PitchIn: Eavesdropping via Intelligible Speech Reconstruction using Non-Acoustic Sensor Fusion

Jun Han
Carnegie Mellon University
Moffett Field, CA
jun.han@sv.cmu.edu

Albert Jin Chung
Carnegie Mellon University
Moffett Field, CA
albert.chung@sv.cmu.edu

Patrick Tague
Carnegie Mellon University
Moffett Field, CA
patrick.tague@sv.cmu.edu

ABSTRACT

Despite the advent of numerous Internet-of-Things (IoT) applications, recent research demonstrates potential side-channel vulnerabilities exploiting sensors which are used for event and environment monitoring. In this paper, we propose a new side-channel attack, where a network of distributed non-acoustic sensors can be exploited by an attacker to launch an eavesdropping attack by reconstructing intelligible speech signals. Specifically, we present *PitchIn* to demonstrate the feasibility of speech reconstruction from non-acoustic sensor data collected offline across networked devices. Unlike speech reconstruction which requires a high sampling frequency (e.g., > 5 KHz), typical applications using non-acoustic sensors do not rely on richly sampled data, presenting a challenge to the speech reconstruction attack. Hence, *PitchIn* leverages a distributed form of Time Interleaved Analog-Digital-Conversion (TI-ADC) to approximate a high sampling frequency, while maintaining low per-node sampling frequency. We demonstrate how distributed TI-ADC can be used to achieve intelligibility by processing an interleaved signal composed of different sensors across networked devices. We implement *PitchIn* and evaluate reconstructed speech signal intelligibility via user studies. *PitchIn* has word recognition accuracy as high as 79%. Though some additional work is required to improve accuracy, our results suggest that eavesdropping using a fusion of non-acoustic sensors is a real and practical threat.

CCS CONCEPTS

•Security and privacy →Embedded systems security;

KEYWORDS

Sensor Fusion; Speech Reconstruction; Non-acoustic Sensors; Security; Privacy

ACM Reference format:
Jun Han, Albert Jin Chung, and Patrick Tague. 2017. PitchIn: Eavesdropping via Intelligible Speech Reconstruction using Non-Acoustic Sensor Fusion. In *Proceedings of The 16th ACM/IEEE International Conference on Information Processing in Sensor Networks, Pittsburgh, PA USA, April 2017 (IPSN 2017),* 12 pages.

IPSN 2017, Pittsburgh, PA USA

DOI: 10.1145/3055031.3055088

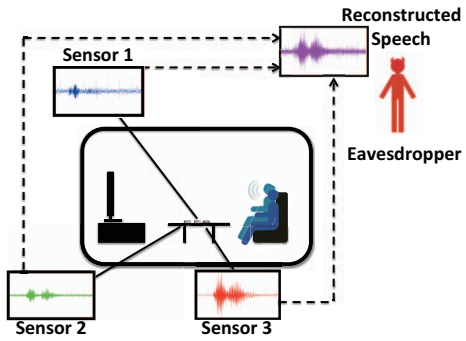


Figure 1: Example scenario where non-acoustic sensors in IoT devices are "listening" to conversations.

DOI: 10.1145/3055031.3055088

1 INTRODUCTION

Emerging technologies in the Internet of Things (IoT) give rise to wide deployment of pervasive networked sensors. This trend is evidently increasing as recently demonstrated [33, 39], projecting an IoT and global sensor market of $1.7 Trillion and $190.6 Billion, respectively, by 2021. As the number of IoT devices increase, sensors will surround us to monitor various parts of our lives at homes, offices, and numerous other places.

While sensors contribute to numerous constructive applications, some of the recent research demonstrate the feasibility of launching side-channel attacks to leak privacy sensitive information. Specifically, these research demonstrate the feasibility of inferring sensitive information of a victim (e.g., location or keystrokes) using only accelerometer data from a smart phone [19, 30, 34].

All of the aforementioned side-channel attacks focus on extracting private information from an individual sensor. However, the expected penetration of IoT devices into our homes and workplaces inspires us to consider additional threats due to wide deployment of sensors, including structural and activity sensors used for common IoT applications. We are witnessing structural sensors such as geophones and accelerometers in smart buildings and smart cities [25, 35, 36, 46] often as array of sensors for various applications such as occupancy, structural health, and earthquake monitoring. Furthermore, beyond the already prevalent sensing capabilities of smartphones, smart watches, and tablets, we are now seeing activity sensors (such as accelerometers and gyroscopes) deployed

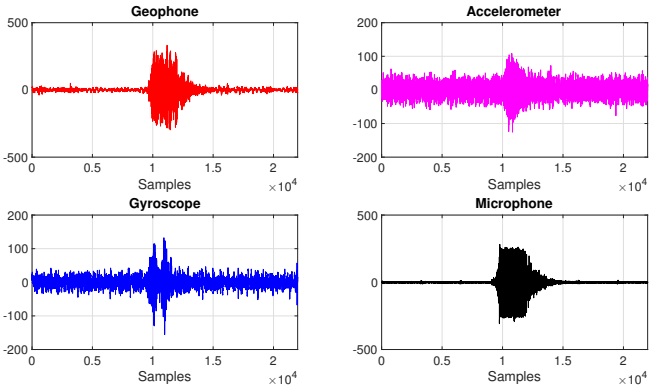


Figure 2: Recording of single geophone, accelerometer, gyroscope, and microphone of the word "one" each sampled at 8 KHz.

in smart TV remotes and gaming controllers (e.g., Wii remote, PS4 and Xbox controllers) [16, 45]. Many of these sensors are widely deployed in experimental and generic wireless sensor boards for multi-purpose sensing [26, 28, 48], and we expect their deployment and inclusion in commercial services to increase dramatically based on the market projections mentioned above.

With such wide deployment of sensors in IoT devices, we find that a large portion of research community has concentrated on finding and defending against vulnerabilities of individual sensors or devices, and what the posed risks are for the users. However, we are more interested in exploring new vulnerabilities if an attacker compromises data collected from multiple devices. Specifically, we pose the question – what unforeseen information can one extract from fusion of these sensors across networked devices?

In search for the answer to the above question, we present ***PitchIn*** to demonstrate the feasibility of achieving the seemingly unrealizable goal of reconstructing an *intelligible speech signal* by fusing non-acoustic sensor data collected from a network of nodes. Specifically, we consider scenarios of potential security breaches of a smart home/office's gateway or in service provider's database, which has logs of sensor data from victim's IoT devices. Such breaches have been witnessed in many real-world examples recently [7, 9]. Hence, the attacker does not have to compromise individual devices equipped with sensors in victim's home or office to gain access to the sensor data. We illustrate an example scenario depicted in Figure 1.

Traditionally, non-acoustic sensors such as geophones, accelerometers, and gyroscopes are thought to be unresponsive to acoustic signals, as they are designed to capture motion signals (vibrations, movements, and tilt angles, respectively). However, we find from our experiments and from related work, that when exposed to sound waves, the sensors vibrate to output minuscule signals, sufficient to be processed to reconstruct intelligible acoustic signals [32, 50]. Figure 2 depicts the time series plots of non-acoustic sensors such as a geophone, an accelerometer (x-axis), and a gyroscope (x-axis), when sampled at 8 KHz[1]. We also show microphone data for comparison.

Unfortunately, a sampling frequency of 8 KHz is much higher than the typical rate at which these motion sensors are configured to be sampled at in commercial devices (further discussed in detail in Section 2.1.2). Obtaining intelligible speech signals, however, require a high sampling frequency, with a minimum of 5 KHz [37], while telephones and CDs are sampled at 8 KHz and 44.1 KHz, respectively [29] for higher quality audio. Hence, an attacker cannot recover an intelligible speech from sensor data of a single device.

Hence, to increase the overall system sampling frequency, *PitchIn* builds upon the idea of Time Interleaved Analog-Digital-Conversion (TI-ADCs) [22], which is a method to parallelize the sampling task with multiple ADCs with temporal offset. *PitchIn* extends this idea to create **Distributed TI-ADCs** so that the reconstructed signal, which we refer to as the *Amalgam signal*, has an overall effect of being sampled at a high sampling frequency. In reality, however, each node is sampled at a much lower sampling frequency. Hence, each node is "*pitching in*" to contribute to the *Amalgam* signal.

Even with the high overall *Amalgam* signal sampling frequency thanks to *PitchIn*'s Distributed TI-ADC, achieving intelligibility from the reconstructed *Amalgam* signal is extremely challenging because fusion of sensor data creates mismatches in amplitude alignments and causes distortions. Hence, we transform the signals using different signal processing techniques (e.g., normalization and denoising) to reconstruct a final speech signal that can be interpreted by humans.

We evaluate the intelligibility of *PitchIn* via a user study (approved by our Institutional Review Board (IRB)) by reconstructing two sets of *Amalgam* signals constructed of varying number sensors sampled with per node sampling frequency of 500 Hz and 1 KHz.

In summary, we present the following contributions.

- We present an eavesdropping attack by enabling **intelligible speech signal reconstruction** by fusing seemingly innocuous *non-acoustic sensory data* across networked sensor devices: the reconstructed signal has a high sampling frequency despite *low per-node sampling frequency* by leveraging the distributed TI-ADC. We highlight that *PitchIn* is an eavesdropping attack, and is not a substitute of an Automated Speech Recognition (ASR) engines, although *PitchIn* can be complemented with ASRs to increase the efficiency of the attack (Discussed in Section 6.4).
- We demonstrate a **feasibility study and evaluation** of *PitchIn* and resulting *Amalgam* signals: we study the feasibility of speech recognition via proof-of-concept implementation and evaluation of human recognition of the resulting signals. We demonstrate that *PitchIn*'s reconstructed signals yield highest recognition accuracy of 79%, 53%, and 35% for varying sensor modalities, sampling frequencies, and number of nodes.

[1]We note that we only use x-axis of accelerometer and gyroscope throughout this paper for simplicity, but the axis can be interchanged or combined with other axes.

2 BACKGROUND

In this section, we first present information on sensor device physics and where these sensors are used today in different IoT applications. Following that, we discuss the main idea of interleaved ADC and how it increases the overall sampling frequency. We then present relevant information on speech intelligibility. We then present the related work.

2.1 Sensors

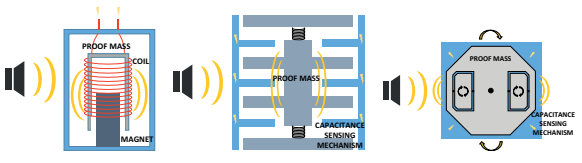
We now introduce a brief discussion about how each sensor captures physical signals and transform them into electrical signals, as depicted in Figure 3, and present their real-world use cases in IoT applications.

2.1.1 *Sensor Device Physics.*

Geophone captures mechanical vibrations that travel through solid media [13]. As mechanical waves reach the base of a geophone, small vibrations cause the base magnet to vibrate. Subsequently, an electrical coil attached to the proof mass experiences changes in magnetic flux, which in turn translate the mechanical signal to voltage induction, which is output as analog signal. As geophones are tuned to capture longitudinal mechanical waves, it is no surprise that vibrations from sound waves induce small vibrations of the sensory mechanism, so acoustic waves are registered as small but detectable signals in the analog output.

Accelerometer similarly captures mechanical vibrations through its sensing axes [15] (Figure 3(b)). As the MEMS sensor accelerates along the axis of interest, a fictitious inertial force shifts the proof mass to swing between springs. The change in the distance between the metal plates results in the change in capacitance, yielding the analog signal change which can be mapped to the acceleration value using a predetermined conversion factor. Since acoustic waves exert a force on the proof mass, small vibrations occur and yield an analog signal output that would otherwise be interpreted as acceleration.

Gyroscope MEMS gyroscopes also have a similar structure to that of MEMS accelerometers [44]. Ass a gyroscope is rotated, the proof mass rotates as a result of the fictitious Coriolis force. This force is analogous to that of inertial force in translation. As metal plates rotate as a response, the capacitance change is registered as an analog signal. As acoustic waves come in contact with a MEMS gyroscope, small vibrations that reach the proof mass also create vibration along the rotating axis, translating to electrical signals through capacitance.



(a) Geophone (b) Accelerometer (c) Gyroscope

Figure 3: Illustration of how mechanical sensors translate physical movements into voltages.

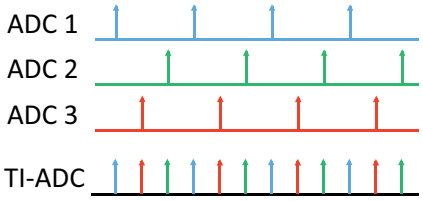


Figure 4: Illustration of how TI-ADC increases the overall sampling frequency by leveraging multiple ADCs in parallel with temporal offset.

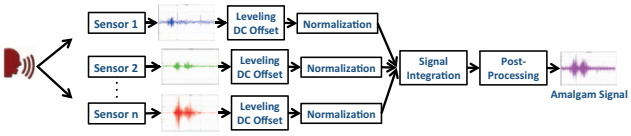


Figure 5: System overview diagram of *PitchIn* speech signal reconstruction.

2.1.2 *Sensors Embedded in IoT Devices.*

Different IoT devices have various sensors depending on their applications. We highlight example IoT devices that include geophone, accelerometer, or gyroscope. Different devices sample these sensors at varying frequencies depending on the application. Higher sampling frequency captures more information resulting in more accurate representation of the signal, but at a cost of higher computational and energy costs. Table 1 depicts some of the IoT applications and the corresponding sensor modalities.

Structural and building monitoring solutions leverage (often an array of) sensors such as geophones, accelerometers, and gyroscopes. Device-free user occupancy identification and monitoring solutions are proposed for smart buildings [35]. Structural health monitoring devices monitor the condition of buildings and/or bridges [25, 46]. Earthquake detection devices and indoor footstep monitoring systems also leverage geophones to measure and analyze seismic vibrations, and perform occupancy monitoring, respectively [36]. These devices sample on the order of 1 KHz.

Furthermore, mobile devices such as smartphones, smart watches, and tablets embed a large number of sensors, including accelerometers and gyroscopes, used for various applications (e.g., activity/gesture recognition, gaming, etc). Mobile OSes such as iOS and Android restrict the sampling frequencies of these sensors to a maximum of 200 Hz. Controllers for gaming consoles (e.g., Wii Remote, PS4 Dualshock4 Controller, Xbox Controller) embed accelerometers and gyroscopes to detect user motion for dynamic gaming experiences [45]. Similarly, smart TV remotes embed sensors for user gesture recognition and identification [16]. These devices sample sensors on the order of 100 Hz.

2.2 Time Interleaved ADC

Time Interleaved Analog-Digital Conversion (TI-ADC) has been shown to acquire high sampled data on resource-constrained systems. The main idea behind TI-ADC is that while each ADC is bounded by a relatively low sampling frequency, it is possible to

IoT Devices for Different Applications	Structural Health Monitoring Device	Earthquake Detection Device	Footstep Monitoring Device	Smartphones/ Smartwatches/ Tablets	Gaming Controller (Wii Remote)	Smart TV Remote
Sensors	Geophone/ Accelerometer/ Gyroscope	Geophone	Geophone	Accelerometer/ Gyroscope	Accelerometer/ Gyroscope	Accelerometer/ Gyroscope

Table 1: IoT devices used for different applications and the corresponding sensors embedded in the devices.

increase the effective sampling rate by using multiple ADCs in parallel. Specifically, a set of multiple ADCs are placed at different temporal points to sample at a low frequency [22]. Subsequently, software recombines the pieces of sampled data. Assuming time synchronization, TI-ADC allows effective sampling frequency to increase by a factor of the number of ADCs. This is depicted in Figure 4. *PitchIn* builds upon this idea, but rather than using multiple ADCs on a single physical system, we treat distributed devices in a network as "virtual" ADCs.

2.3 Speech Intelligibility

Two of the main factors contribute to achieving speech intelligibility – (1) sampling frequency and (2) contextual information. Human auditory systems process acoustic signals up to 20 KHz. Due to the Nyquist sampling theorem – which defines minimum required sampling rate for the signal [41] – audio files on CDs are created using a sampling rate of 44.1 KHz to avoid distortion [29]. We also note that minimum sampling frequency of 5 KHz is required for intelligibility of human speech signals [37].

Another factor to consider is the context within speech. Speech recognition by humans is known to be a complex experience that subconsciously perceives words that make best sense within the given context. When a distorted signal is presented to human perceptual system, it is known to perform much better when the context of the information is also presented [47]. Inspired by the human speech recognition, automatic speech recognition (ASR) tools also use language models to increase the recognition accuracy [24].

In this paper, we take into consideration how sampling frequency from each sensor affects reconstruction of speech signals. Furthermore, we also take into consideration of contextual information when designing our user study to reflect the reality of speech recognition performed by humans.

2.4 Related Work

We now present related work relevant to *PitchIn*. We first present papers that exploit a non-acoustic sensors to capture sound signals. We then present related work exploring methods to leak side-channel information via sensor data.

2.4.1 Sensors Capturing Acoustic Signals.

Sensors in Smartphones. Recent research has demonstrated keyword detection using an accelerometer [50] and a gyroscope [32] in smartphones. Gyrophone demonstrates that commercial gyroscopes that are implemented in smartphones are capable of capturing acoustic signal even at low sampling frequency [32]. With proper signal processing and machine learning algorithms, this is enough to show speaker identification and speech finger printing. AccelWord demonstrates hot word detection using accelerometer, while achieving low energy consumption [50]. In addition to demonstrating high accuracy in hot word detection, this work also demonstrates the feasibility of an accelerometer capturing rich data more so than conventionally expected.

However, both of these approaches rely on machine learning to train a classifier on a small, predefined group of keyword fingerprints (on the order of tens of words) and later test whether the spoken words' fingerprints match the trained fingerprints, neither reconstructing intelligible speech signals. While these are promising first steps, each work mainly focuses on recovering fingerprints of a small predefined word group. Furthermore, we find that Gyrophone is limited as a practical eavesdropping tool because of the low recognition accuracy when evaluating *speaker-independent* experiments, which resembles a more realistic attack scenario than *speaker-dependent* experiment, yet only yielding 7% to 17% on different phones. Gyrophone also provides a preliminary evaluation of interleaving two gyroscope signals from different smartphones to increase the overall sampling frequency. However, Gyrophone neglects to evaluate the results of *speaker-independent* experiments. We imply that the results must be less accurate than that of the single sensor experiment which yielded a best case of 17% because the recognition accuracy from the interleaved signals would not be higher than that of a single sensor experiment.

In this paper, we are rather interested in reconstructing intelligible speech signals *without restriction of predefined keywords nor any prior training*. Instead of predefined keywords, we can leverage any additional context information relevant to the deployment scenario to infer a restricted language model that is independent of the *Amalgam* signal, which aids in speech intelligibility. Hence, the problem we are tackling is necessarily more challenging than the previous approaches because there is no prior restriction on possible fingerprints when the *Amalgam* signal is constructed, requiring much more information to be extracted from the *Amalgam* signal.

Sensors embedded in Non-smartphone Devices. There have been approaches to capture acoustic signals from non-smartphone environments as well. Son et al. describe how gyroscopes respond to acoustic signals of certain frequency, enough to malfunction the flight control of drones [43]. Visual Microphone leverages a camera to capture small vibrations on object surfaces due to sound waves, which recovers the acoustic signal of the sound source [14]. Once again, while *PitchIn* has a synonymous initial idea of capturing sound signals from non-acoustic sensors, we are more interested in

fusing disparate non-acoustic sensors that inherently are sampled at low sampling frequencies.

2.4.2 Side-Channel Attacks.
ACComplice presents a side-channel attack on an accelerometer in a smartphone by inferring a driver's starting location within a 200 meter radius, along with the traveled route [19]. ACCessory also exploits vulnerabilities of an accelerometer in a smartphone by inferring tapped keystrokes, and is able to extract six character passwords within a median of 4.5 trials [34]. spiPhone uses accelerometer readings of a smartphone placed close to a computer keyboard to infer text entered on the keyboard [30]. These work look into exploiting sensor side-channel vulnerabilities from a single device. *PitchIn*, however, looks into interesting potential vulnerabilities when fusing sensor signals from different devices.

3 THREAT MODEL

We now present the threat model of *PitchIn*. Specifically, we present the goals and capabilities of the attacker as well as the assumptions made. The main goal of the attacker is to launch a successful eavesdropping attack on victim's spoken verbal communications in his/her home, office, conference rooms, etc. Specifically, we consider an *offline attack* made possible by potential breaches of recorded sensor data from a gateway in a smart home or service provider's database, often encountered in many real-world incidents [7, 9, 18]. However, each of these sensor data are sampled at low sampling rate, resulting in non-intelligible sound. Furthermore, we consider attackers who does not have the capability of remotely controlling individual device to modify and increase the sampling frequency. The attacker thus attempts to interleave multiple signals offline to achieve a *Amalgam* signal that has an overall effect of a single device with a high sampling rate, increasing the intelligibility. We note that the attacker only launches *PitchIn* attack if (s)he does not gain access to a microphone data (always sampled with high sampling rate). Otherwise, the attacker will directly make use of the microphone data instead of the non-acoustic sensor data, eliminating the need to interleave signals of different devices in the first place. This is a reasonable assumption because there are not many homes, offices, and conference rooms that are constantly recording microphone data, as opposed to structural or motion sensors, which are designed to constantly monitor their environment.

4 DESIGN AND IMPLEMENTATION

We now discuss the implementation details of reconstructing an intelligible *Amalgam* signal by fusing data collected from a network of sensors. We first present an overview of the *Amalgam* signal generation, and then discuss the details.

4.1 Overview

To construct *Amalgam* signals from different sensors, *PitchIn* leverages a distributed form of Time Interleaved Analog-Digital Conversion (Distributed TI-ADC). This is to generate an effect of high sampling frequency (Fs_{Amal}) signal from a fusion of multiple sensor data that are sampled at low per-node sampling frequency (Fs_{sensor}). However, distributed TI-ADC requires addressing difficult challenges to produce an intelligible speech signal. Figure 5 depicts the flow chart diagram of *PitchIn Amalgam* generation steps. First, each sensor data is sampled locally with its low Fs_{sensor}. Then each individual signal is leveled to account for DC offset mismatches that occurred during the ADC phase. Subsequently, individual signals are normalized to be aligned because different physical sensors lead to gain mismatches. We then leverage distributed TI-ADC to interleave different signals into one *Amalgam* signal and then perform post-processing such as interpolation and denoising.

4.2 Main Challenges of Amalgam Generation

We discuss in detail how *PitchIn* addresses the following main challenges: *levelling DC offset*, *gain normalization*, accounting for *temporal offset mismatches*, and *post-processing*.

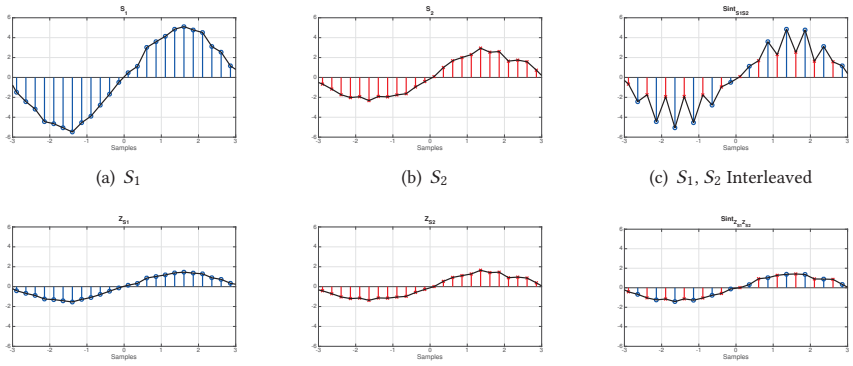
4.2.1 Leveling DC Offset.
Data sets from different sensors may have distinct DC offset, or average value offset from 0 volts [11] due to variations in hardware. With the aggregated data from all the nodes, *PitchIn* reconstructs the *Amalgam* signal by first leveling the DC offset. Leveling the DC offset is important to speech intelligibility because the DC offset contributes to either a clipping of loudest parts of the signal, distortions, and/or reduced audio volume.

4.2.2 Gain Normalization.
Data sets from different sensors also exhibit different amplitude levels due to the differences in how each sensor captures the vibrations from the sound signal and the differences in the amplification level before going through the ADC. Amplitude normalization is imperative for *PitchIn* to reconstruct intelligible speech signal by fusing different sensor readings. Figure 6 depicts a toy example that illustrates this concept. Figure 6(a) and 6(b) depict two signals, S_1 and S_2, respectively, exemplifying noisy sensor readings of a sinusoidal signal with non-aligned amplitudes. Figure 6(c) depicts the resulting interleaved signal, $Sint_{S1S2}$, when no amplitude normalization is performed. (We explain the details of signal interleaving in Section 4.2.3 and 5.3.) We note that the resulting signal is heavily distorted.

However, we show the effect of normalization with the remaining subfigures. Figures 6(d) and 6(e) depict Z_{S_1} and Z_{S_2}, which are output of Z-Score normalization of S_1 and S_2, respectively. Figure 6(f) depicts the resulting interleaved signal, $Sint_{Z_{S_1}Z_{S_2}}$ of the normalized signals, Z_{S_1} and Z_{S_2}. As depicted from this figure, the resulting signal has a high resemblance to the original sinusoidal signal.

While other types of normalization methods may be applied, we leverage Z-Score because it computes the statistical quantification of how much each score is distant from the mean in terms of standard deviations. Within a sensory modality, the signal to noise ratio of audio signal is expected to be similar between the sensors. This allows usage of Z-scores to project the signals in a statistically normalized space, where the amplitude of the signals in all the sensors will be aligned to one another based on signal to noise ratio. The normalized value of Z-Score Z_{S_i} is computed for data S_i from the ith sensor that has a known mean μ_i and standard deviation σ_i is computed as $Z_{S_i} = (S_i - \mu_i)/\sigma_i$.

4.2.3 Accounting for Temporal Offset Mismatches.
Different devices start sampling their sensors at different times. We



(a) S_1 (b) S_2 (c) S_1, S_2 Interleaved

(d) Z_{S_1} (e) Z_{S_2} (f) Z_{S_1}, Z_{S_2} Interleaved

Figure 6: A toy example of amplitude normalization and its effects.

note that to achieve the best results for the distributed TI-ADC, each device has to sample at a regular interval relative to each other, resulting in perfectly interleaved signals. This increases the *Amalgam* signal sampling frequency by n times, where n is the number of sensors. As a proof-of-concept, we demonstrate this with experiments in Section 5. However, achieving perfectly interleaved data is extremely infrequent in practice. Rather, the temporal offset is close to being modeled as random. We demonstrate that even with such limitations, *PitchIn* obtains reasonable recognition accuracy depicted in Section 5.3.2.

4.2.4 Post-processing.

Interpolation. Once data points are collected, spline interpolation is used to estimate the original signal. We interpolate the signal to output a *Amalgam* signal with a sampling frequency of 40 KHz. This method uses pieces of polynomials to estimate the region with no signal. Because spline interpolation has no restriction on how available data points are spaced, it is appropriate to use especially in the current implementation where data points may be available at random temporal offsets.

Filtering. We then perform high-pass filtering to the normalized signal to remove the transient noise. We leverage a fourth-order Butterworth filter [12] with a cutoff frequency at 300 Hz. Butterworth filter design uniformly preserves the passband frequency, while attenuating stopband frequencies.

5 EVALUATION

We now describe the evaluation details of *PitchIn* eavesdropping attack. We first present the experiment setup and the implementation details. We then present and analyze different evaluation scenarios. We report corresponding statistical test results in Appendix A.

5.1 Experiment Setup

Apparatus. We implement *PitchIn* by interfacing the sensors with Arduino Uno boards [4]. Each Arduino board interfaces with one distinct sensor, namely a geophone, accelerometer, or gyroscope. For ground truth, we also interface an Arduino with a microphone. The apparatus is depicted in Figure 7. The SM-24 geophone [13] is designed to detect ground movement and translates to an output voltage. The ADXL-335 three-axis MEMS accelerometer [15] measures and creates signals to represent the acceleration experienced by the sensor in the range of -3 to 3 g. The LPY403AL two-axis gyroscope [44] measures and outputs signals for the angular velocity of the pitch (X) and yaw (Z) axes in the range of -30 to 30°/s. Each sensor is amplified in hardware using two operational amplifiers [23] and then fed into the Arduino's ADC. We refer to each of the board-sensor combinations as a *node*.

The Arduino Uno board uses an 8-bit ATmega328P microprocessor [10]. It has 32 KB flash memory, 2 KB SRAM, and 1 KB EEPROM and a clock speed of 16 MHz. It has six analog interface pins. The single ADC has a resolution of 10 bits and output voltage range of 0 to 5 Volts. In our work, we modify the Arduino setting to range from 0 to 3.3 Volts to match the maximum output voltage of the sensors.

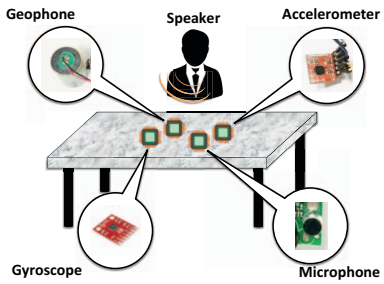


Figure 7: Experimental apparatus with a geophone, an accelerometer, a gyroscope, and a microphone.

Data Collection. Each node logs data on a microSD card, leveraging Arduino Ethernet Shield [2]. We make use of SdFat Analog Bin Logger library [3] to enable low latency SD card writes so the Arduino can write while sampling at such a high frequency.

We place the apparatus on a desk about a meter away from the person speaking (henceforth called speaker). The speaker's average Sound Pressure Level (SPL) is 85 dB, a typical "presentation-level"

Names of People	Joseph	Catherine	Thomas	Jefferson	Elizabeth	Michelle	Anthony	Emmanuel	Hilary	Patrick
Cities	Atlanta	Los Angeles	New York	San Francisco	Washington D.C.	Paris	London	Moscow	Tokyo	Hong Kong
Companies	Apple	Microsoft	Google	Facebook	Amazon	Comcast	Tesla Motors	Starbucks	Walmart	United Airlines
Numbers	One	Two	Three	Four	Five	Six	Seven	Eight	Nine	Ten

Table 2: WG_1, WG_2, WG_3, and WG_4 of *names of people, cities, companies,* and *numbers (1 to 10),* respectively.

volume. We measure the average SPL using SkyPaw's dBMeter app on an iPhone 6 [42] positioned close to the speaker. The speaker is a male and a fluent but non-native English speaker.

User Study Process. The goal of this study is to determine the intelligibility of the reconstructed *Amalgam* signals. Participants were given instructions to transcribe recordings of different words. The participants were given an additional information of the word group that each recording belongs to. There are four word groups of ten words, WG_1 constituting names of *people*, WG_2 constituting names of *cities*, WG_3 constituting names of *companies*, and WG_4 constituting *numbers* from one to ten. The additional information serve to provide contextual information synonymous to context within speech (e.g., words in a sentence), reflecting the reality of how humans perform speech recognition [47]. The words are listed in Table 2.

We recruit a total of 230 participants, and presented randomized words so that each participant does not listen to the same word from different signals. Hence, each data point in the figures of this section consists of 230 transcriptions. The participants were recruited via Amazon Mechanical Turk [8]. We performed the user study after receiving approval from our Institutional Review Board (IRB) and complied to the IRB's recommendation.

5.2 Non-Acoustic Sensors

Before presenting the *Amalgam* construction, we first evaluate how each of the individual non-acoustic sensors respond to human speech, and how the intelligibility varies corresponding to their sampling frequencies, F_s. We further investigate these sensors to test the relationship between the recognition accuracy (i.e., intelligibility) and the sampling frequency, F_s. Figure 8 depicts the recognition accuracy of non-acoustic geophone, accelerometer, and gyroscope sensors each sampled at varying sampling rate (i.e., F_s = {1KHz, 2KHz, 4KHz, and 8KHz}), compared to the baseline case of a microphone. This figure clearly depicts the fact that the non-acoustic sensors respond to speech signals, yielding non-negligible accuracies when sampled at 8 KHz. We note the trend of increasing recognition accuracy as F_s increases from 1 to 8 KHz. Additionally, the accuracy is extremely low for all sensors when F_s=1 KHz, including the microphone. Hence, we highlight that intelligibility decreases significantly as the sampling frequency decreases. We demonstrate statistical significance of the results with paired t-test reported in Appendix A (along with t-test results of all following evaluations in this section).

To provide a better understanding of these signals and deeper insight into our results, we have posted audio and video clips at http://mews.sv.cmu.edu/research/pitchin/. The video clips show spectrogram reconstructions of the spoken word "apple" using the open source audio editor Audacity. We strongly advise the readers to view the video clips together with the figures in this section.

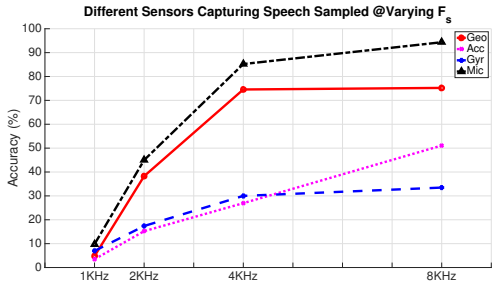


Figure 8: Recognition accuracy increases as F_s increases for each sensor.

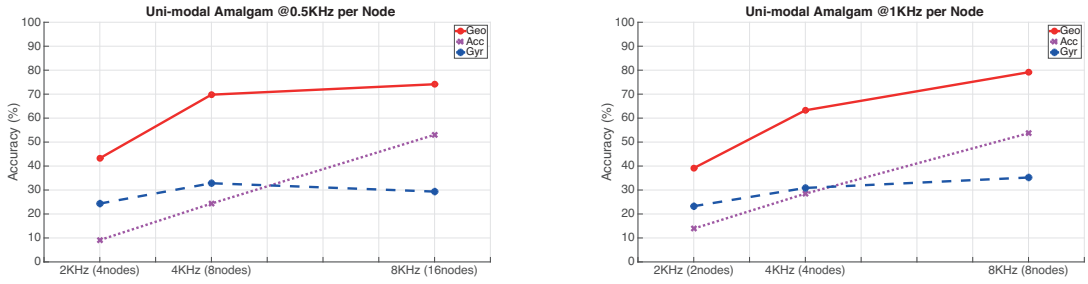
5.3 Amalgam Evaluation

We evaluate *Amalgam* signals constructed from fused sensor data. We first present the results of a proof-of-concept where sensor fusion is performed by interleaving signals with a regular temporal offset. We then present the results when we relax this assumption, more closely resembling the real-world scenarios. We also present an idea of fusing sensor data across sensor modalities.

5.3.1 *Ideal Temporal Offset.*

We test the effects of achieving a higher *Amalgam* sampling frequency $F_{s_{Amal}}$ as we increase the number of nodes that "pitch in" to constructing the *Amalgam* signal. We report two sets of experiments as following. In the first experiment, we fix the per node sampling frequency, F_s=500 Hz, and vary the number of nodes to 4, 8, and 16. Similarly, in the second experiment, we fix F_s=1 KHz and vary the number of nodes to 2, 4, and 8. Both experiments yield $F_{s_{Amal}}$ of 2 KHz, 4 KHz, and 8 KHz. Figures 9(a) and 9(b) depict the two experiments, respectively. We defer the discussion of how we "simulate" different sensor data from a single physical sensor readings for each of these sensors in Section 5.3.3.

In both experiments, the trend of increasing recognition accuracy with increasing $F_{s_{Amal}}$ is preserved, similar to the non-*Amalgam* findings depicted in Figure 8. More specifically, the accuracy (i.e., intelligibility) significantly increases within most sensor modalities, yielding accuracies as high as 79%, 53%, and 35%, for geophone,



(a) F_s=500Hz (b) F_s=1KHz

Figure 9: *Amalgam* signals constructed with F_s=500 Hz and 1KHz for Figure 9(a) and 9(b), respectively. Recognition accuracy of each *Amalgam* signal increases as $F_{s_{Amal}}$ increases from 2, 4, and 8 KHz by varying number of nodes from 4, 8, and 16 for Figure 9(a), and 2, 4, and 8 for Figure 9(b), respectively

accelerometer, and gyroscopes, respectively. We note that these numbers may significantly empower the attacker, as any additional information to the attacker is a gain when launching eavesdropping attacks, potentially posing serious threat to the victims. As an analogy, most people would feel uncomfortable or even threatened if 79% of their phone conversations are eavesdropped.

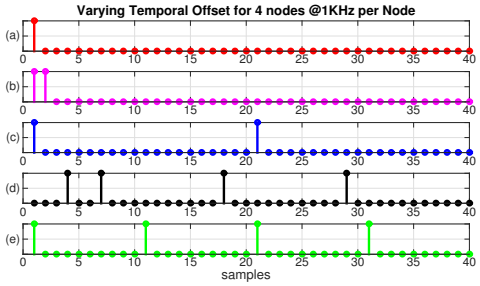


Figure 10: Varying temporal offsets from worst to best case sample scenarios for four nodes.

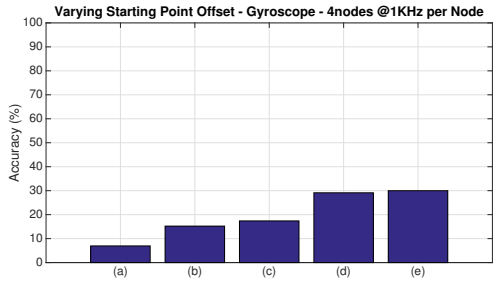


Figure 11: Comparison of recognition accuracy of four gyroscopes (F_s=1KHz each) sampled at different temporal offset.

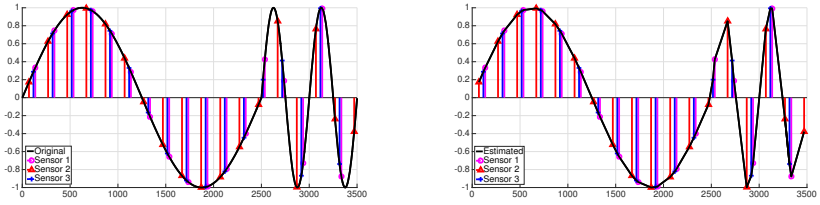
5.3.2 Practical Temporal Offset.
Recall that the aforementioned results assume a regular temporal offset, which inherently results in the best case scenario for the *PitchIn* attack. However, in reality, temporal offset may be randomly distributed among devices. We investigate this aspect by exploring how varying temporal offset affects recognition accuracy.

To provide an intuition, we provide five different temporal offsets of four nodes sampling different gyroscopes. Figure 10 illustrates pictorial representation of a spectrum of varying temporal offsets (i.e., sampling patterns) from the worst case to the best case scenario. (a) depicts the situation when all four nodes are sampling exactly at the same time (hence the worst case scenario). (b) and (c) depict the situations when two of the nodes are sampling at the same time. Specifically, (b) depicts an example where there is not too much information gain from the temporal offset due to samples being clustered. We note that (c) resembles the situation synonymous to when two nodes are sampling at an evenly distributed interval. (d) depicts the situation when four nodes are sampling at different times, but are not evenly distributed. Hence, the samples are more distributed, allowing larger temporal coverage. (e) depicts the situation when four nodes are sampling at an evenly distributed time (hence the best case scenario). We denote these as *Sample Scenarios (a)* through *(e)*.

Figure 11 depicts the recognition accuracy of (a) through (e) for four gyroscope sensors with F_s=1 KHz. We chose gyroscope to demonstrate the lower bound of recognition accuracy among the sensors (as seen from Figure 8). It is interesting to note that the recognition accuracy increases from (a) to (e), from 7% to 30%), which justifies the spectrum of varying temporal offsets from worst to best case scenario. Furthermore, we note that (c) yields roughly twice the accuracy of (a) and half of (e).

While scenarios (b), (c), and (d) are each single instances of temporal offset of these four sensors in between worst and best case scenarios (i.e., (a) and (e)), this example serves to demonstrate the trend of increasing recognition accuracy as temporal offset lies in between the two extremes.

We also present an example to provide an intuition of how "random" temporal offset still contributes to reasonable recognition



(a) Original signal sampled by sensors 1, 2, and 3 (b) Estimated signal from the sampled data

Figure 12: An example of Distributed TI-ADC and its effects when sensors 1, 2, and 3 are sampling the original signal with random temporal offset.

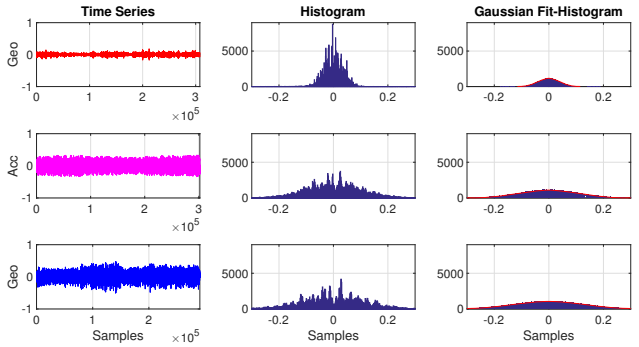


Figure 13: Inherent noise in time series of each sensor and the corresponding histogram and Gaussian fit. Data are collected from a quiet room.

accuracy by providing an example. The accuracy of the estimation depends on many factors including the frequency of the signal to sample, the number of nodes sampling, and the sampling frequency per node. This idea is illustrated in Figure 12.

Figure 12(a) displays a scenario where different sensors sample a sine wave which varies its frequency from 2 Hz to 10 Hz), with a constant sampling frequency of 25 Hz across the sensor nodes (Sensor 1, 2, and 3). The starting point of the nodes were not synchronized and were taken at random.

We demonstrate that even without synchronizing the nodes, the attacker gains enough information to estimate a sensible signal for certain portion of the original signal. Figure 12(b) depicts this idea, where the solid line shows the estimated signal after interleaving the sampled data from the sensors. Specifically, the 2 Hz portion of the sine wave can be estimated more closely than that of the 10 Hz portion, even though the three sensors did not sample with an evenly distributed temporal offset. This is intuitive as more points are sampled for the slower portion of the signal. However, the 10 Hz portion of the signal is not well estimated as shown from the same figure.

5.3.3 *Amalgam Signal Simulation.*

A realistic simulation of sensor node requires acknowledgment of the noise that are unique to each physical sensors. Using a Gaussian fit, we make an assumption that sensors of the same sensor modality has similar signal to noise ratio, and therefore, Gaussian noise of similar variance. In the aforementioned experiments, we sample ambient noise (in a quiet room) from each sensor to estimate the inherent noise distribution in each sensor modality. The values that are sampled are interpreted as a result of a Gaussian noise corrupting the audio signal. We create a generative model to model the noise characteristic of each sensor modality and then estimate the Gaussian fit of such profile. This profile is then used to create multiple instances of possible noise given a sensor. As we add this known noise to the signals we acquired, we simulate realistic sensor data. This process is repeated for all signals used in the present study. Figure 13 depicts this process.

6 PRACTICAL CONSIDERATIONS

This section presents practical considerations of *PitchIn*.

6.1 Time Synchronization

We note that the assumption of tight time synchronization made in the paper are only for the purpose of proof-of-concept experimentation but are not required for the general problem at hand for the attacker. In the experiment, we assumed the tight synchronization due to simplicity of fusing the aggregated sensor data collected from the network. However, in practice, even if the devices are not tightly synchronized, we are inspired by previous work in time interleaving ADCs (of local devices) that make use of a known reference signal to try to detect and correct timing mismatches or skews among signals sampled by different ADCs [17, 38]. While it is infeasible for an attacker in *PitchIn* to have such a reference signal, we claim that it is feasible for an attacker to perform a manual search (in a bruteforce manner) to shift and find optimal results. While this may be time consuming, it is certainly feasible due to the nature of offline attacks.

Furthermore, it is quite reasonable to assume a tight time synchronization among IoT devices in the near future due to many applications requiring high synchronization accuracy (e.g., sensor fusion, precise indoor localization, etc.) Many proposals and standards already propose sub-millisecond to microsecond accuracy [6, 20, 40]. Specifically, analogous to how synchronization using NTP is common today, we carefully speculate that a more accurate time synchronization protocols such as Precision Time

Protocol (PTP) may be prevalently used in the near future among the IoT devices, as we already find many open source libraries that support PTP protocol on even cheap devices like Arduino [6]. This range is sufficiently accurate to aid the attacker because *PitchIn* devices require sampling frequencies far less than 8 KHz per node, which translates to a minimum of 125 microsecond per sample, well above the sub-microsecond synchronization accuracy ranges.

6.2 Controlled Experiment Setup

Recall from Section 5.1 that the experiments are conducted with the speaker about a meter away from the co-located sensors. While the experiments are conducted under such controlled environment, we claim that such a setup still provides a practical use case as an exemplary scenario. As depicted in Figure 1, the abundance of sensors embedded in existing commercial products today (e.g., smart TV remote [16] and game controllers [45] depicted in Table 1) are often found co-located on a coffee table or sofa in a living room. It is not difficult to imagine that the sensor data may be collected in the near future by the manufacturers for user behavior analysis, as numerous TV (along with other device) manufacturers are notoriously known to have been collecting privacy sensitive information including viewing and searching data as well as speech from TVs [21, 27, 31]. This scenario demonstrates a strong yet potential case for the attacker, specifically illustrating the practicality of close proximity setup of sensors and the speaker. While we acknoledge that this is a generous scenario for the adversary, it provides an intuition of how potential attack may be carried out under favorable conditions for the attacker.

6.3 Amplification

As mentioned in Section 5.1, the sensor output are amplified in hardware using operational amplifiers (op-amps) before being interfaced to the Arduino's ADC. We note, however, that the hardware amplification reflects reality as many IoT devices are manufactured with circuitry that leverages hardware amplifiers for sensors [1]. In addition, many IoT devices use digital MEMS sensors, which already come equipped with op-amps within the MEMS circuitry [5].

6.4 Automating the Attack

An attacker may automate *PitchIn* attack by feeding in the results obtained by *PitchIn* to an existing Automatic Speech Recognition (ASR) engine. While we had conducted a preliminary experimentation with publicly available Speech Recognition Engine [49], the results were not satisfying, due to the fact that the ASR is trained with microphone data. From consultations with speech recognition experts, we are hopeful that if an attacker trains an ASR with non-acoustic sensors with varying sampling rate, it would most likely yield a relatively high accuracies.

7 CONCLUSION AND FUTURE WORK

We present *PitchIn* to demonstrate a feasibility of fusing non-acoustic sensors (e.g., geophone, accelerometer, gyroscope) to reconstruct intelligible speech signals using various speech processing techniques. *PitchIn* minimizes per-node sampling frequency by leveraging a distributed Time Interleaved Analog-Digital-Converter (TI-ADC) across network of sensor devices. We conduct user studies to evaluate the intelligibility of the reconstructed signals. *PitchIn* achieves speech recognition accuracy ranging from 79% to 35% depending on the sensor modalities, sampling rate, and number of nodes.

We explore the *PitchIn* signal reconstruction attack by exploring the metrics from the adversaries perspectives. While further theoretical and empirical study on the impact of signal quality from TI-ADC would provide interesting results, we delay this to future work. We also find many potential extensions to *PitchIn*, including increasing scalability of *PitchIn* attack by leveraging automated speech recognition engines to create a fully automated remote eavesdropping tool. Through this work, we hope to highlight a potential problem of pervasive IoT devices that may be densely deployed in our homes and offices, surpassing the known and obvious risks. While other researchers have demonstrated the feasibility of capturing voice signals for non-acoustic sensors, we illustrate that a naive solution of merely reducing the sampling rate per node may be insufficient to thwart against the above problems. Rather, we hint at a new paradigm of a room-level security policy to mandate an upper-bound of a cumulative sampling rate across devices that is low enough to sufficiently thwart such attacks.

A T-TEST RESULTS

We show the significance of evaluation results reported in Section 5 using paired t-tests. Analysis of Variance (ANOVA) on these evaluation showed significance.

	F_s		p-value			
	KHz		Geo	Acc	Gyr	Mic
Fig.8	1	2	.86	<0.001	.41	<.001
	2	4	<.001	<.001	<.001	<.001
	4	8	<.001	<.001	<.001	<.001
	1	8	<.001	<.001	<.001	<.001
Fig.9(a)	2	4	<.001	<.001	.43	N/A
	4	8	.24	<.001	.43	N/A
	2	8	<.001	<.001	.22	N/A
Fig.9(b)	2	4	<.001	<.001	.05	N/A
	4	8	<.001	<.001	.28	N/A
	2	8	<.001	<.001	<.001	N/A

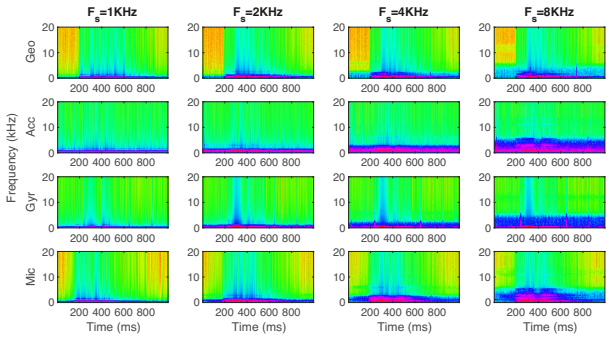
Table 3: Paired t-test for Figures 8 and 9

Comparison Pair		p-value
Pattern (a)	Pattern (b)	.006
Pattern (c)	Pattern (b)	.52
Pattern (e)	Pattern (b)	<.001
Pattern (a)	Pattern (d)	<.001
Pattern (c)	Pattern (d)	.003
Pattern (e)	Pattern (d)	.84

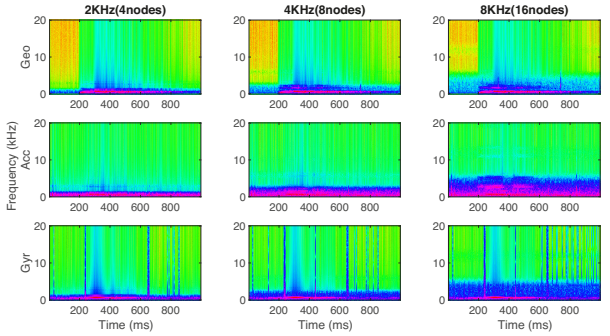
Table 4: Paired t-test for Figure 11

B SPECTROGRAM

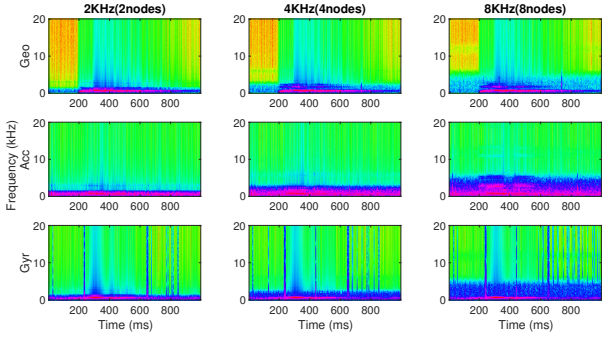
We present spectrograms depicting corresponding signals represented in Figures 8 and 9.



(a) Spectrogram of signals in Figure 8



(b) Spectrogram of signals in Figure 9(a)



(c) Spectrogram of signals in Figure 9(b)

Figure 14: Spectrogram of signals evaluated in Figure 8 and 9. We strongly advise the readers to view this figure in color, and to watch the corresponding video clips at http://mews.sv.cmu.edu/research/pitchin/.

ACKNOWLEDGMENTS

We would like to thank our shepherd, Rui Tan, and anonymous reviewers for their valuable comments. This research was supported in part by the National Science Foundation under grant CNS-1645759. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of CMU, NSF, or the U.S. Government or any of its agencies.

REFERENCES

[1] *ADXL103/ADXL203 Datasheet.* Analog Devices Data Sheet.
[2] *Arduino Ethernet Shield.* https://www.arduino.cc/en/Main/ArduinoEthernetShield.
[3] *Arduino FAT16/FAT32 Library.* https://github.com/greiman/SdFat.
[4] *Arduino/Genuino UNO.* https://www.arduino.cc/en/Main/arduinoBoardUno.
[5] *LIS331DLH: MEMS digital output motion sensor ultra low-power high performance 3-axes "nano" accelerometer.* STMicroelectronics Datasheet.
[6] *PTPd.* http://ptpd.sourceforge.net/.
[7] Spencer Ackerman and James Ball. 2014. *Optic Nerve: millions of Yahoo webcam images intercepted by GCHQ.* http://www.theguardian.com/world/2014/feb/27/gchq-nsa-webcam-images-internet-yahoo.
[8] Amazon. 2015. *Amazon Mechanical Truk.* https://www.mturk.com/mturk/welcome.
[9] Tali Arbel. 2016. *Verizon Business Accounts Hacked.* http://www.theguardian.com/world/2014/feb/27/gchq-nsa-webcam-images-internet-yahoo.
[10] Atmel. *Atmel 8-bit AVR Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash.* Atmel. http://www.atmel.com/Images/doc8161.pdf.
[11] Audacity. *DC Offset.* http://manual.audacityteam.org/o/man/dc_offset.html.
[12] S. Butterworth. 1930. On the Theory of Filter Amplifiers. *Experimental Wireless and the Wireless Engineer* 7 (1930), 536–541.
[13] IO Sensor Nederland b.v. *SM-24 Geophone Element.* http://cdn.sparkfun.com/datasheets/Sensors/Accelerometers/SM-24%20Brochure.pdf.
[14] Abe Davis, Michael Rubinstein, Neal Wadhwa, Gautham Mysore, Fredo Durand, and William T. Freeman. 2014. The Visual Microphone: Passive Recovery of Sound from Video. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 33, 4 (2014), 79:1–79:10.
[15] Analog Devices. *ADXL335 Datasheet.* http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL335.pdf.
[16] Analog Devices. *LG Smart TV platform guide.* http://docs.madewithmarmalade.com/display/MD/LG+Smart+TV+platform+guide.
[17] M. El-Chammas and B. Murmann. 2011. A 12-GS/s 81-mW 5-bit Time-Interleaved Flash ADC With Background Timing Skew Calibration. *IEEE Journal of Solid-State Circuits* 46, 4 (April 2011), 838–847. DOI: http://dx.doi.org/10.1109/JSSC.2011.2108125
[18] Chuck Goudie and Ross Weidner. 2015. *Home hackers: Digital invaders a threat to your house.* http://abc7chicago.com/technology/home-hackers-digital-invaders-a-threat-to-your-house/515520/.
[19] Jun Han, Emmanuel Owusu, Le T. Nguyen, Adrian Perrig, and Joy Zhang. 2012. ACComplice: Location inference using accelerometers on smartphones. In *Communication Systems and Networks (COMSNETS).* DOI: http://dx.doi.org/10.1109/COMSNETS.2012.6151305
[20] T. Hao, R. Zhou, G. Xing, and M. Mutka. 2011. WizSync: Exploiting Wi-Fi Infrastructure for Clock Synchronization in Wireless Sensor Networks. In *Real-Time Systems Symposium (RTSS), 2011 IEEE 32nd.*
[21] Chris Heinonen. 2015. *Your Privacy, Your Devices, and You.* http://thewirecutter.com/blog/your-privacy-your-devices-and-you/.
[22] Texas Instruments. 2011. *ADC081000, ADC08D1000: Interleaving ADCs for Higher Sample Rates.* Texas Instruments Whitepaper SNAA111.
[23] Texas Instruments. 2014. *LMV3xx Low-Voltage Rail-to-Rail Output Operational Amplifiers.*
[24] Jungsuk Kim and Ian Lane. 2014. Accelerating large vocabulary continuous speech recognition on heterogeneous CPU-GPU platforms. In *Acoustics, Speech and Signal Processing (ICASSP), IEEE.* DOI: http://dx.doi.org/10.1109/ICASSP.2014.6854209
[25] Sukun Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. 2007. Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks. In *Information Processing in Sensor Networks (IPSN).* DOI: http://dx.doi.org/10.1109/IPSN.2007.4379685
[26] Wise Lab. *FireFly3.* "http://wise.ece.cmu.edu/redmine/projects/firefly/wiki/FireFly3_x2".
[27] Sapna Maheshwari. 2017. *Is Your Vizio Television Spying on You? What to Know.* https://nyti.ms/2kKRKpS.
[28] Rahul Mangharam, Anthony Rowe, and Raj Rajkumar. 2007. FireFly: A Cross-layer Platform for Real-time Embedded Wireless Networks. *Real-Time Syst.* 37, 3 (Dec. 2007), 183–231. DOI: http://dx.doi.org/10.1007/s11241-007-9028-z
[29] Chung-Tse Mar, Mat Hans, Mark Smith, Tajana Simunic, and Ronald Schafer. 2002. *A High-Quality, Energy Optimized, Real-Time Sampling Rate Conversion Library for the StrongARM Microprocessor.* Technical Report HPL-2002-159.

[30] Philip Marquardt, Arunabh Verma, Henry Carter, and Patrick Traynor. 2011. (Sp)iPhone: Decoding Vibrations from Nearby Keyboards Using Mobile Phone Accelerometers. In *Conference on Computer and Communications Security (CCS)*. ACM, New York, NY, USA. DOI: http://dx.doi.org/10.1145/2046707.2046771

[31] Chris Matyszczyk. 2015. *Samsung's warning: Our Smart TVs record your living room chatter.* https://www.cnet.com/news/samsungs-warning-our-smart-tvs-record-your-living-room-chatter/.

[32] Yan Michalevsky, Dan Boneh, and Gabi Nakibly. 2014. Gyrophone: Recognizing Speech from Gyroscope Signals. In *23rd USENIX Security Symposium (USENIX Security 14)*. USENIX Association, San Diego, CA, 1053–1067. https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/michalevsky

[33] Steven Norton. 2015. *Internet of Things Market to Reach $1.7 Trillion by 2020.* http://blogs.wsj.com/cio/2015/06/02/internet-of-things-market-to-reach-1-7-trillion-by-2020-idc/.

[34] Emmanuel Owusu, Jun Han, Sauvik Das, Adrian Perrig, and Joy Zhang. 2012. ACCessory: Password Inference Using Accelerometers on Smartphones *(HotMobile)*. ACM, New York, NY, USA. DOI: http://dx.doi.org/10.1145/2162081.2162095

[35] Shijia Pan, Mostafa Mirshekari, Pei Zhang, and Hae Young Noh. 2016. Occupant traffic estimation through structural vibration sensing. (2016).

[36] Shijia Pan, Ningning Wang, Yuqiu Qian, Irem Velibeyoglu, Hae Young Noh, and Pei Zhang. 2015. Indoor Person Identification Through Footstep Induced Structural Vibration *(HotMobile)*. ACM, New York, NY, USA. DOI: http://dx.doi.org/10.1145/2699343.2699364

[37] Pery Pearson. *Sound Sampling.* http://www.hitl.washington.edu/projects/knowledge_base/virtual-worlds/EVE/I.B.3.a.SoundSampling.html.

[38] B. Razavi. 2012. Problem of timing mismatch in interleaved ADCs. In *Proceedings of the IEEE 2012 Custom Integrated Circuits Conference*. 1–8. DOI: http://dx.doi.org/10.1109/CICC.2012.6330655

[39] BBC Research. *Global Markets and Technologies for Sensors.* "http://www.bccresearch.com/market-research/instrumentation-and-sensors/sensors-tech-markets-report-ias006g.html".

[40] Thomas Schmid, Prabal Dutta, and Mani B. Srivastava. 2010. High-resolution, Low-power Time Synchronization an Oxymoron No More. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '10)*. ACM.

[41] Claude E. Shannon. 1949. Communication in the Presence of Noise. *Proceedings of the IRE* (1949). DOI: http://dx.doi.org/10.1109/JRPROC.1949.232969

[42] SkyPaw. *SkyPaw Multi Measures – Decibel.* http://www.skypaw.com/apps/multimeasures/.

[43] Yunmok Son, Hocheol Shin, Dongkwan Kim, Youngseok Park, Juhwan Noh, Kibum Choi, Jungwoo Choi, and Yongdae Kim. 2015. Rocking Drones with Intentional Sound Noise on Gyroscopic Sensors. In *USENIX Security*.

[44] STMicroelectronics. *LPY503AL Datasheet.* https://www.sparkfun.com/datasheets/Sensors/IMU/lpy503al.pdf.

[45] Daniel Turner. 2007. *Hack: The Nintendo Wii.* http://www.technologyreview.com/hack/408183/hack-the-nintendo-wii/.

[46] Hasan S. Ulusoy, Erol Kalkan, Jon Peter B. Fletcher, Paul Friberg, W. K. Leith, and Krishna Banga. 2012. Design and implementation of a structural health monitoring and alerting system for hospital buildings in the United States. In *World Conference on Earthquake Engineering*.

[47] K. J. Van Engen, J. E. Phelps, R. Smiljanic, and B. Chandrasekaran. 2014. Enhancing speech intelligibility: interactions among context, modality, speech style, and masker. *Experimental Wireless and the Wireless Engineer* (2014).

[48] WaspMote. *Events 2.0 Technical Guide.* http://www.libelium.com/downloads/documentation/events-sensor-board_2.0.pdf.

[49] Anthony Zhang. *Speech Recognition.* https://github.com/Uberi/speech_recognition.

[50] Li Zhang, Parth H. Pathak, Muchen Wu, Yixin Zhao, and Prasant Mohapatra. 2015. AccelWord: Energy Efficient Hotword Detection Through Accelerometer *(ACM MobiSys)*. DOI: http://dx.doi.org/10.1145/2742647.2742658