the information is valuable to the eavesdropper.

4.1.4 Signal reconstruction from non-uniform samples

Assuming we have compensated for offset, gain mismatch and found the precise time-skews between the sampling devices, we are dealing with the problem of signal reconstruction from periodic, non-uniform samples. A seminal paper on the subject is [28] by Eldar et al. Among other works in the field are [39, 46] and [31]. Sindhi et al. [45] propose a discrete time implementation of [28] using digital filterbanks. The general goal is, given samples on a non-uniform periodic grid, to obtain estimation of the values on a uniform sampling grid, as close as possible to the original signal.

A theoretic feasibility justification lies in Papoulis’ Generalized Sampling theorem [38]. Its corollary is that a signal bandlimited to $\pi/T_0$ can be recovered from the samples of $N$ filters with sampling periods $T = NT_0$.\textsuperscript{13} We suggest using one of the proposed methods for signal reconstruction from periodic non-uniform samples. With only several devices the reconstructed speech will still be narrow-band. While it won’t necessarily be easily understandable by a human listener, it could be used for better automated identification. Applying narrowband to wideband speech extension algorithms [36] might provide audio signals understandable to a human listener.

We suggest using one of the methods for signal reconstruction from periodic non-uniform samples mentioned above. With only several devices the reconstructed speech will still be narrow-band. For example, using readings from two devices operating at 200 Hz and given their relative time-skew we obtain an effective sampling rate of 400 Hz. For four devices we obtain a sampling rate of 800 Hz, and so on. While a signal reconstructed using two devices still won’t be easily understandable by a human listener, it could be used to improve automatic identification.

We used [28] as a basis for our reconstruction algorithm. The discussion of recurrent non-uniform sampling directly pertains to our task. It proposes a filterbank scheme to interpolate the samples such that an approximation of the values on the uniform grid is obtained. The derivation of the discrete-time interpolation filters is provided in Appendix A.

This method allows us to perform reconstruction with arbitrary time-skews; however we do not have at the time a good method for either a very precise estimation

---

\textsuperscript{11}Each device samples with a period of 5 ms, therefore even 1 ms clock accuracy would be quite coarse.

\textsuperscript{12}While an attacker may be able to play using one of the phones’ speakers a known tone/chirp (no special permissions are needed), it is unlikely to be loud enough to be picked up well by the other device, and definitely depends on many factors such as distance, position etc.

\textsuperscript{13}It is important to note that in our case the signal is not necessarily bandlimited as required. While the base pitch of the speech can lie in the range $[0, 200 \cdot N]$, it can contain higher frequencies that are captured in the recording due to aliasing, and may interfere with the reconstruction. It depends mainly on the low-pass filtering applied by the gyroscope. In InvenSense’s MPU-6050, Digital Low-Pass Filtering (DLPF) is configurable through hardware registers [11], so the conditions depend to some extent on the particular driver implementation.
of the time-skews or automatic evaluation of the reconstruction outcome (which would enable searching over a range of possible values). For our experiment we applied this method to the same set of samples used for speaker-dependent speech recognition evaluation, which was recorded simultaneously by two devices. We used the same value for $\tau$, the time-skew for all samples, and therefore chose the expected value $\tau = T/2$ which is equivalent to the particular case of sampling on a uniform grid (resulting in all-pass interpolation filters). It is essentially the same as interleaving the samples from the two readings, and we ended up implementing this trivial method as well, in order to avoid the adverse effects of applying finite non-ideal filters.

It is important to note that while we propose a method rooted in signal processing theory, we cannot confidently attribute the improved performance to obtaining a signal that better resembles the original, until we take full advantage of the method by estimating the precise time-skew for each recording, and applying true non-uniform reconstruction. It is currently left as an interesting future improvement, for which the outlined method can serve as a starting point. In this sense, our actual experiment can be seen as taking advantage of better feature vectors, comprised of data from multiple sources.

\subsection*{4.1.5 Evaluation}

We evaluated this approach by repeating the speaker-dependent word recognition experiment on signals reconstructed from readings of two Nexus 4 devices. Table 7 summarizes the final results obtained using the sample interleaving method\(^\text{14}\).

<table>
<thead>
<tr>
<th>SVM</th>
<th>GMM</th>
<th>DTW</th>
</tr>
</thead>
<tbody>
<tr>
<td>18%</td>
<td>14%</td>
<td>77%</td>
</tr>
</tbody>
</table>

There was a consistent noticeable improvement compared to the results obtained using readings from a single device, which supports the value of utilizing multiple gyroscopes. We can expect that adding more devices to the setup would further improve the speech recognition.

\section{5 Further Attacks}

In this section we suggest directions for further exploitation of the gyroscopes:

**Increasing the gyro’s sampling rate.** One possible attack is related to the hardware characteristics of the gyro devices. The hardware upper bound on sampling frequency is higher than that imposed by the operating system or by applications\(^\text{15}\). InvenSense MPU-6000/MPU-6050 gyroscopes can provide a sampling rate of up to 8000 Hz. That is the equivalent of a POTS (telephony) line. STMicroelectronics gyroscopes only allow up to 800 Hz sampling rate, which is still considerably higher than the 200 Hz allowed by the operating system (see Appendix C). If the attacker can gain a one-time privileged access to the device, she could patch an application, or a kernel driver, thus increasing this upper bound. The next steps of the attack are similar: obtaining gyroscope measurements using an application or tricking the user into leaving the browser open on some website. Obtaining such a high sampling rate would enable using the gyroscope as a microphone in the full sense of hearing the surrounding sounds.

**Source separation.** Based on experimental results presented in Section 2.3.4 it is obvious that the gyroscope measurements are sensitive to the relative direction from which the acoustic signal arrives. This may give rise to the possibility to detect the angle of arrival (AoA) at which the audio signal hits the phone. Using AoA detection one may be able to better separate and process multiple sources of audio, e.g. multiple speakers near the phone.

**Ambient sound recognition.** There are works (e.g. [42]) which aim to identify a user’s context and whereabouts based on the ambient noise detected by his smart phone, e.g restaurant, street, office, and so on. Some contexts are loud enough and may have distinct fingerprint in the low frequency range to be able to detect them using a gyroscope, for example railway station, shopping mall, highway, and bus. This may allow an attacker to leak more information on the victim user by gaining indications of the user’s whereabouts.

\section{6 Defenses}

Let us discuss some ways to mitigate the potential risks. As it is often the case, a secure design would require an

\(^{14}\text{We also compared the performance of the DTW classifier on samples reconstructed using the filterbank approach. It yielded a slightly lower correct classification rate of 75% which we attribute to the mentioned effects of applying non-ideal finite filters.}\)

\(^{15}\text{As we have shown, the sampling rate available on certain browsers is much lower than the maximum sampling rate enabled by the OS. However, this is an application level constraint.}\)
overall consideration of the whole system and a clear
definition of the power of the attacker against whom
we defend. To defend against an attacker that has only
user-level access to the device (an application or a web-
site), it might be enough to apply low-pass filtering to
the raw samples provided by the gyroscope. Judging by
the sampling rate available for Blink and WebKit based
browsers, it is enough to pass frequencies in the range 0 –
20 Hz. If this rate is enough for most of the applications,
the filtering can be done by the driver or the OS, subvert-
ing any attempt to eavesdrop on higher frequencies that
reveal information about surrounding sounds. In case a
certain application requires an unusually high sampling
rate, it should appear in the list of permissions requested
by that application, or require an explicit authorization
by the user. To defend against attackers who gain root
access, this kind of filtering should be performed at the
hardware level, not being subject to configuration. Of
course, it imposes a restriction on the sample rate avail-
able to applications.

Another possible solution is some kind of acoustic
masking. It can be applied around the sensor only, or
possibly on the case of the mobile device.

7 Conclusion

We show that the acoustic signal measured by the gyro-
scope can reveal private information about the phone’s
environment such as who is speaking in the room and,
to some extent, what is being said. We use signal pro-
cessing and machine learning to analyze speech from
very low frequency samples. With further work on low-
frequency signal processing of this type it should be pos-
sible to further increase the quality of the information
extracted from the gyro.

This work demonstrates an unexpected threat result-
ing from the unmitigated access to the gyro: applications
and active web content running on the phone can eaves-
drop sound signals, including speech, in the vicinity of
the phone. We described several mitigation strategies.
Some are backwards compatible for all but a very small
number of applications and can be adopted by mobile
hardware vendors to block this threat.

A general conclusion we suggest following this work
is that access to all sensors should be controlled by the
permissions framework, possibly differentiating between
low and high sampling rates.

Acknowledgements

We would like to thank Nimrod Peleg, from the Signal
and Image Processing Lab (SIPL) at the Technion, for
providing assistance with the TIDIGITS corpus. We also
greatful to Sanjay Kumar Sindhi, from IIT Madras, for
providing implementation and testing of several signal
reconstruction algorithms. We would also like to thank
Prof. Jared Tanner, from UC Davis, and Prof. Yonina Eldar,
from the Technion, for advising on reconstruction of
non-uniformly sampled signals. We thank Hriso Bojinov
for taking part in the initial brainstorming related to this
research and finally, Katharina Roesler, for proofreading
and advising on writing and formulation.

This work was supported by NSF and the DARPA
SAFER program. Any opinions, findings and conclu-
sions or recommendations expressed in this material are
those of the author(s) and do not necessarily reflect the
views of NSF or DARPA.

A Signal reconstruction from Recurrent
Non-Uniform Samples

Here we present the derivation of the discrete-time inter-
polation filters used in our implementation. The notation
in the expressions corresponds to the notation in [28].
The continuous time expression for the interpolation fil-
ters according to Eq. 18 in [28] is given by

\[ h_p(t) = a_p \text{sinc} \left( \frac{t}{T} \right) \prod_{q=0,q \neq p}^{N-1} \sin \left( \frac{\pi (t + t_p - t_q)}{T} \right) \]

We then sample this expression at times \( t = nT_Q - t_p \)
and calculate the filter coefficients for 48 taps. Given
these filters, the reconstruction process consists of up-
sampling the input signals by factor \( N \), where \( N = T/T_Q \)
is the number of ADCs, filtering and summation of the
outputs of all filters (as shown in Figure 8).

B Code for sampling a gyroscope via a
HTML web-page

For a web page to sample a gyro the DeviceMotion class
needs to be utilized. In the following we included a
JavaScript snippet that illustrates this:
if (window.DeviceMotionEvent) {
    window.addEventListener('devicemotion', function (event) {
        var r = event.rotationRate;
        if (r != null) {
            console.log('Rotation at [x,y,z] is: [' +
            r.alpha + ', r.beta + ', r.gamma + ']');
        }
    });
}

Figure 9 depicts measurements of the above code running on Firefox (Android) while sampling an audio chirp 50 – 100 Hz.

C Gyroscope rate limitation on Android

Here we see a code snippet from the Invensense driver for Android, taken from hardware/inven sensors/65xx/libsensors_jio/MPLSensor.cpp. The OS is enforcing a rate of 200 Hz.

D Code Release

We provide the source code of the Android application we used for recording the sensor measurements, as well as the Matlab code we used for analyzing the data and training and testing of the speech recognition algorithms. We also provide the gyroscope recordings used for the evaluation of our method. The code and data can be downloaded from the project website at http://crypto.stanford.edu/gyrophone. In addition, we provide a web page that records gyroscope measurements if accessed from a device that supports it.

References


