Large-scale audio eavesdropping using smartphones

Dimitrije Erdeljan
Trinity College

UNIVERSITY OF CAMBRIDGE

A dissertation submitted to the University of Cambridge in partial fulfilment of the requirements for the Part III of the Computer Science Tripos

University of Cambridge
Computer Laboratory
William Gates Building
15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom

Email: de298@cam.ac.uk

May 31, 2019

Declaration

I Dimitrije Erdeljan of Trinity College, being a candidate for Part III of the Computer Science Tripos, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

Total word count: 11111

Signed:

Date:

Abstract

The goal of this project was to evaluate the advantages an eavesdropper would gain from access to a large number of compromised mobile devices, with a focus on metrics relevant in a large-scale deployment. A prototype of such a system was implemented, consisting of an experimental setup used to gather data and a signal processing pipeline which produces the combined recording. The output is computed by aligning and summing the recordings, with offsets chosen to create constructive interference and amplify sound coming from the target under surveillance.

The implementation was evaluated on three sets of recorded speech samples, made using an experimental setup consisting of three mobile devices. In all three sets, the combined output resulted in a higher signal-to-noise ratio than the highest-SNR input. The gain depends on the positions of the devices, ranging from (2.0 ± 0.3) dB to (3.3 ± 0.3) dB.

Transcribing the recordings using the DeepSpeech speech-to-text engine has shown that it is unsuitable for use with smartphone recordings, as it has a word error rate above 95 % and produces no output for 25 % of the unprocessed recordings. Performance on the combined output is improved, but not practically useful: the error rate is improved by 0.8 %, and no speech is detected 10 % of the samples.

Smaller-scale evaluation using the Google speech-to-text API has shown that it performs significantly better on the recorded samples, and that the error rate is improved by integrating the recordings in all three datasets, with the improvement ranging from 3 % to 13 %.

Contents

1 Introduction **1**

2 Background **5**
2.1 Overview 5
2.2 Related work 7
2.3 Challenges 8
2.4 Speech-to-text 9
2.4.1 Engine and dataset 9
2.4.2 Word error rate 10

3 Theoretical model **11**
3.1 Upper bound on processing gain 11
3.2 Effect of calibration errors on gain 13
3.3 Speech-to-text performance in noisy conditions 16

4 Design and Implementation **19**
4.1 Recording setup 19
4.1.1 Recording app 20
4.1.2 Servers 21
4.2 Calibration 22
4.2.1 Calibration tone 23
4.2.2 Offsets 23
4.2.3 Signal-to-noise ratio 27
4.3 Combining recordings 28

5 Evaluation **31**
5.1 Setup 31
5.2 Offset calibration accuracy 32
5.2.1 Results 33
5.3 Improvement in signal-to-noise ratio 37
5.3.1 Results 38

5.4 Speech-to-text word error rate 41
5.4.1 Results 42

6 Summary and Conclusions **45**
6.1 Future work 46

List of Figures

3.1 Heat map of the upper bound on achievable gain for a café-like scenario. Microphones, shown as white circles, are randomly scattered on tables, shown as dotted circles. Dashed lines represent the boundary at which the gain is 3 dB. 12
3.2 Heat map of the upper bound on achievable gain for four microphones, shown as white circles, placed in the corners of a room. Dashed lines represent the boundary at which the gain is 3 dB. 13
3.3 Plot of lost processing gain as a function of offset error, for a two-microphone setup. 14
3.4 Plot of lost processing gain as a function of the standard deviation of offset error, for a six-microphone setup. 15
3.5 Transcription word error rate as a function of the input signal-to-noise ratio. The dashed line shows the baseline WER (without noise). 17

5.1 Distribution of offset errors near the zero-offset (not showing outliers). 35
5.2 Distribution of offset errors near the zero-offset (not showing outliers). 36
5.3 Distribution of gains obtained by combining recordings, compared to the baseline of the best microphone available, for each of the three evaluation runs. 39
5.4 Box plot of available gain when using the best k microphones. The red marker indicates the median, with the box covering values from the 25^{th} to the 75^{th} percentile. Outliers further than three standard deviations from the median are displayed as red crosses. 40

List of Tables

5.1 Distribution of differences in offsets of the two microphones for three sets of recorded samples. 34
5.2 Distribution of offset errors aggregated across all runs. Each entry includes values within five samples from the offset. Entries with less than three occurrences are omitted. 34
5.3 Distribution of offset errors aggregated across all runs. Each entry includes values within five samples from the offset. . . . 35
5.4 Gain obtained by combining the recorded samples for each run, and the upper bound for the improvement (computed as the sum of SNRs of recordings). 38
5.5 Performance of the DeepSpeech speech-to-text engine on the recorded samples for the microphone with the highest SNR and the combined output, showing the percentage of inputs for which no speech was detected, and the word error rate. . . 43
5.6 Word error rate of transcriptions made using the Google speech-to-text engine when processing the recording with the hignest SNR and the combined output. 43

Chapter 1

Introduction

Consider the privacy risks of having one's phone compromised – an attacker with such access would easily be able to eavesdrop on any conversation, as long as the background noise is low enough. The limited quality of the microphones, however, makes them significantly less suitable in a noisy environment, where little can be heard on a recording. Such environments, however, usually do not only contain a single phone. For example, imagine a crowded café, where we can often see a smartphone on almost every table. Given the large attack surface of these devices, there is a possibility that an attacker might be able to take control of them – for example, by exploiting a vulnerability such as a recently-reported buffer overflow in WhatsApp, which could be triggered remotely by calling the target.

The goal of this project is to evaluate whether an eavesdropper with access to a number of compromised devices can integrate their recordings and amplify the sound coming from a target they are interested in, with particular focus on large-scale audio capture and transcription. It is based on a principle similar to a phased array of antennas: recordings are aligned based on the time sound takes to travel from the target to the device, and then summed to produce an amplified output. As a baseline, the gain from such processing is compared with the highest-quality individual recording the attacker would have access to, chosen as the one with the highest signal-to-noise ratio.

To provide an example of a scenario in which such an attack might occur, consider the crowded café mentioned above. While having access to a single visitor's phone is unlikely to result in a useful recording, an attacker who can compromise devices at a large scale could combine their microphones and listen in a chosen conversation (or, with sufficient computational resources, to all conversations in the café). This scenario is particularly interesting since it can be deployed at scale, as it requires no physical presence by the eavesdropper. Compromised phones would effectively form a large-scale surveillance system, and with automated speech recognition technology widely available, potentially automate searching for conversations of interest.

As a second motivating example, consider a scenario where devices in one's home are compromised. While such a situation provides the attacker with fewer devices, it is still likely that they would have access to several smartphones or tablets. Additionally, they could benefit from other Internet-connected household devices with recording capabilities, with voice-activated home assistants, such as Google Home and Amazon Echo, becoming increasingly common. With the recent discovery of a previously undisclosed microphone in Google's Nest Secure hub, the privacy-related implications of these devices are becoming a topic of discussion, and voice assistants might be yet another source of microphones that could be exploited by an attacker.

The remainder of this dissertation starts with Chapter 2, which provides an outline of the system that was produced and evaluated, a brief overview of related work and a summary of the challenges involved in working with smartphone microphones.

In Chapter 3, a theoretical model for the system is presented, giving upper bounds for the gain that can be obtained by processing the output of multiple microphones. A simulation of the effects of calibration errors on the gain shows that offsets used to align signals must be accurate within several samples, which demonstrates the necessity of an automatic calibration procedure. Finally, an experimental evaluation of the behaviour of a speech-to-text engine provides an estimate of expected impact of improvements of the signal-to-noise ratio on transcription accuracy.

Chapter 4 describes the implementation of a simulated smartphone-based surveillance system, covering the architecture of the setup used to record data, the algorithms used for calibration, and the processing done to combine the recordings.

Experimental results of the system evaluation are presented in Chapter 5 on recorded data, covering the calibration accuracy, processing gain and improvement in speech-to-text accuracy.

The results demonstrate that an attacker with access to three devices can combine their recordings to produce an output whose signal-to-noise ratio is greater than the SNR of the best microphone available. The gain depends on the exact layout of the devices, ranging from (2.0 ± 0.3) dB in the worst-performing setup to (3.3 ± 0.3) dB in the best.

Evaluating the word error rate of the DeepSpeech speech-to-text engine on recorded speech before and after processing shows that the engine is unsuitable for processing recordings made using smartphone microphones, failing to recognise speech in 25 % of the samples when using the output of a single microphone. Combining the recordings reduces the number of unrecognised samples to 10 %. The error rate is reduced by 0.8 %, but is still prohibitively high, at above 95 %.

The behaviour of the system was further evaluated using the Google speech-to-text API, which provides better performance in noisy conditions. The processed audio files result in a noticeably lower error rate than the transcriptions made using the single lowest-noise recording, with the improvement varying from a 13 % drop in the word error rate for the most favourable setup, to 3 % for the least favourable.

Finally, it should be noted that the gain obtained by combining multiple recordings is not the only advantage the attacker gains by a large number of available devices. As the results have shown, even in a setup where all devices are placed at approximately the same distance from the target, the noise levels in their recordings can vary substantially. Therefore, even the capability to select the most suitable device, instead of relying on a single

compromised smartphone, can significantly improve the capabilities of an eavesdropper.

Chapter 2

Background

This chapter will provide a summary of the background information relevant to the rest of this dissertation. A brief description of the system implemented and used for evaluation is given in the first section, followed by an overview of the related literature. The third section lists the challenges encountered when working with smartphone microphones and their implications on the system design. The final section describes the speech-to-text engine and metric used for evaluation.

2.1 Overview

This section will give an overview of the hardware and software components involved in a surveillance attempt using the approach evaluated in this project. The attack starts with a number of compromised devices turning on their microphones, recording their surroundings, and uploading the results to a centralised location for further processing.

After the recordings are made, the first stage of processing is calibration, which computes the parameters used to combine them into the amplified output. The first set of required parameters are the offsets at which the sound from the target appears in the recording, caused by the different travel times to the microphones. The offsets are used to align the recordings, creating

constructive interference in their sum.

After the offsets, the second set of parameters that is computed are the signal-to-noise ratios of the recordings. These are used as weighting coefficients when combining the inputs, emphasising those with lower noise levels.

It should be noted that the offsets can vary between recording sessions even if the setup is not moved, due to small delays between the start of recording on the devices. Therefore, the calibration procedure must be run for each set of recorded samples, and repeated if the session is interrupted.

In this project, two approaches to calibration were tested, one based on a known calibration tone coming from the target, and another that only uses already-existing sound such as the target's speech. The motivating scenario for the first approach was one where the attacker uses the speakers of the compromised phones to infer the positions of devices in the room and simplify calibration. Evaluation, however, shows that the played tone is distorted significantly by the phone speakers, and that the second approach is significantly more reliable.

After calibration, the recordings are combined into a single output audio signal, in which the sound coming from the target is amplified. The computed offsets are used to align the recordings to produce constructive interference of the copies of the source signal. The output is computed as the weighted sum of the recordings, favouring those with a higher signal-to-noise ratio.

Finally, an adversary deploying a surveillance system at a large scale would likely be interested in further processing of the output audio to identify targets of interest and reduce the workload necessary to investigate all recordings. To evaluate the advantage they would gain from combining recordings, a speech-to-text engine was used to transcribe the output and compare the result to transcriptions of the unprocessed inputs.

2.2 Related work

The usage of an array of microphones to improve the quality of recorded speech is a well-studied area, with applications in a variety of devices, from teleconferencing systems to hearing aids [1] [2]. An overview of the field can be found in a 2017 review paper by Gannot et al, which classifies the approaches into two categories: those based on microphone array parameter estimation, and statistical methods based on blind source separation [3].

The time delay estimation approach used in this project, based on generalised cross-correlation, was first proposed in 1976 [4]. An evaluation of its behaviour in the presence of reverberation shows that the ML estimator, while optimal for Gaussian noise, exhibits "dramatic performance degradation in reverberant rooms", with other estimators such as PHAT demonstrating better behaviour [5].

While they were originally developed as an approach to localising a single source, methods based on the time difference of arrival (TDOA) have been shown to be a viable approach for tracking multiple speakers [6]. With a high-resolution localisation system, short bursts of time in which only a single speaker is active can be used to separate the sources.

More recently, there has been interest in ad-hoc microphone systems, in which the positions are not known prior to recording. TDOA-based approaches have been shown to be a viable method for inferring microphone positions in such scenarios. Using a large array (consisting of over fifteen microphones), positions can be estimated with sub-millimetre accuracy, while smaller arrays have been shown to be accurate to a few centimetres [7] [8].

Ad-hoc arrays of smartphones have been investigated as well. Hennecke and Fink demonstrate a time-delay approach that can be used to localise devices in a room, with a positional error around 7 cm [9]. A study of beamforming using smartphone ad-hoc arrays identifies microphone directionality as a significant challenge, especially at higher frequencies (above 1 kHz) [10].

Apart from methods based on time delay estimation, an important approach

to speech improvement is blind source separation [11]. This is a statistical method, which does not use a model of sound propagating from multiple sources, but relies on an assumption of statistical independence of the source signals. It has been demonstrated as a viable approach for separating multiple speakers in a reverberant room, with the recordings made using directional microphones or smartphones [12] [13].

2.3 Challenges

This section will briefly outline the main challenges related to signal processing encountered in this project. The recordings used in the project are made using off-the-shelf smartphone microphones, which are not designed to record sound from a distant source. Therefore, their characteristics require consideration that would not be necessary if purpose-built equipment was used.

Since the design of microphones in smartphones is optimised for space constraints and a primary use in voice calls, the quality of their recordings is limited. This primarily manifests in recordings that are noisier than those that could be obtained using non-embedded microphones. Further, the frequency response of the microphones is far from uniform. Both low and very high frequencies are attenuated significantly, limiting the range which can be used for calibration (if it depends on the presence of a predefined tone).

Another challenge a smartphone-based eavesdropping system is faced with is microphone directionality. In a setup where we cannot control the direction in which the phones are facing (since the attacker has no physical presence), it would be best for the microphones to be omnidirectional, and therefore usable regardless of the location of the target we are listening to. In practice, smartphone microphones are highly directional, with the received signal power dropping significantly (in some cases, over 10 dB) if they are facing away from the source.

Even if the attacker could choose the positions of the devices, directionality

would still be a significant factor. Since most smartphones have microphones on opposite ends, one of the two will be facing away from the target and will underperform compared to the one facing the target directly. Therefore, using both microphones on each phone does not double the effective number of devices, as the gain will be lower than what is obtainable using twice as many smartphones with one microphone each.

The final obstacle is not due to hardware, but comes from the behaviour of sound waves – since the waves reflect off walls and other objects, the recordings will contain multiple copies of all sounds in the room. In most cases, these echoes are quieter than the sound travelling via the direct path, and can safely be ignored. They can, however, introduce false positives in the calibration procedure, which should be ignored to avoid using a weaker reflected copy of the signal we are interested in. On the other hand, this effect can be beneficial – if a microphone is facing away from the source, but is near a wall, the reflected path can result in a louder recording than the direct one, since the sound wave will arrive facing the microphone instead of being shielded by the device body.

2.4 Speech-to-text

As a part of the evaluation, this project uses a speech-to-text engine to compare the quality of transcriptions before and after processing. This section will briefly outline the engine and dataset used for the evaluation, and define word error rate, the metric used to evaluate the transcriptions.

2.4.1 Engine and dataset

The engine used for evaluation is the DeepSpeech speech-to-text engine developed by Mozilla, based on a recurrent neural network model developed by Baidu Research [14]. The Mozilla-provided "deepspeech-0.4.1" pre-trained model was used, since the project does not require any specialised training.

It should be noted that the engine was trained using audio sampled at 16 kHz,

while the smartphone recordings were made using a 44.1 kHz sampling rate. Therefore, all recordings were downsampled before transcription.

The LibriSpeech automated speech recognition corpus was used as a source of labeled speech samples. This is a dataset consisting of labelled audio files, generated from public-domain audiobooks [15]. Each file contains a single English sentence, and is paired with a text file containing its transcription. The dataset is separated into a training and test subset, and both are further split into "clean" and "other" speech. The "test-clean" dataset, which contains over five hours of speech across 2620 samples, was used to evaluate the project.

It should be noted that LibriSpeech is the dataset used to develop the DeepSpeech model used by the engine. During this process, the "test-clean" samples were used as test data to evaluate the model, and this part of the LibriSpeech corpus does not contain the samples used during the model's training. On this dataset, the engine has a word error rate of 9 %, which provides a lower bound for the project's evaluation.

2.4.2 Word error rate

Word error rate (WER) is a common metric used to evaluate speech recognition systems. It can be informally viewed as the normalised number of errors made in the transcription of a sentence, but it is not a "true" error rate – while the WER for a completely correct transcription is 0, it is not bounded above and can be greater than 100 %.

To compute the word error rate for a ground truth sentence S and its transcription T, the two sentences are first aligned. The alignment is chosen to minimise the edit distance d, defined as the total number of word insertions, deletions and substitutions required to transform S into T. The error rate is then calculated as $W = \frac{d}{|S|}$.

As an example, consider the sentence "This engine works" and a transcription "This is engine words". The edit distance is $d = 2$ (insertion of "is" and substitution of "works" with "words"), resulting in a word error rate of $\frac{2}{3}$.

Chapter 3

Theoretical model

This chapter will present models of individual components of the system presented in this dissertation, with simulations which provide upper bounds for the performance that can be obtained.

Firstly, a simplified model of sound propagation and microphones is shown, which gives a bound on the gain that the attacker can obtain in the scenarios presented previously. The second section describes a simulation of the effects of errors in offsets computed during calibration, motivating the need for automatic calibration. Finally, an experimental evaluation of the effect of noise on transcriptions by the DeepSpeech engine is presented.

3.1 Upper bound on processing gain

The aim of this simulation is to compute an upper bound on the expected gain we can obtain using a system that combines microphone recordings in several scenarios. For a given set of positions where microphones are placed, it computes the power of the output signal if a source was placed in any point in space. The result is expressed as the gain obtained over using the closest microphone, computed as the logarithm of the ratio of the signals' powers.

This simulation makes several simplifying assumptions, and it therefore only

provides an upper bound for the gain. In particular, sound is modelled as a wave whose power is inversely proportional to distance travelled (assuming that far-field approximations hold at all distances), and ignore all reflections. Further, all microphones are assumed to be omnidirectional.

In the following figures, the positions of all microphones are shown as white circles. For convenience, the boundary at which the gain is 3 dB is shown as a dashed line.

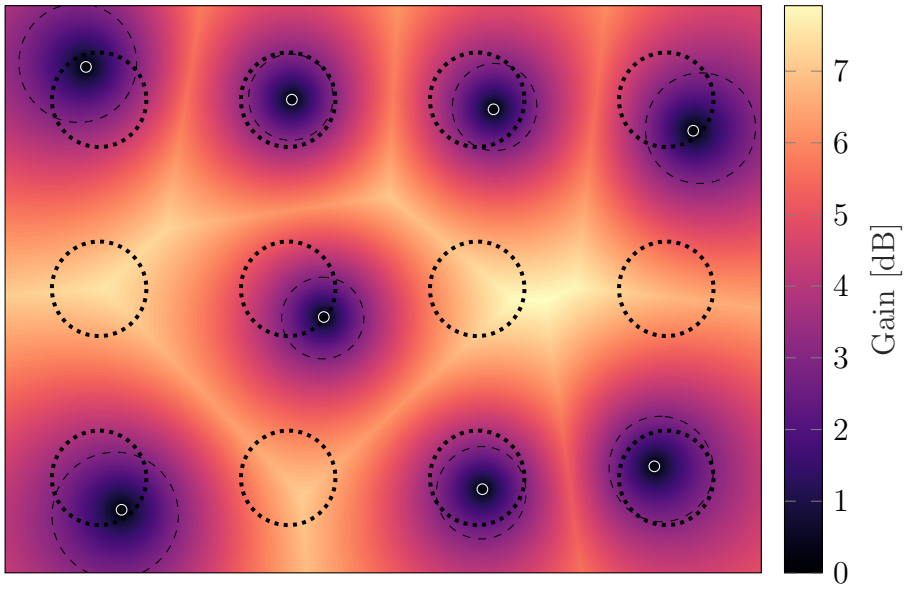


Figure 3.1 – Heat map of the upper bound on achievable gain for a café-like scenario. Microphones, shown as white circles, are randomly scattered on tables, shown as dotted circles. Dashed lines represent the boundary at which the gain is 3 dB.

Figure 3.1 presents a model of the "crowded café" scenario described in the Introduction. Recording devices are randomly placed on circles representing tables (shown as dotted lines). The gain at the edges of tables, around 3 dB, provides us with an upper bound on the gain the attacker could expect if the target's phone is one of the compromised devices. While this is an improvement over using a single device, it is sensitive to calibration errors, and would likely be lower in a practical implementation. More significantly, the simulation shows that the attacker can use the recordings from tables

with compromised devices to listen to conversations at a table where they do not have a microphone available – the gain in the part of the room not directly adjacent to a table is between 5 dB and 8 dB.

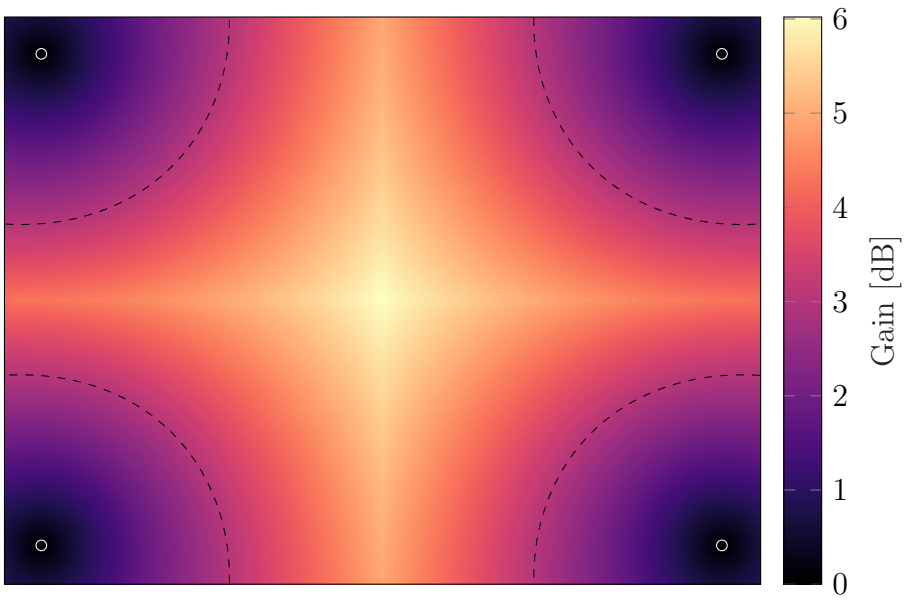


Figure 3.2 – Heat map of the upper bound on achievable gain for four microphones, shown as white circles, placed in the corners of a room. Dashed lines represent the boundary at which the gain is 3 dB.

The simulation shown in Figure 3.2 demonstrates the coverage obtained by placing microphones in the corners of a room. While this is a somewhat unlikely set of positions for smartphones, such a scenario could result from compromise of other household devices such as voice-activated assistants. We can see that there is little improvement near the corners of the room, where the speaker is close enough to one of the microphones to make the rest redundant. Combining the four, however, results in a gain greater than 3 dB in most of the room, with the highest gain being 6 dB in the centre.

3.2 Effect of calibration errors on gain

The goal of the second simulation is to estimate the losses resulting from calibration errors, and therefore provide an idea of how accurate the calibration

must be. It models the stage of calibration outputting the offsets at which the target signal occurs in the inputs, and investigates the effect of errors in the offsets on the final combined signal.

Here, we assume that we are recording the sound from a single source, which is playing a sine wave of constant frequency. The simulation computes the loss of gain of the output signal, compared to the best output possible (which would be obtained if there was no error). The loss is expressed in decibels, as the logarithm of the ratio of powers of the miscalibrated and error-free outputs.

To provide an upper bound on the loss, all recordings are assumed to have the same signal-to-noise ratio. Sound is sampled at 44.1 kHz, the rate used to capture audio in the rest of the project.

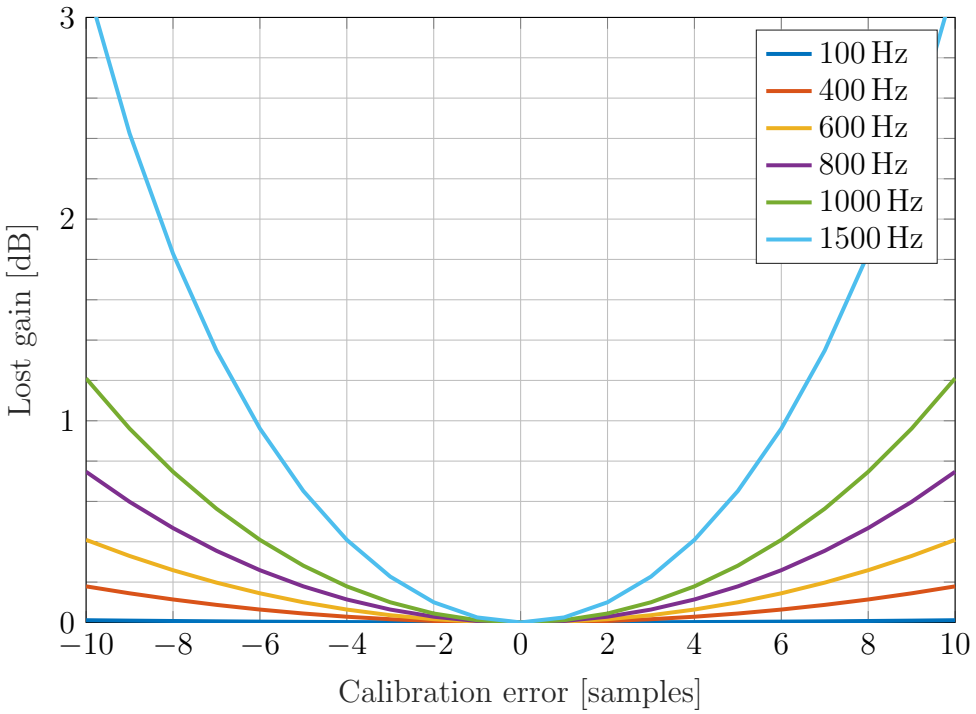


Figure 3.3 – Plot of lost processing gain as a function of offset error, for a two-microphone setup.

Figure 3.3 shows the loss of gain when two microphones are used as a function of the calibration error.

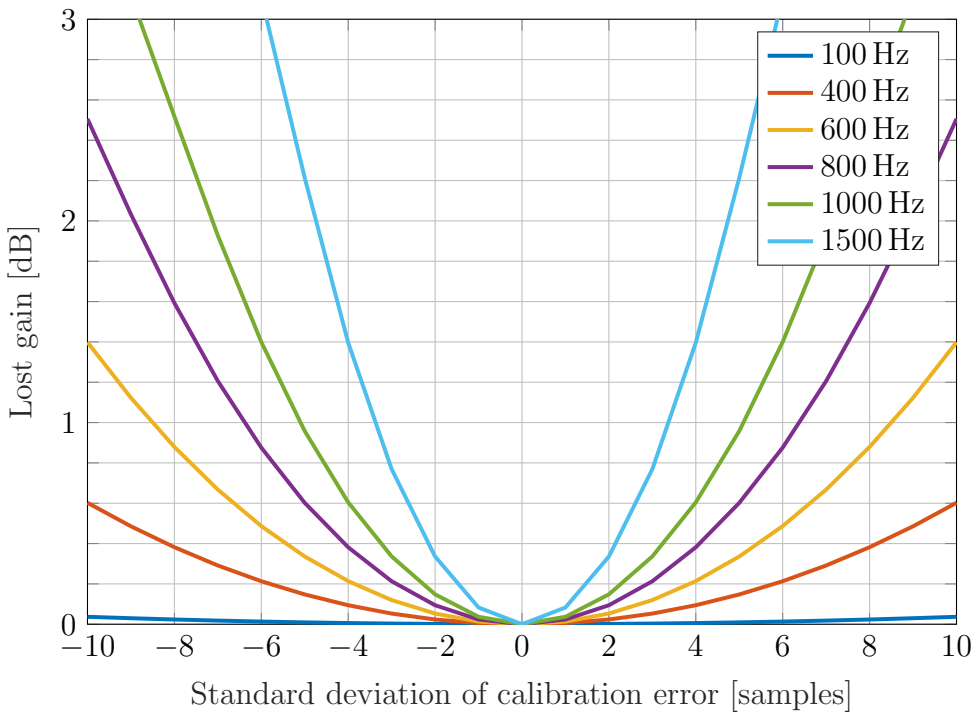


Figure 3.4 – Plot of lost processing gain as a function of the standard deviation of offset error, for a six-microphone setup.

Figure 3.4 presents the results for a six-microphone setup. Unlike the two-microphone setup, in this case there are multiple offsets which can be inaccurately computed. To allow for a single-dimensional presentation, the offsets computed by the calibration procedure are modelled as a zero-mean, normally distributed error added to the true distance from the source. The plot presents the loss as a function of the standard deviation of the error.

As can be seen from the figures, the calibration procedure must be accurate in order to ensure usable gain. For example, if we limit ourselves to a loss of 1 dB, the maximal error we can tolerate is three or four samples at the higher frequencies, with the lower-frequency signals allowing for slightly larger errors. A four-sample error corresponds to the time sound takes to travel 3 cm, which means that calibrating the system by measuring distances from the speaker to microphones is likely infeasible.

Even if we could achieve sub-centimetre accuracy, an accurate model of sound

propagation is required to compute the travel time, which would require knowledge of other relevant parameters. For example, if the distance between the speaker and microphone is 2 m, an error of $1\,\mathrm{m\,s^{-1}}$ in the speed of sound used by the model would result in the offset changing by over a sample. Such an error is likely to occur in practice – for example it could result from, a 2 °C change in ambient temperature (since the speed of sound changes by $\approx 0.606\,\mathrm{m\,s^{-1}}$ for each degree Celsius).

The results also indicate that manual calibration, where a human operator aligns the recordings "by ear", is not a realistic approach. Even at high frequencies, changing the offset by a single sample results in a change in SNR of around 0.3 dB. Detecting this change reliably in a noisy recording is difficult, and is further complicated by the need to align recordings with varying signal-to-noise ratios, where the change in the output's SNR is even lower.

In conclusion, as neither manual nor measurement-based approaches are likely to work, the calibration procedure must be automatic, based on the recordings to be combined.

3.3 Speech-to-text performance in noisy conditions

This test serves to provide a baseline for the expected improvement in speech-to-text performance on the processed recordings, by evaluating the accuracy of the engine on samples with varying levels of noise.

A randomly-chosen 100-sample subset of the LibriSpeech "test-clean" dataset (twelve minutes long in total) was used in this test. Gaussian noise was added to each sample, with the amplitude chosen to achieve the desired SNR. For this purpose, the signal power was computed as the average squared amplitude of the input audio.

The resulting noisy samples were transcribed using the DeepSpeech speech-

to-text engine. To compute the word error rate, each sample was aligned with the ground truth transcription separately. The resulting WER was computed by dividing the total number of errors with the total length of inputs, instead of averaging the error rates for individual sentences. This avoids overweighting errors in short samples (otherwise, a single error in a two-word sentence would be penalised as much as five errors in a ten-word sentence).

The baseline error rate of the speech-to-text engine on the samples used is 9 %. This is the word error rate without any additional noise in the inputs, and therefore serves as a lower bound for the engine's performance on the noisy samples.

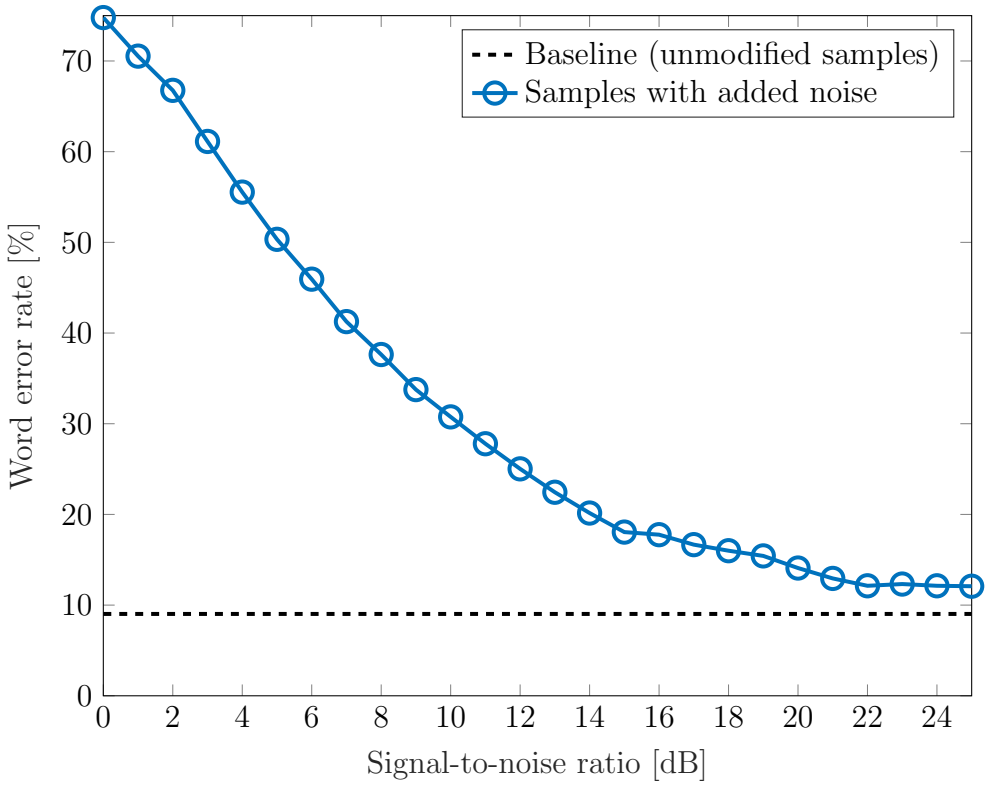


Figure 3.5 – Transcription word error rate as a function of the input signal-to-noise ratio. The dashed line shows the baseline WER (without noise).

Figure 3.5 shows the obtained word error rates as a function of the signal-to-noise ratio. The error rate starts at 74 % at 0 dB, and drops with increased

SNR until stabilising above 22 dB, at 12 %. The improvement in the word error rate is greatest at lower signal-to-noise ratios, with diminishing returns as the signal power improves.

Approximating the segment of the curve up to 10 dB (where the improvement in the error rate per decibel is greatest) as a linear function, we can estimate the improvement in this region as $4.4\,\%\,\mathrm{dB}^{-1}$. For two microphones in this region, with equal signal-to-noise ratios (therefore providing the highest possible gain), the maximal improvement in WER is therefore $3\,\mathrm{dB} \cdot 4.4\,\%\,\mathrm{dB}^{-1} = 13.4\,\%$

For a three-device setup used in the project's evaluation, the maximal possible gain is 7.8 dB, which gives an upper bound of 34.2 % for the word error rate improvement. This is, however, not a realistic bound, since it assumes both that all six microphones record the speaker with equal SNRs. A more conservative estimate of three equally-noisy recordings (for the microphones facing the speaker) results in an upper bound of 21 % if the input signal-to-noise ratios are near 0 dB, with the bound lowering as the signal power increases.

Chapter 4

Design and Implementation

This chapter will describe the implementation of the setup used to record samples in the evaluation of the eavesdropping system, as well as the algorithms used to process the recordings.

It starts with an overview of the setup, outlining the implementation choices made in the recording application and associated servers, and their implications on the work done in the rest of the project. This description is followed by the approach used to compute the calibration parameters. The final section of the chapter describes the algorithm used to combine the recordings after the calibration phase.

4.1 Recording setup

The first component of the project is the system used to record data from multiple devices simultaneously. This system allows the user to automatically start recording from all connected devices at the same time (up to a small delay) and retrieve the recordings over the network. For convenience, additional features such as playing audio files are supported to allow for automated testing.

The recording system consists of the following components:

- an Android application running on each device,
- a command server the devices connect to, which is used to remotely control their operation, and
- a file server used to upload and download audio files.

4.1.1 Recording app

All devices (smartphones and tablets) in the system run an Android application, which is used to make recordings and upload them to the file server. The application can be controlled manually using a simple user interface, or set up to connect to the command server, listen to commands and execute them.

The application uses the Android AudioRecord API, which provides the capability to start and stop recording in a separate background thread, letting the user interface respond to events in the meantime. The recorder is configured to output two 16-bit channels. The sample rate is set to 44.1 kHz, since this is the only value which is guaranteed to be supported in all devices. The recorded audio is saved as a PCM (pulse-code modulation) file, which contains the raw sampled values represented as 16-bit integers, without compression.

Since many mobile devices have two independent microphones, the application attempts to record from both of them simultaneously when possible. This functionality is not officially supported by the Android API, but some devices support it in certain recording modes.

For the three devices used in this project, configuring the recording mode to "camcorder" enables access to both microphones. This option produces a stereo recording, which has the outputs of the two microphones written to the two audio channels.

To allow for automated testing, the application connects to a server and executes remote commands. While the primary purpose of this is to trigger the start of a recording, remote control can also be used to transfer files to

and from the server, as well as to play an audio file (if the device serves as the sound source in a test).

4.1.2 Servers

The application communicates with two servers. The first is used to control the application remotely, and the second serves as a point to which files are uploaded. Since the functionality of these two servers is completely separate, they are implemented independently. This means that they can in principle run on separate hosts, but in practice it is more convenient to use a single computer as the setup controller.

Command server

The command server's purpose is to automatically control a testing setup by sending commands to the connected devices. It implements a simple TCP-based protocol for issuing commands to the phones, as well as a small scripting language that lets the user automate tests.

All tests start with the server waiting for a specified list of devices to connect (since the connections are initiated client-side). After opening the connection, each device identifies itself by sending an ID, which the server uses to determine which client should receive each command. When all devices used in the test are connected, the execution continues.

Ideally, we would want the start times of the recordings on all devices to be synchronised. This is, however, difficult to achieve, since there is no reference clock that is shared by all devices. If we wanted the start times to differ by less than a sample, we would require a shared clock with sub-millisecond accuracy (since the delay between samples is approximately 23 µs). A further obstacle is the delay between the recording request and the actual start of recording, which requires several API calls and requests to the operating system.

Consequently, the command server does not implement any synchronisation mechanism, and will only dispatch commands as fast as possible (unless a delay is explicitly specified). If commands are sent to several devices to

trigger the start of recording, there will be a small delay between them, and there is no guarantee that the recordings will begin at the same time. Therefore, the offset between recordings from separate devices does not only depend on the distances to the source, but can also include a small delay, and the calibration procedure cannot assume that inter-device offsets are bounded by some small distance.

File server

The second server component is a simple HTTP server, which hosts audio files that are used during testing and provides a location to which the application can upload recordings. To provide the samples used for tests, it exposes a part of the local filesystem as a static directory. Recordings are uploaded by making a POST request to the server, providing the file contents and the name the file should be saved under.

4.2 Calibration

Before the recordings can be combined into a single output, a calibration step is necessary to obtain the required parameters: time offsets between the source and each microphone, and signal-to-noise ratios of the recorded audio. As described in Section 3.2, these parameters cannot be tuned "by ear" or by measuring the positions of devices, since the required precision is lower than what can be practically achieved by such methods.

To evaluate whether the knowledge that a particular tone was played at the location of interest could help with calibration, an approach relying on an attacker-generated "calibration tone" was tested, as well as one that does not depend on such knowledge. The generation of the tone and its properties are described in the first subsection, followed by the algorithms used to compute offsets and signal-to-noise ratios.

4.2.1 Calibration tone

For the calibration methods relying on a known tone played at the target's location, a "calibration tone" with properties useful to the attacker was generated. In particular, the tone was generated to have a sharp peak in its autocorrelation at the zero-offset, and low values elsewhere. This makes locating it in a recording simple, since the cross-correlation of the tone and the recording will only have a large peak at the offset where they are aligned.

Further, the spectrum of the calibration tone is restricted by the behaviour of smartphone speakers and microphones. Since they attenuate both low and high frequencies, the range in which we can expect the highest-power recording is around 1 kHz, and the generated tone should not contain extremely low or high frequencies.

To obtain a tone with a sharp autocorrelation peak, a pseudonoise sequence was generated using a maximum-length linear-feedback shift register. The resulting binary sequence was interpreted as a square wave, with zero-bits mapped to -1 and one-bits to 1. The result, for a pseudonoise sequence of length N and a chip rate B, is an output with an autocorrelation of N at the zero-offset and $-\frac{1}{N}$ elsewhere. In the frequency domain, the spectrum of the output is centered around 0 Hz, and has bandwidth B.

To shift the spectrum of the tone, the square wave is multiplied by a sine wave whose frequency is the desired centre of the spectrum f_c. Equivalently, we can interpret this process as binary phase modulation of a pseudonoise sequence on a sinusoidal carrier. In the frequency domain, we can view this transformation as a convolution with a pair of delta-functions at f_c and $-f_c$, which will shift the spectrum to be centered at that frequency. The result is therefore a calibration tone band-limited to $[f_c - B, f_c + B]$.

4.2.2 Offsets

Two approaches to computing offsets at which the sound coming from the target occurs in the recordings were tested. The first relies on the presence of a known calibration tone in the recordings, and searches for that tone in each

input separately. The second approach does not make such an assumption and is based on pairwise cross-correlations of the recordings. It is, however, less suited for scenarios with multiple speakers in a room, where there are multiple sources to be distinguished.

It should be noted that, in both cases, the inputs to the calibration procedure are recordings which do not necessarily start at the same time. Therefore, the output offsets are not bounded by the distance between microphones, and the search cannot be restricted to a small set of possible results.

For the rest of this section, it will be useful to adopt a simplified model of the recordings $s_i(t)$. We can treat each recording as a sum of several scaled and shifted copies of the sound $x(t)$ coming from the source (due to reverberation) and a noise component $n_i(t)$:

$$s_i(t) = \sum_k A_k x(t - d_{i,k}) + n_i(t)$$

Without loss of generality, assume that A_0 is the largest amplitude, corresponding to the shift $d_{i,0}$ the calibration procedure is expected to return. This is likely to be the sound travelling from the source to the microphone via the direct path, as this is the shortest distance and should therefore have the highest power. In practice, this might not be the case for a highly directional microphone facing away from the source. In such a case, a reflection with a longer travel time might result a higher amplitude in the recording than the direct path. In such a case, however, returning $d_{i,0}$ is still the correct behaviour, as we are searching for the highest-power copy of the signal regardless of its path.

Searching for the calibration tone

In the first approach, we assume that the recording contains a known tone transmitted from the source. While this approach requires an attacker who can not only record from all devices, but also cause them to play a sound (which might risk detection if it is in the audible range), it is more robust to

interference from secondary sources than methods that assume no knowledge about the signal, as it allows the attacker to differentiate it from the sound coming from other sources.

A straightforward method for finding the offset would be to compute the cross-correlation of the recording and calibration tone. In the model above, let the sound transmitted by the source be the calibration tone $x(t) = c(t)$. Assuming that $n(t)$ is uncorrelated with $c(t)$, their cross-correlation will be zero, and (with $R_{s,c}$ representing the cross-correlation of $s(t)$ and $c(t)$)

$$R_{s,c}(\tau) = \sum_i A_i R_{c,c}(\tau + d_i) + R_{n,c}(\tau) = \sum_i A_i R_{c,c}(\tau + d_i)$$

In this idealised model, the computed cross-correlation will be a sum of time-shifted copies autocorrelation function of the calibration tone $R_{c,c}$. Since the maximum of $R_{c,c}$ is at the zero offset, each echo of the calibration tone will result in a local maximum in the cross-correlation, with the height of the maximum proportional to the amplitude A_i. If the autocorrelation of the calibration tone is narrow enough that its copies do not overlap, the peak corresponding to A_0 will be the highest, and d_0 can therefore be found as the index at which $R_{s,c}$ is maximal.

To improve the accuracy of the calibration procedure and reduce the influence of echoes, the approach described above was implemented using generalised cross-correlation, which multiplies the inputs' spectra with a weighting function ("processor") before computing the cross-correlation. The PHAT processor was chosen, as it has been shown to perform well in the presence of reverberance.

Under ideal conditions, this would result in copies of the calibration tone manifesting as peaks without any spreading to adjacent offsets, as the generalised cross-correlation of a function with itself is zero at all offsets except zero. In practice, this will not happen, as the tone will be distorted by the microphone, so a narrow autocorrelation is still desirable.

Pairwise alignments of recordings

Consider the generalised cross-correlation of two recordings $s_i(t)$ and $s_j(t)$, modelled as described above. To simplify the explanation below, we will assume that the noise signals in the two are uncorrelated. In practice, this assumption does not hold since all microphones will record the same ambient noise, but the impact of the noise on the cross-correlation is low enough not to impact the result.

Under this assumption, the cross-correlation of the two recordings will be equal to the sum of pairwise cross-correlations of the echoes of $x(t)$. For each pair of offsets $d_{i,a}$ and $d_{j,b}$ at which the x occurs in the two recordings, the cross-correlation will contain a peak at $d_{i,a} - d_{j,b}$ with an amplitude proportional to $A_{i,a}A_{j,b}$. As this factor is maximal for $A_{i,0}A_{j,0}$, the difference $d_{i,0} - d_{j,0}$ can be found as the index at which the generalised cross-correlation of $s_i(t)$ and $s_j(t)$ is maximal.

Calibration starts by computing this difference for each pair of recordings. For each pair (i, j), label the result $O_{i,j} = d_{i,0} - d_{j,0}$. Let D_i be the output offsets of the calibration procedure.

Since the offsets are only used to align the recordings, we can set an arbitrary value for one of the outputs and align the rest of the recordings based on that reference – in other words, the outputs should be of the form $D_i = d_{i,0} + \delta$ for an arbitrary constant δ. Note that the offset δ is cancelled when subtracting two offsets D_i and D_j, giving $O_{i,j} = D_i - D_j$.

Without loss of generality, let $s_0(t)$ be the recording with the highest signal-to-noise ratio we will use as the reference, and set $D_0 = 0$. For a pair of recordings (i, j), we have:

$$O_{i,j} - O_{0,j} = (D_i - D_j) - (D_0 - D_j) = D_i$$

For the i-th recording, this provides us with $n - 1$ values for the offset D_i, one for each recording it is compared with. To improve the accuracy of the

calibration procedure and remove outliers, the median of the $n - 1$ values is taken as the output.

4.2.3 Signal-to-noise ratio

The second set of values the calibration procedure computes are the signal-to-noise ratios for each of the recordings. In a scenario where the recordings include a tone known to the attacker, the tone can be used as a template to find the signal power.

The signal power can be computed as the value of the cross-correlation of the recording and template at the offset corresponding to the start of the calibration tone (computed in the previous step). Here, we assume that the signal and noise are uncorrelated, and that the calibration tone's autocorrelation is narrow enough that echoes do not overlap with the direct signal in the cross-correlation. Finally, the noise power can be estimated by subtracting the signal power from the RMS of the recording, allowing us to compute the signal-to-noise ratio.

However, evaluation results show that the output of a smartphone speaker is distorted to a level where an approach based on a known template is not viable. As an alternative, in a single-speaker scenario, where we can assume that the recording only consists of the signal and background noise, a method based on estimating the noise power was used.

The recording is split into 100 ms segments, and the total power of each segment is computed as the mean square amplitude. The segments are classified as "signal" and "silence" based on the power level, with the threshold computed as a fixed gain above the minimal power observed. A 500 ms interval of consecutive "silence" segments is used as a reference, with the noise power N estimated as the average power in this interval.

Assuming that the signal and noise are independent and the only two components of the recording, the total power P is equal to the sum of the signal power S and noise power N. Therefore, the signal-to-noise ratio can be computed as

$$SNR = \frac{S}{N} = \frac{P - N}{N} = \frac{P}{N} - 1$$

While this assumption is not correct in the presence of reverberation, the echoes travelling a reflected path are a significantly lower contribution to the total power than the direct path. Further, the reflected paths which contribute the most are those whose length is close to that of the direct path, and their power is therefore likely to be proportional to the power of the direct-path signal. The contribution is therefore likely to affect the SNR estimates for all recordings similarly, with little effect on the relative signal-to-noise ratios.

4.3 Combining recordings

The final processing step takes the recorded audio files and the parameters computed during the calibration process, and combines them to produce a single audio file. Label the individual recordings $s_1(t), s_2(t), \ldots, s_n(t)$, the calibration offsets $D_1, D_2, \ldots, D_n$ (the calibration tone starts at D_i in the i-th recording), and signal-to-noise ratios $A_1, A_2, \ldots, A_n$.

For convenience, the rest of this section will assume that all input signals have the same power. This assumption does not necessarily hold in practice, both due to varying signal power (since the distance from the source to the microphones varies) and differing microphone gains. The inputs must therefore be normalised before further processing by dividing with their RMS (label it R_i).

Aligning the signals to obtain constructive interference is straightforward. We can simply shift each signal to the left by the corresponding offset and sum them to obtain the output (with the n^{-1} as a normalisation factor):

$$S(t) = \frac{1}{n} \sum_i s_i(t - D_i)$$

Since the inputs can vary in length due to the shifts and inaccuracy in the timer used to stop recording, they are zero-padded after shifting.

Simply summing the recordings as described above is likely to result in an improved output if all the inputs have the same SNR, but is not appropriate when the input signal strengths vary. As an extreme example, consider a case where only a single microphone makes a useful recording of the target, and the rest only pick up noise. In this case, taking their sum will only increase the noise in the result, while keeping the signal power the same, and will therefore lower the SNR.

To avoid such a scenario, we need to assign weights to the inputs, prioritising those with a good signal-to-noise ratio. One viable option, maximum-ratio diversity, is to use the SNRs as the weighting coefficients (note that these are the actual power ratios, not their logarithms). In a simplified model, where all inputs are the sum of a single signal and Gaussian noise, maximum-ratio diversity is the optimal approach to combining them, and results in an output whose signal-to-noise ratio is the sum of the inputs' SNRs. While these assumptions do not hold for realistic recordings, this approach still results in an improved output.

The complete approach to combining the inputs is therefore to align them, and then compute a weighted sum of normalised signals with A_i as the coefficients:

$$S(t) = \frac{1}{\sum_i A_i} \sum_i \frac{A_i}{R_i} s_i(t - D_i)$$

Chapter 5

Evaluation

This chapter will present the experimental tests used to evaluate the performance of the eavesdropping system prototype. The first section describes the physical setup used to record test data, shared between all tests. It is followed by the description of methodologies used in each test and their results, covering the evaluation of calibration accuracy, the gain obtained by combining recordings of speech, and the impact on the word error rate of transcribed recordings.

5.1 Setup

All recordings were made in room which was quiet except for the single sound source. The room was, however, not isolated from background noise, with main contributors being sound coming from the corridor and noise produced by electronic devices.

An LG Nexus 5X smartphone served as a sound source, and three devices were used to record as components of the eavesdropping setup: an HTC Nexus 9 tablet, a Huawei Mate 20 Pro smartphone, and a Xiaomi Note 6 smartphone. All three devices are capable of independently recording using two microphones.

5.2 Offset calibration accuracy

An error in computing offsets used to align recordings will affect the performance of all further processing, making the accuracy of this calibration step a limiting factor in the system's performance. Measuring this accuracy, therefore, is an important evaluation step which provides us with a bound on the gain that can be obtained by the rest of the system.

Since the processing used to compute offsets is independent from the rest of the pipeline, it can be tested in isolation. As the rest of this section will describe, the offset error cannot be measured directly, but it can be inferred from the results of repeated calibration runs.

The most direct way to evaluate the calibration procedure would be to measure the error in the offsets it computes. This, however, requires us to have a ground truth value for the offsets. Obtaining such a value is problematic, since it would require us to accurately measure the distance from the source to the receiver. If we take potential issues with the setup geometry into account (for example, microphones are not exactly on the edge of the smartphone), it is unlikely that this value can be significantly more accurate than the calibration error.

Another issue is the lack of a shared reference clock between the audio source and recording device. The commands to start playing the test tone and to start recording are not sent simultaneously, and will therefore not arrive at the same moment (and even if they were, network delays would desynchronise them). Synchronising clocks on the devices and scheduling commands for a fixed point in the future is not a viable solution either, since aligning the execution with sample-level accuracy would require a sub-millisecond time offset.

To resolve these two issues, the difference between calibration offsets was measured using two microphones on the same device. Assuming that the setup is not moved, the expected value is constant across measurements, since it only depends on the distance between the source and the microphones.

Subtracting the two offsets eliminates the unknown time delta between the source and recording device.

The standard deviation of this difference, σ_δ, is not a direct measurement of the calibration error, but it is a useful proxy. If we assume that the errors are uncorrelated and normally distributed, with zero mean and standard deviation σ, the measured value will be $\sigma_\delta = \sqrt{2}\sigma$. The assumption that the two errors are identically distributed is unlikely to hold in practice, since the recordings have different signal-to-noise ratios. In this case, however, σ_δ is still an upper bound for σ. Finally, the difference in offsets is particularly relevant for a single-device scenario, since it is the parameter which directly impacts the loss of gain.

Four sets of recordings were made, varying the positions of the recording devices. In the first two, the devices were 2.5 m away from the source. In the first set, the bottom microphones were facing the source (i.e. "correctly" positioned when viewed from that point); in the second, the devices were placed sideways. The third and fourth set use the same positioning, with the sound source placed at a distance of 2 m.

For each of the four setups, three sets of 120 recordings were made using different calibration tones. The tones were generated in the following frequency bands: 1200–2000 Hz, 800–1600 Hz, and 800–2000 Hz. Since the recordings for the three devices were processed separately, a total of 36 sets of samples were used to evaluate the offset calibration methods. The same dataset was used to evaluate both approaches.

5.2.1 Results

Searching for the calibration tone

Table 5.1 illustrates the behaviour of the first approach, based on searching for the calibration tone in the two recordings separately, showing the distribution of offsets computed for three sets of recordings. Here, we can see that there are two different types of offset errors: small variation in the result, and large deviations from the expected output, on the order of hundreds of

samples. While the first type of error is due to the limited accuracy this test sets out to measure, the second indicates that the offset found in one of the two recordings does not correspond to the direct path from the speaker to the microphone.

Table 5.1 – Distribution of differences in offsets of the two microphones for three sets of recorded samples.

Set 1	Offset	-3	0	1	
	Count	1	101	18	
Set 2	Offset	-2	-1	572	
	Count	47	71	1	
Set 3	Offset	-629	-628	-366	-2
	Count	102	5	12	1

Due to the second type of errors, the correct offset cannot be computed as the mean of the individual outputs, as the deviations are not symmetrically distributed. Before averaging the values, the outliers must first be eliminated. To classify the offsets, we can assume that the largest group, defined as values within a 20-sample interval, is the one which corresponds to the correct output value.

If we assumed that the direct paths from the source to the microphones is always the one that should be identified during calibration, the correct offset would be at most around 30 samples (due to the devices' dimensions). Taking the largest group instead of the one centered around zero ignores this assumption, as it might be possible that a reflected path results in a stronger signal due to a backwards-facing microphone's directionality.

Table 5.2 – Distribution of offset errors aggregated across all runs. Each entry includes values within five samples from the offset. Entries with less than three occurrences are omitted.

Offset	-755	-577	-400	-377	-247	0	243	262	323	622
Count	44	90	26	126	61	3659	62	46	55	17

Table 5.2 shows the distribution of outliers aggregated across all measurement runs, and Figure 5.1 shows the distribution of errors around the zero offset.

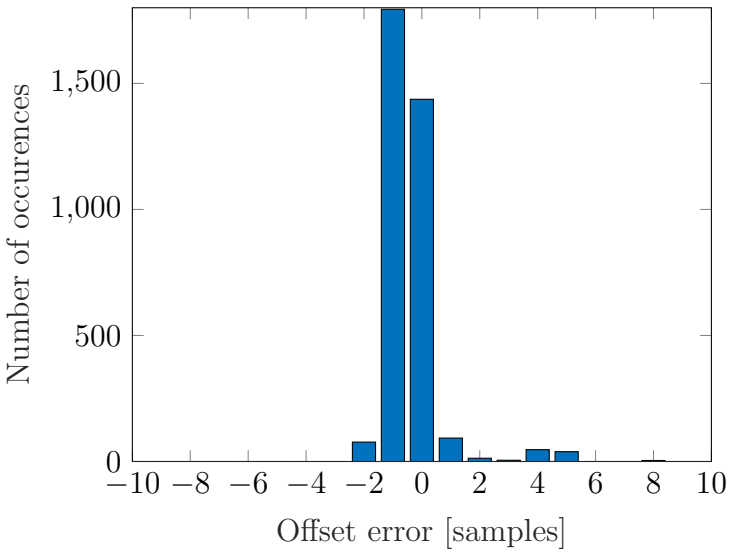


Figure 5.1 – Distribution of offset errors near the zero-offset (not showing outliers).

In each run, the correct value (zero error) was computed as be the mean of the largest group of offsets. The average error due to limited precision was computed as the standard deviation of values in $[-100, 100]$, treating all errors outside this range as outliers due to a failure to detect the best copy of the calibration tone.

The standard deviation of the errors is 1.63. While this is relatively low, there is a significant number of outliers, with 15.1 % of the values outside the $[-100, 100]$ range.

Cross-correlation of recordings

Table 5.3 – Distribution of offset errors aggregated across all runs. Each entry includes values within five samples from the offset.

Offset	-436	-208	378
Count	19	1	12

The same set of recordings was used to evaluate the second approach, where the offset is computed as the position of the highest peak in the generalised

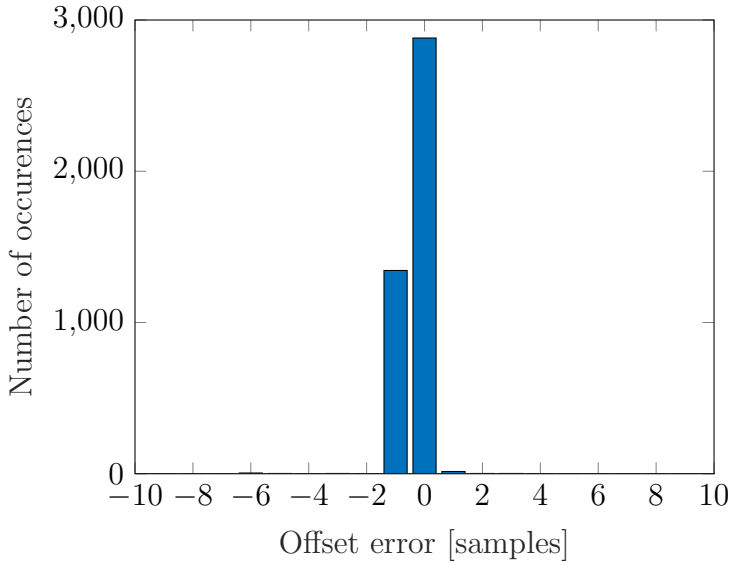


Figure 5.2 – Distribution of offset errors near the zero-offset (not showing outliers).

cross-correlation of the two recordings. The results were processed using the same approach described above, and the distribution of resulting offset differences aggregated across all runs is shown in Figure 5.2, with the outliers presented in Table 5.3.

The standard deviation of the non-outlier offsets is 2.02, with 0.74 % of the offsets outside the $[-100, 100]$ interval.

Unlike the results obtained using the previous approach, the mean for each group is low, and the maximal offset is 22 samples. This distance corresponds to a travel distance of 33.2 cm (assuming the speed of sound is $330\,\mathrm{m\,s^{-1}}$), which is lower than the distance between microphones on the Nexus tablet. We can therefore conclude that the copies of the source audio identified by the cross-correlation are results of the direct source-microphone path, unlike some of the values obtained using the first approach.

The low outlier rate shows that computing the cross-correlation of outputs is a more robust approach than searching for the calibration tone in each

recording separately. This suggests that, while the individual microphones' distortions are not an obstacle when computing the cross-correlation, the output of a smartphone speaker is sufficiently different from the audio used to make searching for the known calibration tone unreliable.

In conclusion, the approach based on pairwise generalised cross-correlation outperforms searching for the calibration tone, with a significantly lower number of outliers. It should, however, be noted that a scenario with multiple speakers would pose a significant challenge to an attacker, as the cross-correlation will result in several offsets that can be considered correct (due to the presence of multiple sources). The results presented above show that exploiting calibration tones played from a smartphone speaker is made difficult due to distortion, and that an attacker wishing to introduce a robust reference signal into the recordings would likely benefit from a higher-quality speaker, which is unlikely to be available in all locations for a large-scale deployment.

5.3 Improvement in signal-to-noise ratio

The purpose of this test is to quantify the gain obtained by combining recordings from multiple devices. The gain is measured by recording a known audio sample and computing the signal-to-noise ratios of the individual microphones' recordings, as well as that for the combined output. The reference point is the microphone with the highest SNR, and the gain is computed as the improvement of SNR from this baseline.

Before processing, the receiving setup must first be calibrated to obtain the microphone offsets and weighting SNR coefficients. Since the delay between the moment when the sample is played and when recording starts varies between devices and tests (due to network and command processing latency), calibration must be done for each test separately. To construct the audio files played at the source, the test sample is appended to the calibration tone after a short delay. As an added benefit, this approach avoids the need to search for the sample in the recorded output, since it is placed at a known offset

from the calibration tone.

To provide a realistic evaluation of the system's behaviour in an eavesdropping scenario, a randomly-chosen set of speech samples from the LibriSpeech dataset was used as the test data. Three sets of 180 recordings each were used, with the position and orientation of the recording devices changed between sets. The placements in the first two runs were similar, with only small adjustments, and all three devices were placed with one microphone facing the source directly. In the third, they were rotated by 90°, positioning them so that both the front and back microphone are at a right angle to the source. The distance from the source to the devices was approximately 2 m in all setups.

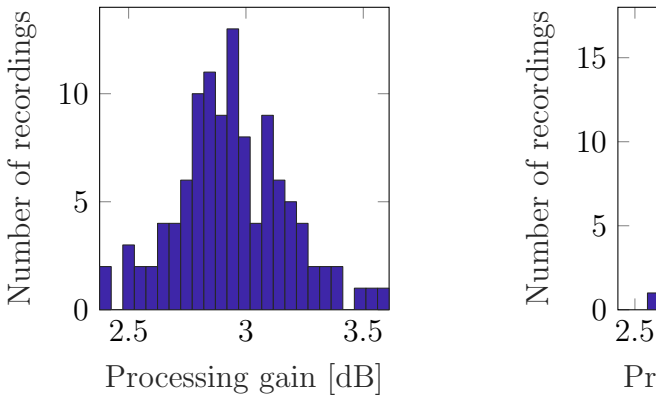
5.3.1 Results

Figure 5.3 shows the distributions of gain obtained by combining the recordings for each of the three runs. The mean and standard deviation of the values for the three runs is shown in Table 5.4. We can see that the gain is relatively constant for a fixed setup, but can vary significantly depending on the placement of the devices, due to the relative signal-to-noise ratios changing with the distances and angles from the source.

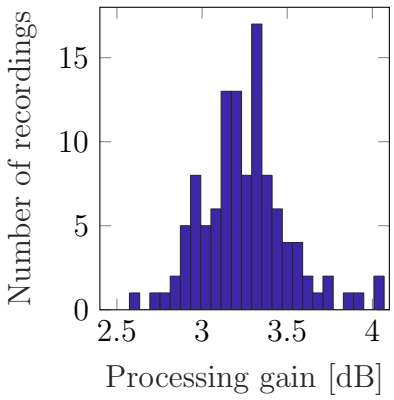
Table 5.4 – Gain obtained by combining the recorded samples for each run, and the upper bound for the improvement (computed as the sum of SNRs of recordings).

	Gain [dB]	Upper bound [dB]	Difference [dB]
Run 1	2.9 ± 0.3	3.6 ± 0.2	0.7 ± 0.2
Run 2	3.3 ± 0.3	4.0 ± 0.2	0.8 ± 0.3
Run 3	2.0 ± 0.3	2.9 ± 0.3	0.9 ± 0.2

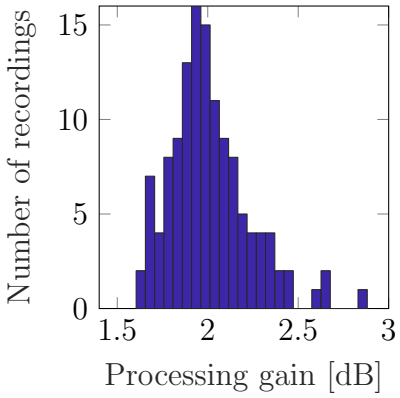
It seems intuitive to expect that the gain in the third run would be greater than in the first two (though the total SNR might be lower), since the signal-to-noise ratios of six microphones at a ninety-degree angle should be closer than if half were facing away from the source. The results show that this is not the case, and that the gain is greater when the devices are pointed towards



(a) Run 1



(b) Run 2



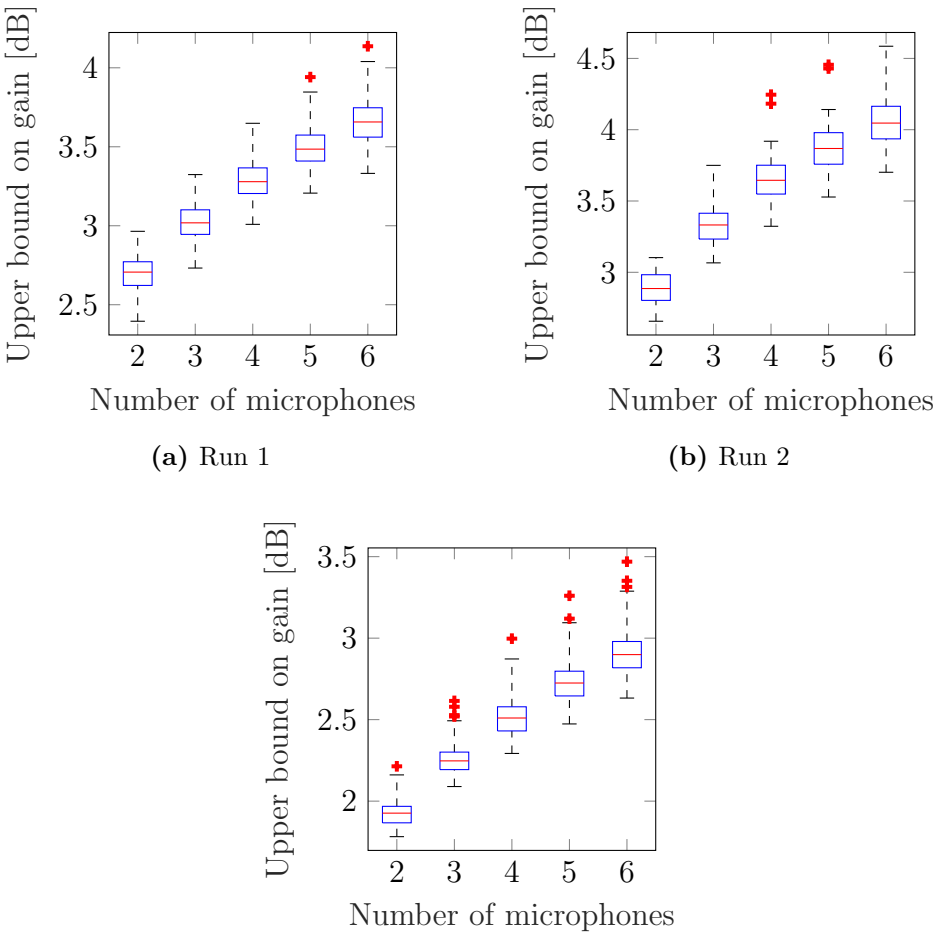
(c) Run 3

Figure 5.3 – Distribution of gains obtained by combining recordings, compared to the baseline of the best microphone available, for each of the three evaluation runs.

the source. This is due to differing directionality of the microphones – turning the devices away from the target lowers the SNR by varying amounts, and one less-directional microphone dominates the results in the third run.

The sum of the signal-to-noise ratios of individual recordings can be used as an upper bound of the processing gain available in the recorded data. These values, shown in Table 5.4, are lower than the 4.77 dB estimate used in Section 3.3, which assumes three microphones with equal SNRs facing the target and no contribution from those facing in the opposite direction.

To further demonstrate the contribution of the individual microphones, consider the gain that could be obtained by using the best k recordings instead of all six. If the signal-to-noise ratio was constant across the recordings, we would expect this value to be $\log_{10} k$, but differing SNRs reduce the improvements from later microphones. Figure 5.4 shows a plot of the upper bounds for these values for each of the three runs, from $k = 2$ (the two microphones with the highest signal-to-noise ratio) to $k = 6$ (all available microphones).



(a) Run 1

(b) Run 2

(c) Run 3

Figure 5.4 – Box plot of available gain when using the best k microphones. The red marker indicates the median, with the box covering values from the 25^{th} to the 75^{th} percentile. Outliers further than three standard deviations from the median are displayed as red crosses.

The majority of the available gain comes from combining the best and second-best recording, with an average ranging from 2.9 dB in the second run to 1.8 dB in the third. Additional recordings do not contribute as much, but are still significant, with the additional improvement falling for each added microphone.

We can see that even in a setup where all distances from the source to the devices are equal, and all orientations are the same, the signal-to-noise ratios of the recordings will not be equal. As the recording quality is influenced by a number of factors, including the quality of the device's microphone, its directionality and the effects of reverberation, that the best-performing device is not necessarily the one closest to the target. Therefore, an attacker with access to multiple compromised devices can gain a significant advantage over one listening to a single microphone just by choosing the best recording available. As the results presented above have shown, the output can be further improved by combining all microphones if multiple recordings with similar signal-to-noise ratios are available.

5.4 Speech-to-text word error rate

The final stage of the project's evaluation is to quantify the improvement in clarity of the recorded audio after processing, measured as the error rate in transcribed text. To allow for easier processing of large sets of recordings, and ensure the repeatability of the experiments, a speech-to-text engine was used to transcribe the recordings.

As a motivating scenario for the use of speech-to-text, consider an attacker deploying an eavesdropping setup at scale, compromising a large number of smartphones in multiple locations. In such a situation, it would be infeasible for the attacker to sift through all recordings manually looking for interesting information due to the volume of recorded data. A better-scaling alternative would be to automatically transcribe all processed recordings. The resulting text could then be used to search for keywords of interest to identify "interesting" conversations (possibly flagging them for human investigation).

The set of samples recorded for evaluation of the improvement in signal-to-noise ratio in Section 5.3 were used to evaluate the improvement in speech-to-text error rate. As described in the previous section, the dataset is comprised of three setups, each consisting of 180 recordings of a randomly-chosen sentence from the LibriSpeech dataset.

After processing, the resulting audio file was split into two segments, containing the calibration tone and the speech sample. The second segment was used as the input to the speech-to-text engine, and the resulting transcription was compared to the ground truth sentence to compute the word error rate.

For each sample played at the source, the recording with the highest signal-to-noise ratio was used as the baseline to compute the baseline error rate that can be obtained without combining multiple recordings. Here, we are comparing the attacker who has access to all devices, but only selects the best-performing microphone, to one who integrates all recordings into a single output. In practice, even choosing the best microphone is often an advantage in such a scenario, providing an improvement over the performance an attacker with access to a single device, potentially with lower signal-to-noise ratio.

The speech samples were transcribed using the DeepSpeech engine. As the results have shown that this engine is not well-suited for use with smartphone recordings, further evaluation was done using the Google speech-to-text API.

5.4.1 Results

Attempting to transcribe the samples using DeepSpeech has shown that it is not suited to processing recordings made using smartphone microphones, with non-Gaussian noise and distortion due to the speaker and microphone frequency responses. The performance, shown in Table 5.5, is significantly worse than the results obtained in Section 3.3 for inputs with the same signal-to-noise ratio ($\approx 10\,\mathrm{dB}$). When transcribing the baseline recordings made by the microphone with the highest signal-to-noise ratio, the engine does not

detect any speech in over 25 % of the recordings. Further, the output not much better than returning an empty string, with an error rate above 97 % in all three runs.

Table 5.5 – Performance of the DeepSpeech speech-to-text engine on the recorded samples for the microphone with the highest SNR and the combined output, showing the percentage of inputs for which no speech was detected, and the word error rate.

		Best microphone	Combined	Improvement
Word error rate	Run 1	97.7 %	95.9 %	1.8 %
	Run 2	98.2 %	97.5 %	0.7 %
	Run 3	98.5 %	97.7 %	0.8 %
% of inputs with no speech detected	Run 1	33.8 %	6.8 %	27.0 %
	Run 2	24.8 %	10.0 %	14.8 %
	Run 3	24.8 %	10.0 %	14.8 %

As shown in Table 5.5, processing and combining the recordings into a single output does not result in a dramatic reduction in the error rate, but it does have measurable impact. The most notable effect is the reduction in the number of recordings where no speech was recognised, which is lowered by 27 % in the first run, and by 14.9 % in the remaining two. The word error rate is lowered slightly, but is still far from reliable.

To provide a second perspective on the improvements due to processing, a subset of the recordings was transcribed using the Google speech-to-text API, which is better suited to smartphone recordings. Due to rate limits on the API, only fifteen randomly-chosen samples were used for each run.

Table 5.6 – Word error rate of transcriptions made using the Google speech-to-text engine when processing the recording with the hignest SNR and the combined output.

	Best microphone	Combined	Improvement
Run 1	70 %	57 %	13 %
Run 2	69 %	62 %	7 %
Run 3	76 %	73 %	3 %

Since few recordings were transcribed, and the error rate within a run varies significantly from sentence to sentence due to the large impact of small

changes (for example, a single mistake in a four-word sentence would increase the error rate by 25 %), the results shown below were obtained by computing a single word error rate for the entire run instead of the average per-sentence error rate. While each transcription was aligned with the expected sentence separately, the edit distances were summed before dividing by the total length of the ground truth transcriptions. This can equivalently be viewed as the word error rate of the concatenated samples, with a restriction forbidding matching words across sentence boundaries.

Table 5.6 shows the results of the word error rate of the Google speech-to-text API on the recorded samples before and after processing. The change in error rate varies significantly between recording setups, but is positive in all three, with the highest improvement being 13 % in the first setup, and the lowest 3 %.

Chapter 6

Summary and Conclusions

This dissertation presented an evaluation of the advantages an eavesdropper would gain by combining the recordings made by multiple compromised mobile devices, with particular focus on a large-scale surveillance scenario. A theoretical model was presented, giving upper bounds on such a system's performance, followed by a description of an implementation of such a setup and the algorithms used to process the recordings.

The system was evaluated in three setups, with three devices placed in varying orientations at a distance of 2 m from the target. The gain obtained by processing the recordings, compared to the baseline of the highest-SNR recording, ranged from (3.3 ± 0.3) dB in the best-performing setup to (2.0 ± 0.3) dB in the worst-performing one.

The improvement in speech-to-text transcription accuracy was evaluated using two engines: DeepSpeech and the Google speech-to-text API. Results show that DeepSpeech is unsuitable for transcribing recordings made by a smartphone microphone, with an error rate above 97 % before processing. Additionally, no speech was recognised on 25 % of the inputs. The combined output does not bring these values to a level where they would be of practical use, but there is still a measurable improvement, with the number of inputs where no speech was recognised dropping to 10 %, and a 0.8 % reduction in

the word error rate.

An evaluation of a smaller subset of the recordings using the Google speech-to-text API shows that combining the recordings results in an improved word error rate, with improvements of 13 %, 7 % and 3 % in the three setups.

While the results have shown that there is a measurable advantage to integrating the recordings, the observed variations in the individual recordings' signal-to-noise ratios indicate that the potential for scaling such a setup to a larger number of devices is limited. Due to varying microphone qualities, their directionality and effects of reverberation, even when the devices are placed at the same distance from the target, large differences in the SNRs were observed. If a single device happens to be placed in a significantly more favourable location than the others, the quality of its recording will be near the upper bound of the available signal-to-noise ratio.

In conclusion, the benefits an eavesdropper gains by compromising multiple devices depends on their layout – if there is only a single device which provides a good-quality recording (note that this is not necessarily the one belonging to the target), its output cannot be improved significantly, and the access to the device is the main advantage. If there are multiple good-quality recordings available, however, the results of this project show that their combination will perform better than each individual microphone..

6.1 Future work

While this project demonstrated the advantage an eavesdropper can obtain by combining recordings from multiple compromised devices, the evaluation was performed under simplified circumstances, with no sound sources apart from the target speaker. In a practical deployment of such a system, the calibration phase becomes more complex, and a robust implementation would have to be able to cope with interference from other speakers and background noise. Further, the system was evaluated with a stationary target, removing the need for dynamic updating of the sound travel times.

Finally, the approach presented in this dissertation, based on computing a single combined audio sample from multiple recordings and transcribing it using an off-the-shelf speech-to-text engine, is not the only stage at which the recordings can be integrated. A possible alternative would be a specialised speech-to-text engine capable of processing multiple recordings, effectively shifting the stage where they are combined to the speech recognition phase.

Bibliography

[1] Ce Wang, S. Griebel, and M. Brandstein. Robust automatic video-conferencing with multiple cameras and microphones. In *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532)*, volume 3, pages 1585–1588 vol.3, July 2000.

[2] M. Zohourian, G. Enzner, and R. Martin. Binaural speaker localization integrated into an adaptive beamformer for hearing aids. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(3):515–528, March 2018.

[3] Sharon Gannot, Emmanuel Vincent, Shmulik Markovich-Golan, Alexey Ozerov, Sharon Gannot, Emmanuel Vincent, Shmulik Markovich-Golan, and Alexey Ozerov. A consolidated perspective on multimicrophone speech enhancement and source separation. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 25(4):692–730, 2017.

[4] C. Knapp and G. Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4):320–327, August 1976.

[5] Joseph H DiBiase, Harvey F Silverman, and Michael S Brandstein. Robust localization in reverberant rooms. In *Microphone Arrays*, pages 157–180. Springer, 2001.

[6] Douglas E Sturim, Michael S Brandstein, and Harvey F Silverman. Tracking multiple talkers using microphone-array measurements. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 371–374. IEEE, 1997.

[7] Nikolay D Gaubitch, W Bastiaan Kleijn, and Richard Heusdens. Auto-localization in ad-hoc microphone arrays. In *2013 IEEE International*

Conference on Acoustics, Speech and Signal Processing, pages 106–110. IEEE, 2013.

[8] Antonio Canclini, Fabio Antonacci, Augusto Sarti, and Stefano Tubaro. Acoustic source localization with distributed asynchronous microphone networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(2):439–443, 2012.

[9] Marius H Hennecke and Gernot A Fink. Towards acoustic self-localization of ad hoc smartphone arrays. In *2011 Joint Workshop on Hands-free Speech Communication and Microphone Arrays*, pages 127–132. IEEE, 2011.

[10] Nikolay D Gaubitch, Jorge Martinez, W Bastiaan Kleijn, and Richard Heusdens. On near-field beamforming with smartphone-based ad-hoc microphone arrays. In *2014 14th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 94–98. IEEE, 2014.

[11] Shun-ichi Amari, Andrzej Cichocki, and Howard Hua Yang. A new learning algorithm for blind signal separation. In *Advances in neural information processing systems*, pages 757–763, 1996.

[12] Satoshi Kurita, Hiroshi Saruwatari, Shoji Kajita, Kazuya Takeda, and Fumitada Itakura. Evaluation of blind signal separation method using directivity pattern under reverberant conditions. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 5, pages 3140–3143. IEEE, 2000.

[13] Keiko Ochi, Nobutaka Ono, Shigeki Miyabe, and Shoji Makino. Multi-talker speech recognition based on blind source separation with ad hoc microphone array using smartphones and cloud storage. In *INTERSPEECH*, pages 3369–3373, 2016.

[14] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[15] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. IEEE, 2015.