Security and Privacy in Speaker Recognition Systems

Henry Turner

Balliol College

University of Oxford

A Thesis Submitted for the Degree of

Doctor of Philosophy

Michaelmas 2021

Acknowledgements

This thesis would not have been possible without the help of many people. My work has been supported by the Centre for Doctoral Training in Cyber Security and funded by the EPSRC, with further support from Mastercard and Armasuisse.

Many thanks go to my supervisor, Ivan Martinovic, for his guidance and leadership, as well as to the rest of the research group for all of their inputs throughout. In particular I would like to thank Giulio Lovisotto and Simon Eberz for always being willing co-authors and to thank the other residents of RHB101 as well as Matt Smith for the entertainment over the years.

I would also like to thank my assessors throughout my DPhil, Kasper Rasmussen, Andrew Martin, and Patrick Traynor, for their invaluable feedback across the transfer, confirmation and final submission phases of my degree.

Finally, I would like to thank my family for their support throughout, in particular to my parents and brothers for supporting me throughout my entire academic journey, as well as my wife, Jenny, for always being a calming voice of reassurance and keeping me focused on the task at hand in the last stages of the process.

Abstract

Voice interfaces continue to grow in popularity, with standalone systems being deployed in our homes, smart assistants being added to our phones and smart watches, and voice based software being added to phone call systems. As part of this trend voice interfaces are also deploying personalised functionality, with the unique features of an individual's voice being used as a biometric to guard access to this.

In this thesis we examine the security and privacy aspects of these systems, with a particular focus on remotely accessed speaker recognition. First, we evaluate the susceptibility of speaker recognition systems to attacks by developing an attack method that allows an adversary to impersonate a chosen user. We demonstrate the capabilities of this attack across several different systems, showing that an adversary can still perform this method even when restricted to audio data of limited quantity and quality.

Having demonstrated the vulnerability of speaker recognition to attacks, we investigate methods to enhance the privacy of those using such systems. We first propose and evaluate a method of voice anonymisation, which removes identifying information but maintains the prosody of the speech. We follow this by developing an alternate method, which allows the user of a remote speech system to protect their own voice from capture, by replacing it with an alternate voice when they speak and removing all voice information from the final audio. We show that this method can be used to maintain the identity of the user over time, and is more resilient to attacks than the status quo for users with exposed voice traits.

Finally, we explore a new method for collecting biometric datasets. We design, implement and evaluate a system for collecting these datasets remotely. This allows researchers to scale their dataset collection more effectively, allowing for the creation of larger and more useful biometric datasets in the future.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Motivation

Once the realm of science fiction, voice interfaces have proliferated in the modern world and now occupy many homes, phones, and watches, as well as being used to add additional features to telephone networks. From a user's point of view, voice interfaces offer an attractive method of interacting with devices, being (generally) intuitive to use and allowing hands-free access. In authentication systems they have much the same benefits, reliving users of the need to remember cumbersome passwords or answers to secret questions, as well as offering a means of communication that almost everyone can naturally take advantage of. Speaker recognition may also increase system utility for users or service providers. It can add customised functionality for individuals, allow providers to track users across different sessions and allow analysis of identity for fraud detection. Furthermore if audio data is recorded at the time of interaction, then future analysis is possible.

However, as these voice interfaces become more prominent, it is important to question the security and privacy implications of them. This is particularly important, as compared to passwords and common biometrics such as fingerprints or face recognition, voice has several distinctive properties: (i) it is broadcast regularly to the world as part of normal life and the technology to capture it is cheaply available and ubiquitous, and (ii) limited special hardware is needed to replay a captured sample.

The first of these properties is possessed by some other biometrics, for example an individual face's could be considered to be broadcast regularly and the technology to capture the face is widely available (both cameras and infrared systems to capture depth). However a face is not easy to recreate and is likely to require advanced printing techniques, 3D models, and other mechanisms. Thus the threat to voice processing systems is much more pronounced, as the ability to capture someone's voice trait and replay it is built into the mobile phones that people carry with them each day, meaning that the ability to conduct attacks is likely greater than with many other biometric modalities. The existence of these two properties for voice even brings into question if it is suitable for use as a biometric: a secure biometric system requires a secret shared between the cooperating parties, a high resistance to forgery (such as is the case with face recognition), or ideally both of these things. Furthermore, unlike most biometrics, speaker recognition may also be deployed in remote contexts, such as on the receiving end of a telephone call. This changes the defensive techniques available to a service provider, ruling out common options such as using the biometric with trusted hardware to authenticate access to a channel, which is itself encrypted with traditional cryptography approaches.

These properties also have implications for the privacy of individuals as voice interfaces continue to capture more and more data, which, if exposed in some way, can lead to linkage attacks against individuals.

The goals of this thesis is therefore to investigate these two issues with voice interfaces. We first propose and develop an attack against speaker recognition, with a focus on developing an attack method that allows an adversary to impersonate a victim with limited captured voice data and of limited quality. Secondly we investigate solutions to protect the privacy of the voices of those interacting with voice processing systems. We do this from a provider context, by investigating techniques that can be used to anonymise voice data after capture and from a user context, by proposing and developing a system to allow a voice system user to replace their voice with an alternate voice as required.

In the final chapter of this thesis we turn our attention to the methodology for collecting datasets for biometric experiments. When conducting the work in Chapter 3 the process of collecting the voice dataset was laborious and time intensive. This is generally true of all biometric dataset collection. Furthermore the COVID-19 pandemic has highlighted difficulties in collecting datasets if external restrictions are imposed. As such we propose and develop a method for collecting biometric datasets remotely. We evaluate this new method and discuss its suitability for future research studies, in particular on voice datasets.

1.2 Ethical Considerations

The work contained in this thesis requires careful thought towards ethical issues throughout. Firstly we address ethical concerns related to the outcomes of the work. One potential concern is the development of an attack in Chapter 3, which we demonstrate to be effective against some existing systems. As far as we know these systems are not used for securing any systems currently, as they have been updated since this work was completed. Furthermore, we are unaware of any sensitive functionality being controlled by these APIs at any point. Additionally there are possible countermeasures to the attack, which we discuss in Section 3.6. Subsequent chapters of the thesis deal with privacy settings and have the intention of protecting users of voice processing systems. As such the ethical considerations for these are more limited. It could be argued that the voice protection system developed in Chapter 5 could be abused by unscrupulous individuals, but this is also the case for most other privacy protecting technologies.

The work also has numerous ethical considerations related to the use of human subjects throughout. Each time human subjects were required for any purpose we completed the ethical approval process within the university. In Chapter 3 we collect a voice dataset from human speakers. There is a risk of this audio being used to impersonate users in a myriad of contexts (as we demonstrate and discuss throughout the thesis) and as such it is imperative that this data is stored properly. As such we do not store the names of the participants with the data. The data is also stored in an encrypted form whenever it is not being used. Furthermore we do not (and can not as per our ethical review) release this dataset for further use.

We also conduct experiments where we obtain biometric datasets from human subjects twice in Chapter 6, firstly collecting photoplethysmogram[1] (PPG) and subsequently by collecting touch dynamics data. The PPG dataset again has biometric uses and could be used to identify the participants in some scenarios; again we respond to this by anonymising the data and storing it without identifying information for the participants. An additional ethical consideration with this dataset is the potential ability to identify heart conditions in participants by examining their pulse

---

[1] An optical technique to measure blood volume changes and by extension pulse.

traces from their PPG signal, however this is distinctly separate from our field of study so we do not consider it further.

Secondly a touch dynamics dataset was collected from participants. Whilst this dataset is not used in this thesis[2], ethical care was still taken when collecting this dataset. In particular this work was exploring the possibility of using Mechanical Turk to recruit participants for biometric dataset collection studies and as such care had to be taken to ensure informed consent was obtained from them in an appropriate manner. Likewise it was also important to ensure that data could not be linked back to individuals and that workers were compensated appropriately. In Chapter 6 there is a more detailed discussion around these ethical issues when conducting these kinds of experiments on Mechanical Turk.

Finally in Chapter 4 and Chapter 5 human participants were needed to give subjectivity scores for the quality of the audio produced by the systems, which we completed using Amazon Mechanical Turk to provide participants for our task. This is a fairly standardised process, with an ITU-T recommendation [67] covering how such subjectivity measurements should be conducted. We follow this recommendation when conducting our tests and only receive the scores the participants give to our produced audio, requiring no further data from them. We also ensure that participants are fairly compensated[3] whilst conducting the listening exercise. Participants are free to choose (or choose not) to participate in the study based on their personal feeling towards the compensation.

All of the studies mentioned above were approved by Oxford's Central University Research Ethics Committee (CUREC) process. The following reference numbers refer to each study conducted and are given again later in the thesis at the appropriate points:

- SSD/CUREC1A_CS_C1A_18_032 – Voice dataset.
- SSD/CUREC1A_CS_C1A_21_010 – Perception of audio quality.
- SSD/CUREC1A_CSC_1A_19_032 – PPG dataset.
- SSD/CUREC1A_CSC_1A_19_013 – Touch dynamics dataset.

---

[2] Only the development and evaluation of the collection method is a contribution of this thesis

[3] $0.70 per task, with a task taking less than 5 minutes.

1.3 Contributions of this Research

The section explains the contributions of the published work that underpins this thesis. All of the publications are joint work with various co-authors, however each chapter of the thesis is primarily based on published work for which I was the primary author and contributor.

- Setting the scene for the thesis, in Chapter 3 we develop a realistic attack against speaker recognition systems and investigate the quantities of audio necessary to perform our attack. The attack method modifies an existing speaker's voice to more closely resemble that of the target, when considered by a speaker recognition system. The results of this study were presented at the 2019 *European Symposium on Research in Computer Security (ESORICS)* [151].

- Chapter 4 proposes a mechanism for anonymising voice data after it has been recorded, such as might be required by a service provider who has collected some audio data. The method involves decomposing the voice into an identifying component (an x-vector) and several non-identifying components. The identity component is then replaced with a newly generated identity. This work was part of the Voice Privacy Challenge and presented as part of the *VoicePrivacy 2020 Virtual Workshop at Odyssey 2020* [152], with a journal extension of this paper accepted for publication in the *Computer Speech and Language Special Issue on Voice Privacy* [153].

- Chapter 5 also concerns voice privacy, but focuses on developing a system that can be used by an end user of a remote voice processing service. The developed system involves generating a new anonymous identity for a voice and performing sequential speech-to-text and text-to-speech with the new identity, resulting in the creation of an anonymous voice. A pre-print of a paper based on this work is available on arXiv [150].

- The onset of the global COVID-19 pandemic in March 2020 caused significant change to working environments around the world and in particular preventing many lab based studies from taking place. Fortunately we had already been exploring the possibility of conducting biometric dataset collection remotely, which we continued with in earnest during the pandemic. Chapter 6 develops methods for obtaining longitudinal biometric datasets remotely using Amazon Mechanical Turk. The chapter draws from my work designing the experimental application for our paper published in the *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* and presented at *15th IEEE Computer Society Workshop on Biometrics 2020* [89][4], as well as a paper based on an evaluation of using Mechanical Turk for collecting longitudinal biometric datasets. A preprint of this paper is available on *arXiv* [149].

1.4 Thesis Outline

The thesis is structured as follows:

- **Chapter 2** provides a background to the work of the thesis. We discuss some preliminary definitions used throughout the rest of the thesis, as well as discussing how existing voice processing systems work, attacks against them, countermeasures to these attacks, and existing voice privacy approaches, drawing on related work throughout.

- **Chapter 3** examines the resistance of speaker recognition systems to adversaries. In particular it develops an attack to impersonate a speaker and evaluates an adversary's ability to perform the attack under different assumptions for quality and quantity of original speech data. This highlights the vulnerability of users with unprotected voice traits to impersonation attacks.

- **Chapter 4** develops a system for voice anonymization. It aims to maintain the speech characteristics of the voice and only remove (and replace) parts of the signal that identify the original speaker, allowing for other speech analysis tasks. We discuss the overall anonymization capabilities of the system and identify future avenues for improvement.

- **Chapter 5** examines voice privacy from a user perspective and seeks to develop a system that enables a user to protect their voice trait whilst using a remotely accessed voice service. By developing a system that the voice service user deploys, they are given agency over the protection of their voice trait, and the ability to decide if and when to protect their voice when using remote voice services. We evaluate the suitability of our proposed system for this task across several variables, including the uniqueness of the generated voices and the naturalness of the resulting voices.

---

[4] I was the second author on this paper, contributing the design and implementation of the collection application.

- **Chapter 6** focuses on the dataset collection phase of the biometric experiment process. We explore the possibilities of collecting biometric datasets remotely, first evaluating its suitability for PPG, before undertaking a large scale dataset collection for touch dynamics. We evaluate the suitability of this experimental method for the overall task and outline ways in which future researchers can extend this technique to conduct their own studies.
- **Chapter 7** summarises the results of the thesis, arguing that the difficulties with remote speaker recognition for authentication may mean that it is better to forgo its use all together. It also discusses the future work necessary to continue improving the security and privacy of voice based systems.

Chapter 2

Background

Contents

In this chapter, we cover important background material and related work to contextualise the overall thesis. In particular we give definitions for common terminology used throughout the thesis, before covering the methods of modern voice processing systems. We follow this with related work on attacks against voice processing systems, methods to impersonate speakers and create artificial voices, existing attack countermeasures, and existing approaches to voice privacy.

2.1 Preliminary Definitions

When discussing speech based research, there are often words that can be used interchangeable for several different things, which can occasionally cause confusion. A clear example of this is the word 'speaker', which can refer to a person who is speaking, or an electronic device that produces audio output. As such, for the remainder of this thesis there are a set of key words which are used for the specific definition given below, in order to improve clarity. Definitions are also introduced for more obvious words in order to improve the readability of the thesis by giving a a clear definition.

Speaker: a human, who either has spoken and been recorded, or is currently speaking (live).

Loudspeaker: an electronic device that is used to play audio that has been previously collected.

Audio Source: a source of audio, such as a speaker or a loudspeaker.

Utterance: a single audio sample containing some text that has been spoken by a speaker.

Phrase: the textual content of an utterance.

Sample: a recording of some audio.

Voice Processing System: (VPS) a system that takes spoken voice data as input and performs some processing with it to produce some output.

Remote Voice Processing System: a voice processing system that is operated in a remote manner, for example over the phone, such that the audio source does not have to be physically located with the voice processing system.

Pseudo-speaker: an artificial speaker, that is an audio source that produces (something resembling) speech, with its own identity.

Keyphrase: a specific phrase that must be spoken (or presented in some way) to access a text-dependent speaker recognition system.

Speech Recognition System: a VPS that processes utterances presented to it and outputs (or attempts to) the textual content of those utterances.

Speaker Recognition System: a VPS that identifies the speaker of an utterance presented to it. This can either be authentication: is the identity of the speaker of this utterance the same as this claimed identity, or identification: which of this set of speakers is speaking.

Equal Error Rate: (ERR) is a statistic used to describe the performance of biometric systems. It defined as the value of the false accept rate (or false rejection rate) at the point when the threshold for acceptance makes the false accept rate and false rejection rate equal.

2.2 Voice Processing Systems

Almost all of the parts of this thesis involve voice processing systems in some capacity, hence the necessity for some understanding of voice processing systems. In this section we outline the background and related work for several key parts of voice processing systems.

2.2.1 Input Features

Many VPS take audio as their initial input. However raw audio data is fairly complex and as such most VPS extract features from this audio data, to provide something easier to work with. By far the most common features to work with are those derived from applying the Mel scale to audio data, the most prevalent of which are Mel Frequency Cepstrum Coefficients (MFCCs) and Mel Spectrograms. The Mel scale is devised so that sounds of equal distance from one another are perceived by a human listening to them to be the same 'distance' apart. The values of the scale have been determined through several subjective listening experiments, meaning several variations exist, but broadly speaking the conversion from frequency to Mel frequency is linear below 1kHz and logarithmic above this. To obtain values on the Mel scale, the audio signal is first sampled with overlapping windows, with Fast Fourier Transforms (FFTs) calculated for each window. The FFT components are then mapped to the mel scale, which can be output directly as a spectrogram.

More commonly MFCCs are selected as the input features, with several variations on the exact parameters being used. MFCC, as proposed by Mermelstein in [96], begins by segmenting the signal into (usually) overlapping windows of roughly 25ms in length[1]. Within each frame the periodgram estimate of the power spectrum of each frame is calculated by squaring the absolute values of the complex Fourier transform of the frame. The Mel filterbank is then applied to the power spectrum, with the energy summed for each filter, typically using between 20 and 40 filters. This results in one number per filter, giving an indication of how much energy was in that part of the spectrum. The log of the filterbank energies is then taken and the discrete cosine transform (DCT) performed on these log energies. The coefficients of the DCT are

---

[1] Length of windows and overlap can both be configured, but 25ms length and 10ms overlap is typical

the MFCC coefficients. Typically the first coefficient is removed and replaced with the energy of the whole frame and some of the higher coefficients of this DCT are discarded (typically coefficients greater than 13, but sometimes greater than 20). Additionally the first and second derivatives of the MFCC coefficients are often also included in the feature vector.

MFCCs have been used - and are still used - in many popular VPS implementations, several of which are described in the following sections. We also make use of MFCCs in Chapter 3, to develop our attack against speaker recognition systems.

2.2.2 Speech Recognition Systems

Speech recognition systems have long been seen as desirable, due to the ease with which humans can interact with them. They have often featured in science fiction, as a futuristic technology through which humans communicate with complex systems[2], and promote a natural style of interaction with machines. As such research into speech recognition systems is well established, and commercial products have existed for many years[3]. However, recent advances in computational power have led to rapid improvements in speech recognition technology, in both its ability to generalise to many speakers, requiring less fine tuning, and the speed with which it can be conducted.

Traditionally speech recognition systems have been based on Hidden Markov Models (HMMs) [48], which model speech as a series of state changes, which can then be converted into speech using acoustic models, language models and a pronunciation dictionary. The most well known of these traditional systems is the CMUSphinx system [80], an open source project that allows for the creation of such speech recognition systems. Other popular alternatives include HTK, the Hidden Markov Toolkit, which has also been used for extensive research into speech recognition [170].

Recently the performance of these systems has been surpassed by *Deep Neural Network* (DNN) methods, in particular the Deepspeech system [56]. These approaches require significantly less domain knowledge and acoustic engineering than more traditional methods. Deepspeech uses a recursive neural network architecture, which takes MFCCs as input and outputs a sequence of characters. These sequence of characters is then passed to a language model decoder, which converts this sequence of characters into a coherent utterance.

---

[2] For example, 2001: A Space Odyssey features HAL, a computer which interacts through voice, which was released in 1968

[3] The Dragon speech recognition system, which still exists today, was first released in 1990

Many commercial speech recognition systems have been deployed in the last few years, in particular in the form of personal assistants, such as Amazon Alexa, Google Home and Apple Siri. Whilst limited details of the workings of these systems are publicly available, we know that Siri uses a neural network approach [8] (at least for the detection of the keyword) and it is likely that the other systems do too. In comparison to traditional methods, the neural network approaches need significantly more data to train, but this is easy to acquire for large companies, as they can store audio that is used to interact with their existing systems, allowing them to rapidly grow large datasets.

In speech recognition systems designed for interaction with humans, the system also incorporates a Natural Language Understanding (NLU) system, which transforms the text into an intent. This process may also correct some errors from the speech recognition system, by using the intent to guide what might actually have been said. This NLU system is another avenue of attack for a potential adversary, and it has been demonstrated that it is possible for an attacker to create skills that take advantage of commonly misinterpreted words to trigger alternative actions on consumer devices [177].

2.2.3 Speaker Recognition Systems

Whilst speech recognition systems are usually essential to speaker recognition systems, the former have progressed faster than the latter. Despite the explosion in speech recognition enabled devices, many of these have yet to exploit the ability to add speaker recognition to add personalised features or authentication mechanisms. Amazon Alexa and Google Home both now support customised functionality for specific users, and Apple Siri uses speaker recognition on their wake commands for iOS devices, to provide some measure of protection from other users accessing people's Siri functionality, either inadvertently or intentionally [9]. The application of the system to the wake commands means that the recognition is *text-dependent*, meaning that a certain keyphrase is required for authentication. Generally text-dependent systems perform better (i.e. have lower equal error rates) than text-independent systems, where any text can be spoken.

However, these implementations of speaker identification are less than perfect, as evidenced by the recent decision to remove the ability to unlock with voice match on some models of Android phones [83], with the option now being a setting that comes with several security warnings when enabled. Whilst this degrades the overall experience of using voice interfaces by requiring users to conduct an out of band authentication for an action to be executed, the security of the voice interfaces is deemed week enough to mandate it. This is generally true of most devices that support personalised voice services, with them delegating to a stronger authentication method before performing sensitive actions.

Extensive research has been conducted into the best methods for modelling the features and conducting the speaker recognition itself. The Gaussian Mixture Model (GMM) was traditionally one of the most widely used methods [73, 122] and is often used as a benchmark for comparing other implementations. The GMM technique for speaker recognition makes use of a Universal Background Model, which is a GMM trained on all of the speaker data and thus models the feature distribution of speech in general. Individual GMM's are then derived from this GMM to model each speaker in the system. When classifying results we then use the log likelihood ratio between the model GMM and the UBM to determine the score, which is used to decide on who is speaking, or, in the case of authentication, is compared with a threshold to determine acceptance. These GMMs use several mixtures to model the voice of the speaker, through training by adapting the parameters of the Gaussian distribution that defines each of these mixtures.

Extensions to this GMM-based system have been suggested, of which the method based on the combination of i-vectors and probabilistic linear discriminant analysis has become the most widely used method [36]. The i-vector technique uses a matrix to model channel and speaker variability of the entire speech corpus, which is then used to generate a vector to represent a given speech sample. Channel compensation is performed and a distance measure (such as cosine distance) between two samples determines if they are spoken by the same speaker.

This approach achieves an EER of 1.12% on the NIST 2008 Speaker Recognition Evaluation data set. Some modern systems use the i-vector approach, for example the Microsoft Azure Speaker Recognition APIs [142], although as the actual implementation of this APIs is a trade secret it is possible that they have moved onto deep learning based methods since the publication of this blogpost. We use these APIs to evaluate our proposed attack in Chapter 3, as well as a GMM-UBM model.

As with speech recognition, recently deep learning approaches have also been applied to speaker recognition. For example Heigold et al. apply an RNN to the problem in [60], demonstrating that a deep neural network can achieve lower EERs than the i-vector based solutions for their "Ok Google" based data set and that a recurrent neural network can reduce the EER on their dataset further.

One of the most prominent methods are x-vectors [137]. Conceptually these are similar to i-vectors, with the final output layer of the network in training have a component for each of the speakers in training. At usage time this last layer of the network is removed, leaving the 512 dimension previous layer as the feature vector. Distance between these vectors is then calculated using PLDA in the paper, but can also be computed using other distance measures, such as cosine distance. The x-vector network is one of the strongest performing currently available and in particular responds well to data augmentation, where noisy audio is added to samples, and performs well when analysing short samples from speakers. In Chapter 4 we use an x-vector network for evaluating the performance of our proposed anonymization system on a speaker recognition system.

Other similar networks have been devised that also create the feature vector in the same way as the x-vector network. Wan et al. propose a network trained with Generalized End-to-End loss (GE2E) [160]. The network is trained using a batch of M utterances from N speakers, creating a similarity matrix for each batch between the network output feature vector of each utterance and the centroid of the output feature vector for each speaker. The network then learns parameters with a loss function that maximises the similarity with utterances and their own speakers centroid and minimises the similarity for utterances and centroids from different speakers. The cosine similarity is used as the distance similarity metric for the computation of the matrix. This network can be used for text-dependant or text-independent analysis, with cosine similarity being used to compare pairs of feature vectors output by the network. Experimental results give EERs of 3.55% for text-independent and as low as 2.38% for text-dependent verification. We use the GE2E network in Chapter 5 when developing our system for creating private voices.

2.3 Attacks on Voice Processing Systems

2.3.1 Speech Recognition

As might be expected, the use of voice of an input mechanism has led to work studying malicious inputs to these devices.

Early work in this area has been focused on *hidden voice* commands. These commands are audio samples, that when played to a device are recognised as a specific phrase, but which a human can not understand. This allows for a command to be played without any observers knowing what has been played.

Vaidya et al. [155] first demonstrated this with the use of an audio mangler, which extracts MFCCs from the voice sample, before inverting the transformed coefficients to generate audio. This is applied repeatedly until the output is unintelligible to humans, yet the original features remain and thus the audio is interpreted correctly by the system. Evaluation of the technique used human listeners provided by Mechanical Turk to transcribe the audio samples, demonstrating that the ability to transcribe the samples correctly was significantly worse for the altered audio.

This work is extended in [23], which demonstrates that an attacker can perform this attack in more realistic conditions, such as against black-box models and in the presence of background noise, while retaining the non-intelligibility of the commands (for humans). This is done by applying the techniques of [155] under more practical settings, as well as using a machine learning-based approach for demonstration against a white box model.

Yuan et al. [171] demonstrated a technique which allows an adversary to embed speech commands into songs, further reducing the suspiciousness of this attack, as now a human listener would be listening to a song and not notice the presence of malicious audio. In [24] this idea is extended by demonstrating an attack which tricks a speech recognition system into believing any pre-chosen command was present in a given audio sample, despite it not being perceivable by humans. This includes situations where a human hears a particular utterance, but the speech recognition system transcribes something completely different.

A slightly different approach is taken in [174], where Zhang et al. demonstrate an attack using commands which lie entirely in (human-)inaudible frequency range, but which are still heard and interpreted by voice assistants. This works by using audio in the ultrasonic frequencies which are then interpreted as normal audio in certain frequencies, due to non-linearity of the amplifier used in the microphone. They also demonstrate that although the non-linearity is different for each specific microphone, they occur in wide ranges for a given device, meaning that attacks can be effective against a family of devices.

Subsequently, Abdullah et al. developed a technique which makes hidden voice attacks more practical, by exploiting knowledge of the MFCC signal processing algorithm, as opposed to computationally intensive GPU-based optimisation problems [1]. The work relies on the idea that almost all speech recognition systems use MFCC features and that several transforms can be applied to any given sample of audio, yet the same MFCC coefficients yielded, as MFCC feature extraction is a many-to-one function. As such they demonstrate four different types of perturbation can be applied to the audio and used to produce adversarial examples that work against a wide variety of models, including some publicly deployed APIs. This demonstrates that adversary capabilities do not necessarily need to have access to high computational power to generate successful examples and that the feature extraction algorithms used in conjunction with the advanced neural network based models can also be exploited.

Finally, several recent works have tried to use the principles of pyschoacoustics to create adversarial examples that are indistinguishable from the original audio when played to human listeners. In [127], Schönherr et al. apply psychoacoustic modelling to add an extra layer of back-propagation to their adversarial example generation. Their psychoacoustic model is based on the MP3 compression algorithm, which used a set of empirical tests to determine which frequencies can be made inaudible, whilst remaining imperceptible to humans. The system was tested against the Kaldi speech recognition system, which uses a combination of neural networks but also a hidden Markov model for classification, as is common in speaker recognition. Their results demonstrate that it is possible to find an adversarial example that will transcribe to a specific text, whilst only adding a small amount of overall noise. Furthermore, human listening tests on the original audio and the adversarial audio, showed only a minimal change in the Word Error Rate (12.59% original vs 12.61% adversarial), indicating humans could not detect the changes to the audio. The technique was not tested with over-the-air playing of samples i.e where audio samples were played over a loudspeaker and recaptured.

Following on from, this Qin et al. applied similar psychoacoustic principles to systems made entirely of neural network components and developed their samples to be robust to playing over the air [119]. This is done by creating simulated acoustic impulse responses of many rooms, which can be applied as a function to any audio. The loss function for the adversarial example generation is then reformulated to minimise the loss over several different simulated environments, thus introducing robustness. An adversarial example is only considered successful, in the generation phase, if it fools a set of randomly generated acoustic environments. The imperceptibility and robustness directly compete with each other, in that increasing robustness also increased perceivability, but it is still possible to create adversarial examples that are both robust and imperceptible. The evaluation results demonstrated that users found the audio cleaner than earlier work on adversarial examples and harder to distinguish from the original audio. Testing the adversarial examples after synthetic room simulation also demonstrated that the examples were still highly effective in many differing environments.

Since adversarial example were first applied to automatic speech recognition systems, great improvements have been made in both the quality of examples, ease of production, robustness and imperceptibility. The latest works in the area have shown that it is possible to create adversarial examples easily and with minimal computation resources, which a human can tell have been altered yet can't interpret [1]; or that intensive back propagation can be used to generate examples that humans can not distinguish from regular audio, but which are still transcribed as targeted [119]. Overall, the research in this area has matured significantly, with future work likely to focus on making things faster, less computationally intensive or more practical.

2.3.2 Speaker Impersonation

There is a large body of work, originating from within the speech community, that focuses on speaker impersonation. Impersonation attacks can be categorised into four types [42]:

1. Direct Impersonation - where an adversary attempts to impersonate a victim by using a different voice from normal voice.
2. Replay - where an adversary captures a sample of a specific user speaking and plays it back to the speaker recognition systems.
3. Voice Conversion - where the voice of some source user saying a specific phrase is converted into that of the target speaker, to sounds as if the target has said the phrase.
4. Speech synthesis - a model is trained, such that text is converted into speech that sounds as though it comes from the target speaker.

As the research on each of these has generally emerged from the speech community, the aim of each of these types of attack has usually been to fool a human listener. This is not the same as deceiving a voice processing system, as the former cares about things such as naturalness, whereas a VPS is only concerned with the output of the algorithm it performs on the supplied audio. This difference gives greater freedom to an attacker targeting a VPS, in terms of audio mangling and manipulation, than if they are deceiving a human.

In impersonation attacks, also known as *mimicry*, human impersonators attempt to alter their own voice in order to mimic another person's voice [81]. Lau et al. demonstrated that this was possible in [81], but this attack vector is typically ignored, as the natural countermeasure to it is to produce more accurate systems, thus making impersonation harder. However, this countermeasure can never render an attack impossible using this method and this is a permanent limitation of voice systems.

Replay attacks involve replaying (with a loudspeaker) audio samples to the system, either in whole or by cutting up original audio files and splicing them together [85].

Many works have addressed the problem of voice conversion and recently the popularity of the Voice Conversion Challenge [143] gave way to a numerous set of works [41, 65, 75, 93, 140]. The challenge provided a parallel dataset, on which participants trained voice conversion algorithms, which they then generated audio with to be tested by a speaker recognition system. Approaches such as [93] use a probabilistic mapping of vocal tract models to convert between speakers, where as [75] use a Gaussian Mixture Model (GMM) trained on aligned audio from victim and attacker, which can then be applied to the source audio. Kobayashi et al. propose a differential voice conversion technique (DIFFVC) in [76], which learns a GMM of the differences between two sets of MFCC parameters, from a parallel data set, which is then applied to the voice to be transformed, in conjunction with some additional transformation of the fundamental frequency.

These approaches are all complex, typically requiring at least one of: large amounts of audio, parallel utterances, or fine tuning of model parameters. As such, although they can deceive some speaker recognition systems, they do not have much real world practicality. These works originate from within the speech research community, where the aim of the challenge was to improve the quality of voice conversion systems, as opposed to create meaningful attacks.

Speech synthesis aims to create a model for generating completely artificial speech. In [35] De Leon et al. proposed a technique based on a Hidden Markov Model (HMM), which adapts a background model in order to derive an audio synthesizer. The analysis shows that such a synthesizer can impersonate users in the well known Gaussian mixture model [13] 81% of the time. The WORLD system improved on these further, creating a real-time vocoder based speech synthesis system [98]. The WORLD approach takes the waveform and uses this determine the fundamental frequency, the spectral envelope, and the aperiodic parameter of the audio. These three components can then be combined together to produce synthetic audio. The WORLD system is used when conducting statistical parametric speech synthesis

Lately significant work has been conducted on both concatenative speech synthesis and wave-based speech synthesis. Concatenative speech synthesis is the system used by many popular digital assistants, such as Apple's Siri and uses a large database of snippets of audio, which use a selection algorithm to select units that can be fitted together to form the words required [22].

In comparison to this, wave-based systems operate on the raw waveform and have only recently become possible due to the introduction of Generative Adversarial Networks (GANs) [50]. In general these systems operate in two parts, with a *synthesizer* and a *vocoder*. The synthesizer takes textual input and outputs a prediction of mel spectrogram frames. The vocoder then takes these frames and turns them into audio

The GAN architecture is applied to the vocoder and pits two neural networks against each other to train itself to produce fake audio: the *discriminator* and the *generator*. The networks are trained one after another, with the generator first creating fake audio that can deceive the discriminator. These new fake samples are then fed into the discriminator to improve it, at which point the process starts again, forcing the generator to learn to create better and better fake samples.

Several GANs have been proposed for this such as Wavenet [156] and MelGAN [169]. In the paper introducing Wavenet, Oord et al. demonstrate that it can be used to generate synthetic speech which they experimentally verify to be better sounding than parametric or statistical speech synthesis approaches. They also demonstrate that the architecture can generate convincing sounding musical fragments. MelGAN is a further improvement on Wavenet to improve speech quality and execution speed, by improving the loss function and training methods.

The Tacotron 2 system uses the Wavenet architecture as part of its end to end text-to-speech framework [130]. The system uses an initial feature prediction network to convert text into predicted Mel-frequency spectograms. It then subsequently uses a Wavenet-based network to turn these Mel frequency spectograms into speech. This speech receives high opinion scores when tested on humans. However, the system needs a lot of data to train both parts and will only produce audio that sounds like the person the training data was from. It is possible to swap the vocoder component in Tacotron 2 for other networks, such as MelGAN, in order to improve the quality of the resulting speech.

Both speech synthesis and voice conversion approaches have been shown to achieve good results in re-creating a person's voice. However, these approaches are designed for non-adversarial scenarios, where conspicuous amounts of high-quality audio for each speaker are available to train large (or deep neural network) models. Voice conversion approaches additionally often require labelled, or parallel training data: both source and target speaker uttering the same sentences (speaking the same transcripts), so that a model can be trained by mapping them on a one-to-one basis. Furthermore, these approaches are generally targeted at fooling human listeners, imposing many constraints on how realistic the voice sounds. In an adversarial scenario, under certain threat models, we are free of this constraint, either because the attack can be conducted out of range of human listeners, or because other techniques, such as hidden voice commands, can also be added.

2.4 Attack Countermeasures and Detection

Whilst we do not develop specific attack countermeasures in the course of this work, it is useful to have an understanding of the overall state of the art in this area. In particular the existence of these systems also has implications for the voice privacy system we develop in Chapter 5, as they may prevent the synthetic audio generated by our system from being used.

2.4.1 Audio Specific Approaches

Audio attack countermeasures and detection methods have been developed for as long as the attacks have been devised. Within the speech recognition community the Automatic Speaker Verification Spoofing and Countermeasures Challenges (ASVSpoof) [164, 165] have been a large driver in developing effective countermeasures against synthetic voice and replay attacks.

The 2015 version of the ASVSpoof challenge focused on distinguishing between human and synthetic speech (generated by voice conversion or speech synthesis) [164]. The database for the challenge consisted of synthetic audio generated with 10 different techniques, of which 5 are unseen until evaluation. The best approach achieved an EER of 2.013% against the unknown attacks in the test set [114], which uses a combination of Cochlear Filter Cepstral Coefficients (CFCC), combined with a change in frequency feature to augment MFCC to detect synthetic audio, which does a poor job in re-creating these two features. Other successful approach used a variety of techniques, such as deep neural networks [26], to attain similarly impressive performance at detecting the synthetic speech. Interestingly, within the 5 unknown synthetic voice samples, four were voice conversion and one was a text-to-speech system. The countermeasures all performed significantly worse against the samples from the text-to-speech system than they did from the voice conversion systems, e.g for the best solution [26] an EER of 8.5% for the synthetic speech, versus an average of 0.39% for the voice conversion approaches.

The 2017 version of the ASVSpoof challenge focused on replay detection [165], using replayed speech audio from 42 different speakers in total, with 24 unseen speakers for the evaluation. In total, 49 systems were entered into the challenge, with the best performing achieving equal error rates of 6.7% [74]. The best solution [82] was based on a convolutional neural network approach and used a normalised log power magnitude spectrum as the input feature vector. Since the end of the challenge, Tom et al. have shown that by using an image based neural network approach, which represents the signal as a group delay-gram, an equal error rate of 0% can be achieved on the ASVspoof17 dataset [145].

Outside of the challenge, Fang et al. have also designed a Generative Adversarial Network (GAN), which enhances the replayed audio [44]. The enhancements makes the baseline and the best neural network architecture from the challenge perform significantly worse, with the EER decreasing by a factor of up to 2.6.

From within the security community several approaches to audio attack detection have been discussed. Several of the attack papers proposed potential countermeasures within them. For example Zhang et al. [174] suggest that their DolphinAttack can be prevented by removing ultrasonic frequencies, whilst Yuan et al. suggest two defences against their CommanderSong attack [171], which are both effective in reducing its success.

A more general promising approach has been proposed by [14], which uses a feature that can distinguish between audio played through a loudspeaker and audio produced by a human. This technique relies on the observation that loudspeakers can not reproduce low frequency signals accurately, in part due to their shape and as such create additional energy in sub-bass regions that is not present in organic speech. This approach achieves a true positive rate of 100% in low-noise and 95.7% in high-noise environments, whilst setting the threshold for activation to maintain a false-positive rate of 5% or below (i.e 1 in 20 or fewer legitimate commands rejected). This feature should be applicable to a wide range of scenarios, although it may struggle in remote authentication scenarios over telephone networks, due to the frequency loss that occurs. For example, the original telephony protocol uses a frequency band of approximately 300Hz to 3400Hz[4], while Wideband Audio uses from 50Hz to between 7kHz and 21kHz.

---

[4] This frequency range excludes the fundamental frequency of human speech and relies on enough harmonics being present for the human listener to believe the fundamental frequency is transmitted

Other techniques have also been proposed that are applicable in certain situations and assumptions. Zhang et al. create the system VoiceLive in [176], which uses dual microphones present on a smartphone to calculate Time Difference of Arrival (TDoA) as a means to validate that the voice is from a human and not replayed. When a human speaks, the different phoneme sounds are produced in different parts of the throat and mouth. This difference in vocalisation location can be detected by the dual microphone solution, as sounds produced in differing places will have a different TDoA, whereas sound being replayed comes from the same location for all phonemes, and as such there is no TDoA. The solution yields high success rates (99% detection accuracy at 1% EER), but due to the dependence on requiring stereo microphone recording and the input microphones needing to be relatively near the mouth (less than 10cm), it is not applicable in all situations, such as remote authentication scenarios over the telephone or smart-home assistants. A similar dual microphone system is presented by Blue et al in [15], which captures the speech data on a second device as well as the smart home device. The audio captured on this second device can then be compared with the smart home device and analysed using TDoA and robust audio hashing to eliminate replay attacks from the environment.

Another proposed technique is the use of the phone as a Doppler radar, by Zhang et al. [175]. They use the loudspeaker on a phone to emit a high frequency sound, which humans can not hear, which is then reflected back at the same time as the humans speech to the microphone. As humans produce different phonemes by moving their mouth and face in particular ways, a different Doppler effect can be detected based on the phoneme being spoken, and this can also be used to improve any speaker recognition taking place. As a liveness detection technique it achieves 99% detection accuracy at an equal error rate of 1%. However, it can not be used in all situations, as it requires the loudspeaker to be near the speaker and microphone e.g just a few centimetres. As such it is only applicable for phone based environments, but only ones in which the system has access to the loudspeaker and the ability to emit specific frequencies from it, with a guarantee that they will be reproduced accurately, which is not the case over telephony protocols. As such it is only usable for applications running directly on the device and not remote services.

Pop noise has been suggested as a potential solution for liveness detection within the speech research community [94, 132]. Pop noise is created when humans produce plosive sounds (most significantly the letter p), which produces a rush of air from the mouth. If the microphone recording the speech is near enough to the mouth, then this rush of air deflects the microphones membrane and is detectable in the audio recording. Whilst this technique shows positive results, it may be possible to enhance synthetic speech generation systems, or replay audio, by adding synthetic pop effects before replay. However, as shown by Blue et al. in their liveness detection mechanism, loudspeakers can't produce low frequencies accurately [14], so it may be that loudspeakers can not produce these pop noises (which happen at a low frequency) accurately either. This technique has also been proposed within the security community [161], with experiments showing that the technique is resistant to various spoofing attacks, and that the relationship between pop noise and a phoneme is unique for individuals. This suggests that in situations where pop noises are reliably produced pop noise could be a good liveness detection mechanism.

2.5 Voice Privacy

In Chapters 4 and 5 we develop systems for voice privacy, with the former operating as an anonymisation scheme on the audio, whilst the latter is more similar to cancellable biometrics for voices. In this section we discuss the related work of these approaches to voice privacy.

2.5.1 Speaker Anonymisation

Speaker anonymization has its origins in analogue processing systems, when methods for securing and encrypting voices were first developed (such as [32]). Recently, analogue methods have become less relevant, as modern machine learning based voice systems take place on data in the digital domain.

Jin et al. [70] presented a speaker anonymization system that uses voice conversion to transform speaker's voices to a new special speaker identity. The approach required parallel training data for each speaker, restricting its use somewhat. This is essentially anonymising by applying voice conversion so that all speakers become the same identity. This obviously has some downsides, in that audio produced by two different individuals can no longer be differentiated after anonymisation (assuming the system works perfectly).

A GMM based approach was proposed in [116], which transformed user voices to a synthetic target voice using a combination of GMM mapping and harmonic plus stochastic models. This method achieved de-identification on 87.4% of samples, albeit with a limited database size of 10 speakers. The generated audio also lacked naturalness, due to the synthetic target speaker. [3] improved on this by transforming the speaker to one of several voices from a pool of speakers, where the target speaker to be transformed to is selected to maximise de-identification performance.

Magrinos et al. [90] improves on these works by using a cepstral frequency warping transformation based approach. A transformation function is applied in the spectral domain, de-identifying the voice. The inverse transform can later be applied to recover the original voice. Target speakers are selected to be the most dissimilar speaker to the original (based on PLDA distance between i-vectors).

This is extended to use CNNs in [11], which transforms the speech to a new voice, created from a set of transformation features from a source voice and a voice database.

Fang et al. [43] advanced the area further, presenting an approach based on decomposing the audio into identifying (x-vector) and non-identifying components, before replacing the identity component and re-creating the audio. This work forms the basis of one of the systems in the Voice Privacy Challenge, as well as our proposed anonymisation system, which we discuss further in Chapter 4. Srivastava et al. [138] further explore this system, examining the anonymization impacts of different parameters on x-vector selection.

2.5.2 Other Approaches

Recently other approaches have been proposed to introduce various degrees of privacy into voice systems. Qian et al. [118] introduced VoiceMask, a middle layer between speech services and the user to protect their voice's privacy, which uses vocal tract length normalization (VLTN) to warp the audio. This prevents re-identification of the original speaker, but does not consider an adaptive attacker and does not study the use of speaker recognition systems with the anonymised voice (or aim to support this). Srivastava et al. [139] perform an evaluation of this approach, as well as a VLTN based voice conversion approach [141] and a voice conversion approach using autoencoders [29], under threat models with informed attackers. The authors find that they can reach similar performance to baseline systems if they have full access to the private voice system. This demonstrates that speaker identifying information has been retained through the technique.

Han et al. [54] propose a scheme for a privacy-preserving release of speech data, in which voice prints of the users in the dataset are protected. This uses X-vectors as the voice print, perturbing the xvector using an application of differential privacy to voices, termed voice-indistinguishability. This approach is not usable by an end user, again requiring an end-user to trust the service provider to protect their voice appropriately.

Abdullah et al. [2] propose a method to attack widespread automatic speech recognition by mass surveillance systems by perturbing the audio at the word or phoneme levels using signal decomposition and reconstruction. This prevents STT systems from inferring the textual content and has some effect on disrupting speaker identification, however their is no perceptible change to the audio for humans, leaving users still vulnerable to more basic privacy invasions. Similarly the attack does not try to move the voice to a specific user and as such could not be used for any service that must be used repeatedly with the same identity, as the system we propose in Chapter 5 attempts to do.

2.5.3 Template Protection & Cancellable Biometrics

In biometric systems, recognition is typically performed using *templates*, produced by feature extraction algorithms applied to data obtained from a sensor [107]. The template(s) generated at enrolment are then compared with a template generated at authentication time to determine if the user is who they claim to be.

Template protection methods are applied to the templates stored in the system, in order to prevent the loss of a users templates to an attacker resulting in the ability to impersonate users. This is done through the introduction of three specific properties: *noninvertibility* - it should be computationally difficult to recover an individual's biometric template from a leaked protected one, *revocability* - it should be possible to invalidate a template and as a consequence losing one template doesn't compromise future templates for an individual, and *nonlinkability* - it is computationally hard to determine if two protected templates are derived from the same user [107].

In a cancellable biometric system [120], a fixed (per-user) distortion is applied to the biometric signal - either at a signal or feature level - and the remainder of the biometric system uses this distorted signal. Many cancellable biometric systems have been proposed for a variety of modalities, such as fingerprint([69]), iris([178]) and face ([18]) based systems.

Within cancellable biometric systems there are two main types of system [121]. *Non-invertible transform* based systems are those that use a specific non-invertible transform to produce the cancellable effect. Changes in parameters enable the updating of templates and it does not matter if the parameters used are exposed. In contrast to this, *biometric salting* uses invertible transforms, the parameters for which are kept secret and are in many ways similar to using a salt in a normal password system.

Very few existing cancellable biometric systems have been introduced for voice biometrics. A system has been proposed for voiceprints [167], which uses a group signature scheme to protect the voice with a key, allowing a voice to be used as a signature. In [115] a technique for converting voice prints from a speaker recognition systems into a binary representation is proposed. Mtibaa et al. [102] propose a shuffling technique applied to a binary representation from a similar model to provide a cancellable voice biometric implementation at a system level. These approaches have two drawbacks: i) the binary technique is developed on earlier systems and may not transfer to deep learning based methods, and ii) the protection only exists if the system creator decides to use it, again requiring users to trust all voice services they use.

Chapter 3

An Attack on Speaker Recognition Systems

Contents

Following on from the background in Chapter 2, we begin by developing an attack technique for speaker recognition systems. We focus on developing an attack that is practical for an adversary to implement, particularly when they are limited in the quantity and quality of audio data that is available to them.

3.1 Introduction

The goal of this chapter is to develop a realistic attack against speaker recognition systems. As voice interfaces become more popular, voice-based devices are now adding speaker recognition to their capabilities, so they can understand both *what* has been said (speech recognition) and *who* has said it (speaker recognition). Speaker recognition allows for customised functionality, as well as authentication, removing the burden of other less user-friendly authentication approaches (e.g., PINs or passwords). Nowadays, speaker recognition is available in commercial products such as Google Home [52] or Apple Siri [9]. Additionally, speaker recognition is increasingly deployed for over the phone authentication by companies in the financial sector (e.g. HSBC [64], Lloyds Bank [88]).

The majority of work in this space has been adaptions of techniques aimed at voice conversion or speech synthesis [35, 41, 65, 75, 93, 140]. Generally the training audio is collected in a well-isolated studio environment and the ultimate goal of the generated audio is to deceive a human listener, as opposed to deceiving a speaker recognition system.

However, from an adversarial perspective, obtaining audio of spoken utterances could be suspicious or unfeasible for certain victims. The unavailability of long samples of victim audio brings two limitations in re-creating the victim's voice: (i) models based on parallel datasets for voice conversion can not be used and (ii) synthesizers or conversion methods based on deep models do not reach sufficient accuracy, as intra-user variability is not efficiently captured. A detailed analysis of related work is given in Section 2.3.

As a result we constrain the datasets available to us for our attack and investigate the audio quantity and quality required to complete an attack successfully. Both of these factors are pertinent to attack feasibility, as both of these variables strongly impact the ease of carrying out such an attack.

The proposed voice conversion attack manipulates individual phonemes from a source voice into sounding like those of a target voice (when seen by a speaker recog-

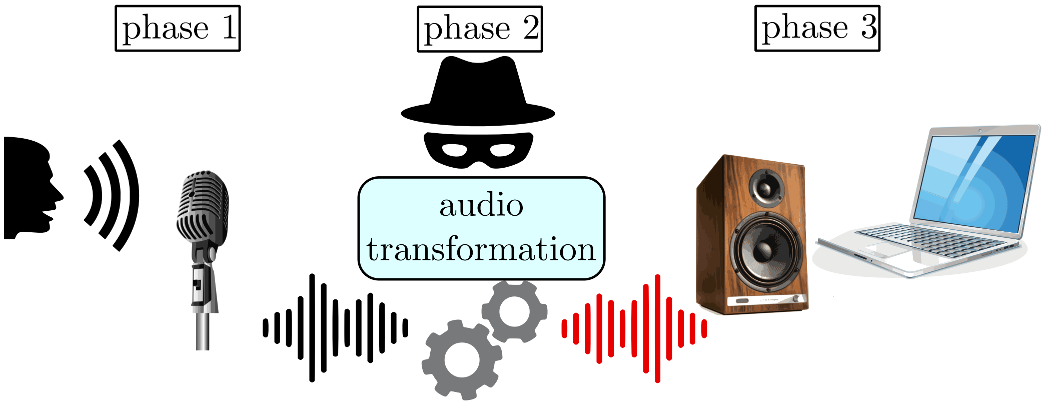


Figure 3.1: Threat model. The adversary initially records audio of the victim. This is used to create a transformation, which is applied to some source audio and replayed to the device (e.g. with a loudspeaker).

nition system). The transformation is based on morphing phoneme-related features in the Mel frequency cepstrum space [96]. Our transformation only requires knowledge of the number of phonemes in the target language and a piece of audio from the victim. We show that an adversary implementing the attack can improve performance by using a population of candidate source voices, as some voices are better at being transformed into others. We provide a method of identifying which source voices are likely to succeed in impersonating a target voice.

We evaluate the success of our attack against several different speaker recognition systems, in particular against the Spear toolkit [72], the Microsoft Azure Speaker Recognition APIs[1] and Apple iOS Siri. We use the Spear white-box model to learn how to improve the voice conversion and we show that the attack can successfully fool the black-box Microsoft and Apple models in both *over-the-wire* and *over-the-air* access. We evaluate the success of the attack across several sets of assumptions for the adversary, including (i) amount of known audio and (ii) recording noise, as well as for text-dependant and text-independent speaker recognition systems.

3.2 Threat Model

The phases of an attack are shown in Figure 3.1. The adversary first records the victim speaking and then constructs a mapping function between another individual's voice and the victim's voice. This mapping function is then applied to a sample of the individual speaking the desired phrase, resulting in a sample that appears (to a speaker recognition system) to now be spoken by the victim. The adversary replays

---

[1] https://azure.microsoft.com/en-us/services/cognitive-services/

the transformed audio sample to the system, with the goal of impersonating the victim.

Background. Users interact with a speaker recognition system, which performs either verification or identification. In the case of verification, the system requires users to utter a specific keyphrase, whilst for identification any utterance can be used. As an example, a laptop could use speaker authentication with the "Hey Siri, it's me" keyphrase in order to be unlocked (rather than typing a password). The keyphrase could either be fixed or contain a challenge, such as asking to speak today's date or utter a set of numbers being shown on the screen at the time of authentication.

Capabilities. Adversaries can: (i) record audio of the victim talking, (ii) replay audio to the voice recognition system (e.g. with a loudspeaker).

Knowledge. Following from the capabilities, adversaries have some knowledge of the victims voice trait (from recording audio of them talking). Additionally, the adversary has a set of audio samples containing spoken words for a population of individuals. This can be achieved easily by utilising free speech datasets such as VoxForge [158]. However, adversaries are limited along the following dimensions:

1. *black-box model*: adversaries do not know what voice processing and recognition algorithms are in place and thus cannot optimise their attack for a specific method.
2. *recorded utterances*: adversaries cannot record victims uttering the exact keyphrase required for authentication, nor its individual words (not all of them). This is straightforward when the keyphrase includes a challenge, but also reasonable when it does not. Keyphrases are typically designed so that they do not occur in normal day-to-day speech to avoid unwanted authentications.
3. *audio quality*: adversaries may only be able to record audio in public settings. This means that the recorded audio would have poor quality, as it involves a combination of (i) background noise, (ii) recording from a distance, (iii) recorded audio being emitted by loudspeakers rather than victims themselves.
4. *audio duration*: adversaries can only record the victim for a limited amount of time before raising suspicion. Consequently, they might have a weak representation of victims vocal characteristics, increasing modelling difficulty.

Scenarios. Following from the considerations of the previous paragraphs, we define three different scenarios that represent realistic attack situations.

- *Conference*: the attacker is attending a conference where the victim is giving a talk and records the victim speaking during their talk. The recorded audio is not of the victim directly, but is a recording of the room loudspeakers connected to the victim's microphone.

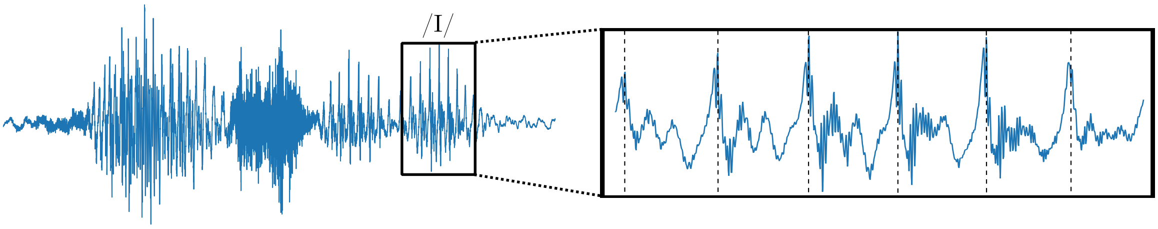


Figure 3.2: Sound wave of the utterance "Hey Siri". Within the same phoneme (/ɪ/) a wave pattern repeats itself, depending on the fundamental frequency of voice [45].

- *Cafe*: the attacker is at the same cafe where the victim is enjoying a coffee while having a conversation with other people. The victim's audio is recorded from a distance and is subject to background noise.
- *Ideal*: the attacker obtains high quality audio of the victim from the internet and uses it for their attack. The audio is extracted from a source such as a podcast, or a video of the victim speaking.

All attackers finalise the attack by playing their generated audio to the device. If attackers want to avoid detection, depending on the scenario, they can wait for the device to be left unattended before replaying audio to the device. These adversaries guide our experimental design. We further discuss how we model them in Section 3.4.

3.3 Attack Method

Overview. We construct the attack using the concept of phonemes, which are the individually perceivable units of sound in spoken language. We show in Figure 3.2 how phonemes appear in an audio wave of a spoken word: each phoneme is composed of a repeating wave pattern. The attack aims to transform each of these phoneme-related patterns so that they closely resemble the victim's. This is done by deriving a function which maps phonemes spoken by a known speaker into phonemes that resembles those spoken by the victim. The strength of using such an approach is that all knowledge requirements about the structure of the spoken language are removed. This way, an attacker can also afford to ignore the relationship between these phonemes and utterances (i.e. whether a particular phoneme occurs in an audio sample). In fact, no phoneme extraction is necessary, knowing the approximate number of phonemes for the language is sufficient (in spoken British English there are 44 phonemes [61]). We construct the mapping in the MFCC domain, as opposed to modifying the raw audio wave. We show that mapping the outputs of the MFCC extraction and reconstructing the audio wave afterwards is sufficient for the transformation to work.

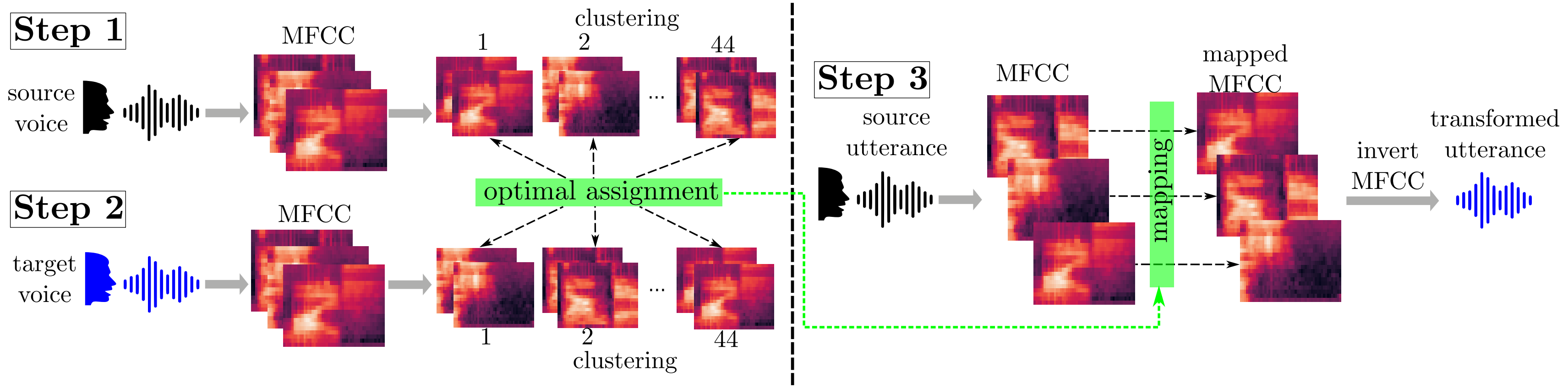


Figure 3.3: Steps to craft transformed utterances. In the first and second step the adversary computes the optimal mapping between the source and the target phonemes, in the third step they use the mapping to transform a specific utterance from the source.

3.3.1 Formulation

Given two speakers S and T (*source* and *target*) and a set of known audio recordings produced by them s_i, t_j, the transformation works as follows. Initially, the audio recordings are transformed into the MFCC spectrum, for a single audio file a we obtain a set of samples (due to the windowing process) as follows:

$$\textsc{mfcc}(a) = \{m_0^{(a)}, \ldots, m_n^{(a)}\} \tag{3.1}$$

where the number of points n depends on the audio length. We extract MFCC features for all audio recordings a_i belonging to a speaker. We apply K-means clustering, where K is the number of phonemes in the language, ($K = 44$ in our case) on all the samples (separately for T and S) to infer the clusters $C_k^{(S)}, C_k^{(T)}$, where each cluster represents a phoneme. With the clusters, we also obtain the cluster centroids

$$C_S = \{s_1, \ldots, s_K\} \text{ and } C_T = \{t_1, \ldots, t_K\}. \tag{3.2}$$

Afterwards, we compute an optimal mapping between individual cluster centroids from the two sets C_S, C_T. We formulate the optimization as an assignment problem, which we solve with the Hungarian algorithm [79]. We use l_1 as the distance function between two centroids. The output of the mapping consists in a set of pairs (k, j) where $k, j \in \{1, \ldots, K\}$ and the pair (k, j) indicates that points belonging to cluster k for speaker S should be transformed into points belonging to cluster j for the speaker T to maximize the similarity.

The above transformation is implemented using a linear shift in MFCC space. Given $m_i^{(a)} \in C_k^{(S)}$ and given the optimal mapping for cluster k, pair (k, j), we compute a transformed sample $o_i^{(a)}$ as follows:

$$o_i^{(a)} = m_i^{(a)} + t_j - s_k. \tag{3.3}$$

For an entire audio recording a, Equation 3.3 is applied sequentially to each sample $m_i^{(a)}$ in $\textsc{mfcc}(a)$, resulting in a set of transformed samples $\{o_0^{(a)}, \ldots, o_n^{(a)}\}$. Finally we invert the MFCC transformation using the method shown by Ellis [40], to give the transformed audio a^*:

$$a^* = \textsc{mfcc}^{-1}(\{o_0^{(a)}, \ldots, o_n^{(a)}\}) \tag{3.4}$$

3.3.2 Attack Execution

There are three steps to generate the attack audio, shown in Figure 3.3. In the first step, adversaries compute the phoneme clustering for a source voice, which can be their own. In the second step, they obtain a recording of the target's voice and compute clustering for this data. Immediately afterwards, the adversary can compute the optimal phoneme mappings between the source and the target clusters. In the final step the adversary selects a source utterance, usually the keyphrase or a voice command used by the system, applies the transformation in Equation 3.3 and creates a transformed utterance audio to be played to the system. The first step can always be computed *offline*, that is before the adversary selects a target, while the remaining steps depend on when the adversary is able to record the victim speaking and when they obtain physical access to the system.

Choice of Source Speaker. We found that the selection of source speaker greatly affects the quality of the transformation, meaning that certain voices can be more accurately mapped to certain targets. We therefore extend our attack to consider a population of individuals as sources, that the adversary can obtain by downloading online voice datasets, or recruiting a population of people to provide a set of potential source voices. This way, the adversary can compute mappings for each individual in the population, giving them several candidates to choose as the source utterance in the last phase of the attack (see Figure 3.3). As it is reasonable for adversaries to limit the number of *failed attempts* (i.e. playing an attack utterance and being rejected or wrongly classified by the system), one strategy is to estimate the chance of successful impersonation based on the mapping output.

Following these considerations, given a mapping composed of a set of pairs $(k_1, j_1), \ldots, (k_K, j_K)$, we use the sum of the L_1 norm of paired cluster centroids as an indicator:

$$\epsilon = \sum_{i=1}^{K} ||s_{k_i} - t_{j_i}||_1 \tag{3.5}$$

Intuitively, the lower the distance (error, ϵ) between the mapped clusters, the more accurate the transformation becomes. Therefore, whenever the adversary carries out an attack they sort the possible source voices based on increasing ϵ and use them as sources in this order.

3.4 Experimental Design

In this section we describe our data collection method, then present how we model the adversaries of Section 3.2 and describe the target systems considered for the evaluation.

3.4.1 Data Collection

Collection Procedure. We collected audio data from 20 male native English speakers, recruited mainly through social media and mailing lists. Participants were mostly from southern England and aged between 18 and 30. Recording sessions took place in an isolated room in a university building, taking approximately 30 minutes. Recordings were conducted using an AmazonBasics Portable USB Condenser Microphone, connected to a Windows laptop. Recordings used the inbuilt "Voice Recorder" software. Participants were instructed to keep the distance between themselves and the microphone between 5 and 15cm. The data collection was approved by the department ethical review process, with reference: SSD/CUREC1A_CS_C1A_18_032. Participants were informed of the purpose of the study and informed consent was obtained from them prior to commencing any recording sessions. As voice is personally identifying information, we do not publicly share the voice dataset.

Transcripts. The participants were required to utter sentences from four different categories: (i) conference transcripts, (ii) conversation transcripts, (iii) commands and (iv) enrolment transcripts. Each utterance source is designed to re-create the scenarios mentioned in Section 3.2. The enrolment and commands transcripts are identical for every participant, while for conference and conversation, to increase the dissimilarity of spoken words, we randomly assign one out of five transcripts to each user[2]. Transcripts were split into utterances of roughly equal length, with an utterance typically containing a single sentence.

3.4.2 Adversary Modelling

Conference Attacker. This attacker only obtains audio samples coming from utterances from the conference transcripts. In order to recreate the "conference" effect (the recorded audio coming from distant loudspeakers), we apply the following processing to the original audio. First we apply the Freeverb [136] algorithm to generate reverberation in the audio (following *data augmentation* practices used in Kaldi [117]). To

---

[2] Transcript summaries are available in appendix A

simulate recording from a distance, we apply a low-pass filter (with cutoff at 8Khz) to attenuate higher frequencies and scale the amplitude of the signal to reduce the volume.

Cafe Attacker. This attacker only obtains audio samples coming from utterances from the conversation transcripts. In order to recreate the "cafe" effect (recording from a distance plus background chatter and noise), we apply the same processing used for Conference Attacker (with less reverberation). Additionally, we mix the audio file with common cafe background noise[3] (the overlaid noise segment is chosen randomly per sample).

Ideal Attacker. This attacker uses the clean recorded audio from the data collection, with no post-processing or noise applied to it. The Ideal Attacker represents a worst-case scenario where the adversary obtains good quality audio samples, and we use it as an indication of the empirical upper bound for the attacker's success rate.

Audio Duration. In order to evaluate the effect of different amounts of audio on the attack success, we model two different audio durations in our experiments: *all* and *one minute*. The *all* case represents the case where we use all audio collected for a given scenario (either conference or cafe). The audio quantity averages 317.7 seconds for the Conference Attacker and 330.5 seconds for the Cafe Attacker, including any silence at the start or end of the speech recording. Ideal Attacker uses all the audio available for that victim, giving an average of 648.2 seconds per victim. In the *one minute* case, we randomly sample utterances from the related transcripts until we reach a cumulative total of 60 seconds of audio, including silence parts. We choose to systematically analyse each combination of these, creating six different scenarios (three attackers, two audio lengths).

3.4.3 Target Systems

We evaluate our experiments against speaker recognition systems, both in the identification and verification use-case. We use three different systems for the evaluation: (i) Spear [72] (ii) Azure Speaker Recognition APIs[4] and (iii) Apple iOS Siri ("Hey Siri"). The Spear toolbox is a set of libraries used to train and evaluate speaker recognition models, which we download and train locally with the VoxForge [158] dataset. Meanwhile, Azure Speaker Recognition only offers online (subscription-based) API access. Microsoft reported that the verification API has performance "competitive with the best published number" and that the identification API has "high precision

---

[3] https://youtube.com/watch?v=BOdLmxy06H0

[4] https://azure.microsoft.com/en-us/services/cognitive-services/

(above 90%) [which] is obtained at around a 5% rejection rate" [97][5]. Apple iOS Siri provides a real world test of the attack against a widely deployed system, which is used for accessing functions on iOS devices. Apple reports that the end-to-end performance of the system has an imposter acceptance rate of 3.2% [9] and an EER of 4.3% on the speaker recognition task alone (i.e not including keyphrase matching). In all cases, we treat the system as a black-box model: we never change nor adapt the method of Section 3.3.

3.5 Experimental Evaluation

In this section we first show some preliminary results on the Spear system, then show the results on the Azure Speaker API and finally on the Apple iPhone Siri.

3.5.1 Spear Toolkit

Setup. We use the Spear toolkit to train a GMM-based classifier, with 20 MFCC features plus their first and second derivatives as input features. Throughout our Spear experiments we use audio data obtained from the VoxForge [158] database. Specifically, we use data from users who define themselves as speaking "American English" and take the 63 users with the longest total amount of recorded audio. The users are then randomly split into three groups of 15 plus one of 18: (i) one group for training the background model, (ii) one for refining the model parameters (development set), (iii) one enrolled into the system (test set), and we use the larger (iv) fourth group as voice sources for the attack.

The classifier decides whether an input audio file belongs to certain enrolled user by computing a similarity score between the audio and the enrolled template for every user (identification), with larger scores being closer matches. We compute the EER on the development set by varying the score threshold for acceptance, finding it to be 7%, which corresponds to a decision boundary threshold of 1.38. We use the learned threshold on the (unseen) test set to compute the system recognition rates, which leads to a false accept rate of 3.7% and a false reject rate of 0%. Since we are using Spear as a baseline system to quickly evaluate the attack, we only consider the Ideal attacker in this section.

Results. Figure 3.4 shows two frequency distributions of distance scores from the acceptance decision boundary (vertical dashed line, set at the EER). The original distribution corresponds to distances obtained by testing an impersonation attack

---

[5] We conducted our experiments against the Microsoft APIs in January 2019.

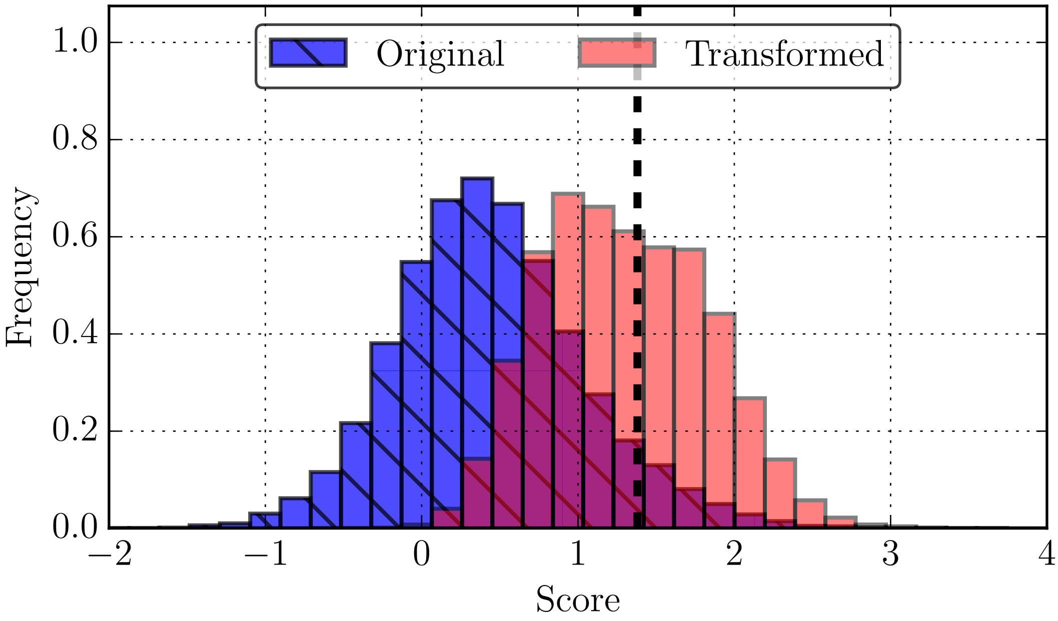


Figure 3.4: Frequency distribution of scores for identification before and after the phoneme transformation. Scores move towards the decision boundary after the application of transformation.

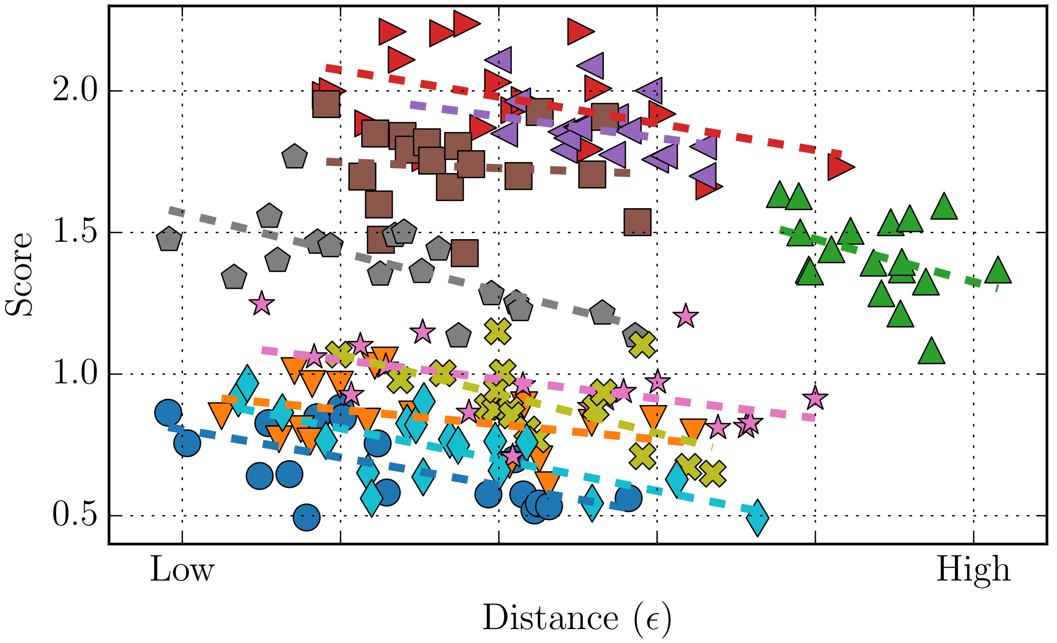


Figure 3.5: Similarity score between transformed audio and target user templates, computed by the classifier, as a function of distance between voices. Reduced distance leads to an increase in score (Spear). Each source voice uses a single colour, with each marking being a transformation to a different victim.

with non-modified voice samples (zero-effort attack), all possible source-target pairs (15×18) are used for the visualisation. The transformed distribution shows the distance scores for the same samples, but when applying the transformation of Section 3.3, no population is used in this case. Figure 3.4 shows that applying the transformation greatly increases the likelihood of the sample lying above the decision threshold and therefore being accepted.

Figure 3.5 shows how the mapping accuracy affects the success rate of the attack. The figure reports the distance from the decision boundary (score) of transformed samples, as a function of the error ϵ measuring the mapping (in)accuracy (see Section 3.3.2): lower error is correlated with higher matching score ($r = .48$). In Figure 3.5, each marker identifies all the data points related to a particular source voice (i.e. for source voice i, each $i \rightarrow j$ transformation with j being a target voice). For each source, we fit a linear regression curve to highlight this trend and we can see that as the distance (error) ϵ increases, the score of transformed samples decreases. Figure 3.5 also shows how some victim voices are more vulnerable to being impersonated than others, with clusters of higher scoring points belonging to some victims. In the next section we build on these results to evaluate the attack against the Azure APIs.

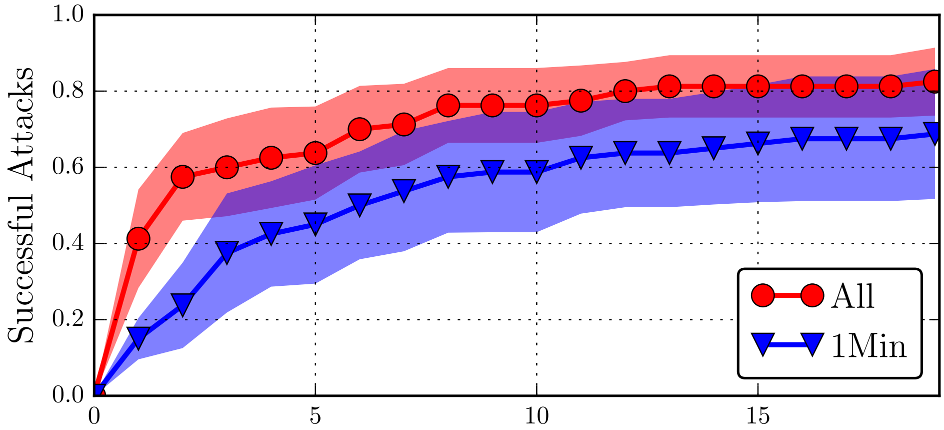
3.5.2 Azure Speaker Verification

Setup. The idea behind this experiment is to see whether the attack can be successfully conducted against a commercially available API, with a proprietary model for speaker verification. The Azure Speaker Verification API (hereafter ASV) is text-dependent and has a set of keyphrases that can be used with it. We collected audio of five of these keyphrases, which we require each participant to speak four times. Each user is enrolled using four samples of a given phrase (ASV requires at least three samples). There are no parameters within ASV to modify its performance and as such no way to adjust any thresholds associated with acceptance or rejection[6].

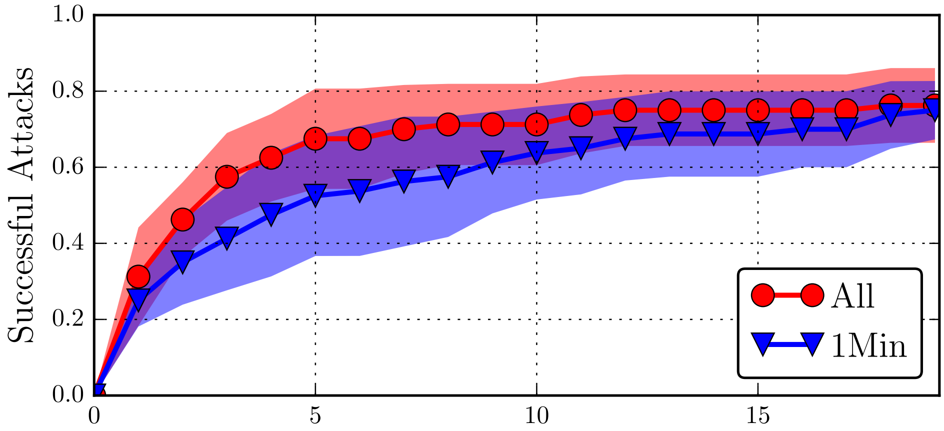
We generate attack samples for these keyphrases using each of our participants as a victim, using all the remaining participants as source voices, for each of our scenarios in turn. As we have four repetitions of each keyphrase, the attacker performs four authentication attempts for one source before moving to the next source. We submit each of these attack samples to ASV and receive a reject/accept response. Across all

---

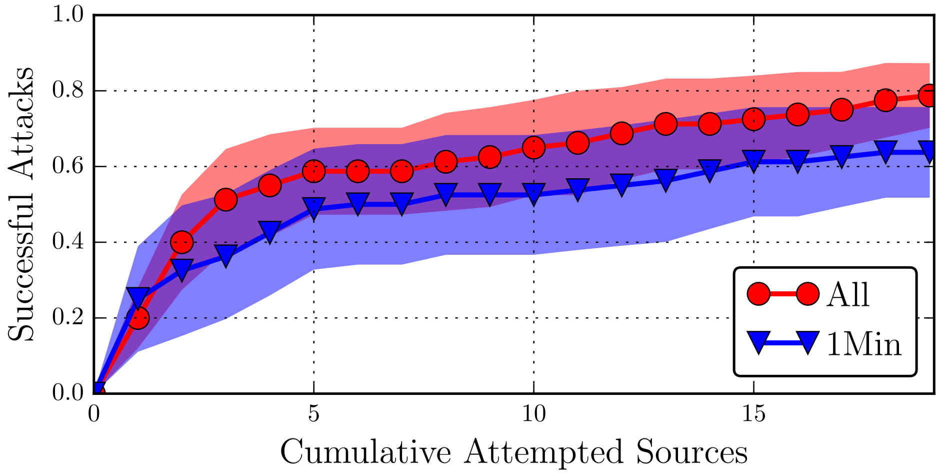
[6] We had to remove one phrase, "Houston we have had a problem", as participants spoke the phrase as "Houston we have a problem", a popular misconception.



(a) Ideal Attacker.



(b) Cafe Attacker.



(c) Conference Attacker.

Figure 3.6: Results of the different attackers on ASV, considering different amounts of audio. Shaded areas show results within one standard deviation, averaged over the four keyphrases.

scenarios we create and evaluate a total of 38,400 attack samples, which we use to evaluate the performance of our attack.

Results. Table 3.1 shows the results of verification experiments for each scenario and keyphrase. The values in Table 3.1 are the percentages of successful impersonation attacks, which are calculated in the following way: the adversary attempts impersonation with the first three sources in the ϵ-ranked list (see Section 3.3), if any of these are successful then we count this as a successful attack.

There is significant variability in the results between different keyphrases: KP_2 obtains the highest success rate on average (85%), while KP_3 performs the lowest (28%). This might be related to the mapping accuracy of the phonemes that form these utterances, which degrades when some phonemes are under-represented (i.e. they occur in low number) in the known victim audio. For example, the phonemes [dʒ], [ʊ] and [θ] all occur in KP_3 and are the 7th, 4th and 3rd least common phonemes respectively [17] and therefore likely to be under represented. We see differing success rates across scenarios and amount of known audio, with the one minute audio scenario performing consistently worse (-16%) than the all audio scenario. Ideal Attacker performs the best, but even the noisy audio of Conference and Cafe Attacker achieves high success rates.

Figure 3.6 shows the cumulative successful attacks as the adversary attempts impersonation with each source voice in his dataset (sources are ranked by ϵ). Unexpectedly, Cafe and Conference attacker do not seem to greatly suffer from the additional audio noise in comparison to Ideal. This suggests that even noisy recordings of the victim audio might carry sufficient information about his vocal tracts and further confirms that most of the distinctiveness of one's voice comes from lower frequencies, which best survive noise during the recording. The plots additionally show how one minute of audio is also sufficient (though with a slight decrease in success rate when compared to all audio) to re-create one's voice. The curve slope indicates that the ranking of possible sources brings a greater percentage of successes in the beginning, where promising sources are tested first. We can see that at around three attempted sources (corresponding to 12 authentication attempts), the adversary can get up to 60% success rate depending on the scenario. Even if there are only marginal increments in the successful attacks after testing 15 sources, using a larger population of sources would increase overall attack effectiveness, as this increases the likelihood of having promising source voices, which can be mapped accurately to the victim.

	Ideal		**Conference**		**Cafe**	
Keyphrase	**1 min**	**all**	**1 min**	**all**	**1 min**	**all**
KP_1:“my voice is stronger than passwords”	26.3%	52.6%	47.4%	57.9%	42.1%	57.9%
KP_2:“my password is not your business”	68.4%	94.7%	84.2%	89.5%	89.5%	89.5%
KP_3:“apple juice tastes funny after toothpaste”	21.1%	42.1%	15.8%	31.6%	21.1%	42.1%
KP_4:“you can activate security system now”	63.2%	73.7%	31.6%	52.6%	47.4%	73.7%

Table 3.1: Percentage of successful attacks using up to three source voices on ASV, computed for all scenarios and keyphrases.

3.5.3 Azure Speaker Identification

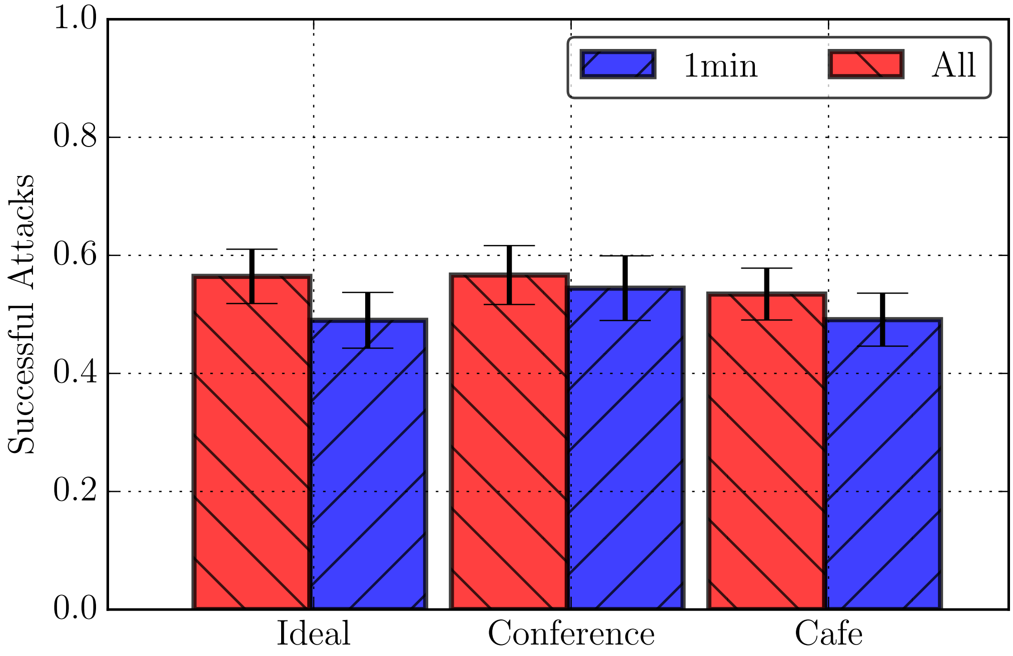
Setup. The Azure Speaker Identification API (hereafter ASI) is text-independent and requires a set of users to be enrolled, which are candidate users for who is speaking. In this case, enrolment requires a minimum of 30 seconds of audio per speaker, once silence is removed. To enrol users, we use audio specifically collected for this purpose, enrolling half of our participants in the system (see Appendix A for details). This gives us 10 potential victims and 10 attackers, for a total of 100 source-victim pairs.

ASI accepts an audio sample as input and replies with the inferred identity from the list of the 10 enrolled users, or an *empty* reply when an audio sample does not match any of them. It is not possible to adjust any threshold for ASI and as such there is no way of adjusting the threshold for when empty is returned. We send all command utterances that have been transformed between a particular source and victim to ASI, but we concatenate audio files together into groups of four to obtain audio samples of approximately 8 seconds. This is because ASI is designed for longer audio samples and without this concatenation the system returns none, as the samples are too short to make a decision. In total we submit 5,400 requests to ASI to conduct our experiments.

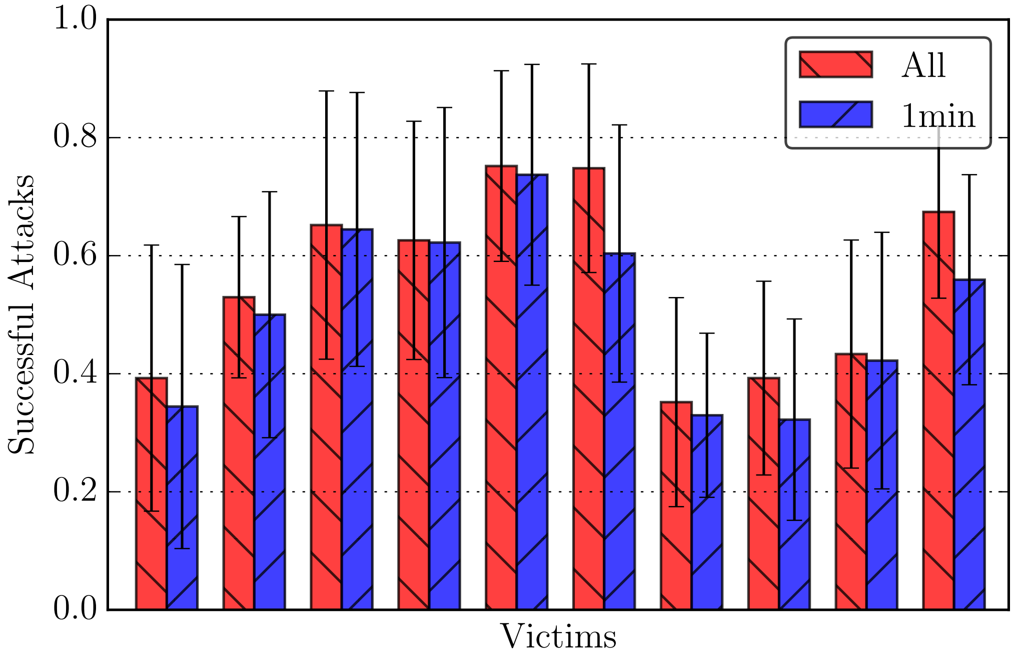
Results. Figure 3.7a shows the overall success rate for ASI for the three attackers and the two audio length combinations. In this use case each success corresponds to a submitted audio sample that is identified by the system as belonging to the victim. We see that the performance is broadly consistent across the scenarios, with a slight worsening of the recognition rates for the Cafe Attacker in particular (though not statistically significant, averaged over the 100 source-victim pairs) The performance slightly decreases in the one minute of audio case, but again with a minimal effect on the overall success.

Interestingly our results also reveal more information about ASI and its sensitivity. Table 3.2 shows that for all scenarios ASI was more likely to assign a speaker to an incorrect label than it was to return the empty user classification. This suggests that the decision boundaries across different users are not very conservative and that generally they can not deal well with outliers.

Figure 3.7b shows how the successful attacks distribute over different victims. The plot highlights that certain voices are more vulnerable to this type of attack than others: comparing the hardest to attack with the easiest to attack victim we get a difference of around 40% in the success rate. Similar uneven distributions of rates have been noticed before in previous work [6, 39]. This suggests that some voices



(a) Scenario analysis.



(b) Per-victim Analysis.

Figure 3.7: Average successful impersonations on Azure Speaker Identification API. Results show that changes in audio quantity and quality only have small effects on success rate. Plots show the successful attacks for each scenario.

	Ideal		**Conference**		**Cafe**	
	1 min	**all**	**1 min**	**all**	**1 min**	**all**
Misclassified	31.1%	27.6%	28.1%	26.4%	31.9%	29.2%
None (Empty)	19.9%	16.0%	17.4%	16.9%	19.0%	17.3%
Correct	49.0%	56.4%	54.5%	56.7%	49.1%	53.5%

Table 3.2: Percentage of incorrect and *empty* responses for the experiment on ASI, for each attacker and audio duration. ASI returns *empty* when the provided audio does not match any of the enrolled users. We report Misclassified whenever the returned identity does not match the target victim.

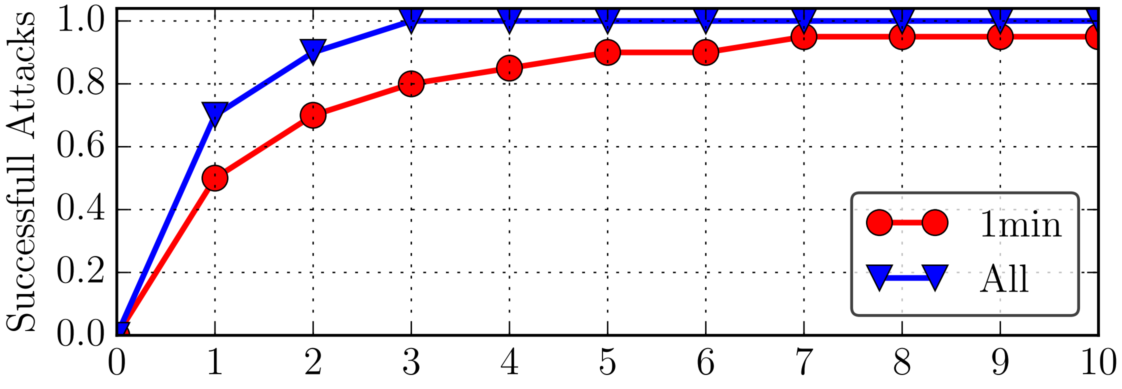
might be inherently harder to replicate, however, in our data, this might be due to a sample bias: some voices might significantly differ from our "average voice". A larger dataset would be required to investigate further whether this is the case.

3.5.4 Apple iPhone's Siri

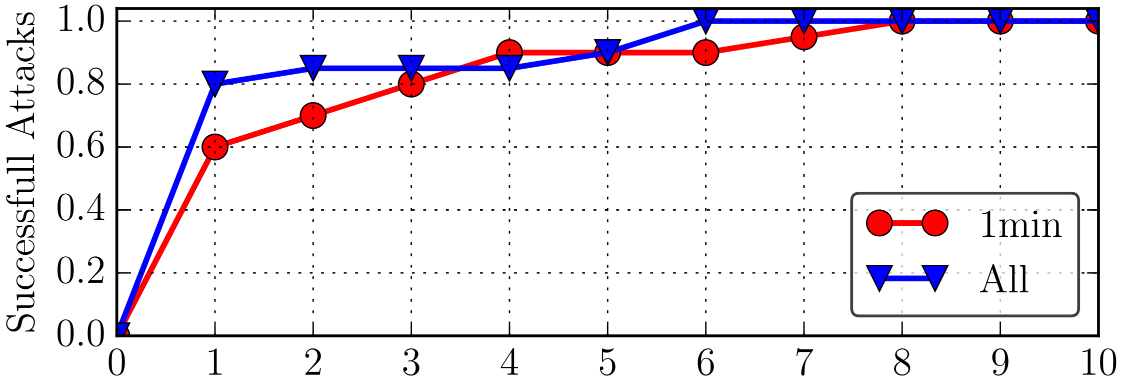
Setup. In order to measure the capability of the attack of being conducted over-the-air we test the samples against the voice activation functionality of the Apple's Siri digital assistant on an iPhone 6S, running iOS version 12.2. We use the collected voice recordings of each of our 20 participants to enrol them onto the device. For both enrolment and attacks, we use a Bose SoundLink Mini 2 speaker to replay the participants audio samples. The speaker is placed 6 centimetres away from the smartphone in an office environment. Initial enrolment requires the user to pronounce four different phrases, which we construct by combining the original recordings of the collected "Hey Siri" utterance with the remaining words of the enrolment utterance added by splicing together audio from other recordings of the same individual. Siri speaker recognition updates the user template after a successful access [9]. Therefore, after a successful attack we erase the user profile and repeat the enrolment process.

We test the system along the same dimensions as our previous experiments. When conducting the attack, we play a single transformed utterance of the keyphrase ("Hey Siri"), from each source voice, in the order suggested by our error function (nearest to furthest). If Siri activates, i.e. the voice is recognised as belonging to the legitimate user, we consider the attack successful and we do not present further samples. At the time of completing these experiments, Apple claimed Siri had an impostor accept rate of 3.2% [9].

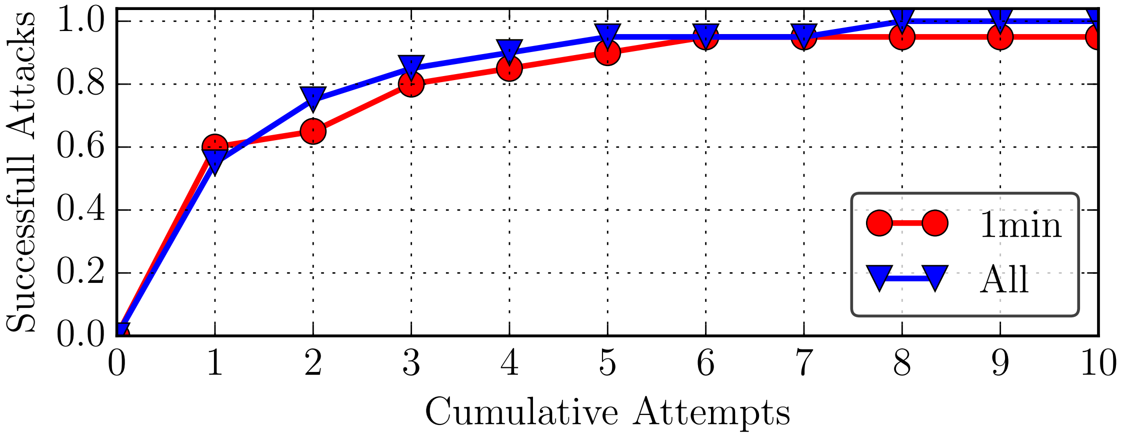
Results. Figure 3.8 shows the percentage of victims successfully impersonated after a given number of attempts, for each attacker and the two known audio amounts.



(a) Ideal Attacker.



(b) Cafe Attacker.



(c) Conference Attacker.

Figure 3.8: Results for different attackers on Siri. Plots show ratio of successful impersonations as the adversary consecutively attempts the attack with different source voices.

The results show that performance is consistent with previous experiments, in that the differing scenarios lead to slightly worse success rates and that performance is also worse in the one minute audio case. Our results demonstrate that the Siri voice activation is easily fooled by our attack. For all scenario and amount of known audio combinations over 70% of victims can be attacked in three attempts or fewer. Excluding one individual in two of the one minute scenario-time combinations, who could not be impersonated, all other attacks were successfully conducted in 8 attempts or fewer. In our dataset an utterance of "Hey Siri" took approximately 2 seconds, meaning that in most cases 20 seconds would suffice to successfully carry the attack out.

3.6 Discussion

3.6.1 Implications

The attack presented here demonstrates that a minimal amount of voice from a victim can be sufficient for an adversary to impersonate that victim with a high success rate. The attack's only requirement is to obtain a recording of the victim talking. Sources such as social media, podcasts and recordings of public speaking events are all easily available sources of such audio. Consequently, the audio becomes even easier to gather for higher profile targets. The ease of collection of voice samples in adversarial scenarios brings an inherent security vulnerability of voice-based systems, as highlighted by our analysis. We point out this vulnerability in order to raise awareness of the limitations of such authentication mechanisms, so that they can be accounted for during the design of voice-based systems.

3.6.2 Replay Detection

Similarly to other voice-based attacks, our method involves replaying audio to the system microphone via a speaker. This is necessary for all attacks on voice systems that use only over-the-air interaction and do not require harder to obtain over-the-wire access. A set of works have addressed the detection of replay attacks on such systems [14, 16, 27, 41, 42, 176]. Some detection techniques rely on a combination of better hardware (e.g. multiple microphones) or require additional interactions from the user. Often replay detection evolves into an arms race with the adversaries improving their audio sample to present the features required to bypass detection.

3.6.3 Rate-Limiting

Oftentimes in verification systems, the number of failed authentication attempts can be used to temporarily block the authentication or swap it with more secure alternatives. For example, in Apple FaceID the face recognition is disabled after five failed authentication attempts, at which point a PIN is required to unlock the phone. We find that even if the 5-attempts limit were the same for Siri, a high percentage of victims would still be attack-able (90% in the 1 minute ideal scenario). Keeping the number of sequentially allowed failures low before locking the system becomes an immediate an effective way to prevent our and other population-based attacks.

3.6.4 Only British English

Our dataset only includes British English native speakers, specifically from England. It does not include other dialects such as the ones from Scotland and Wales, although does include several differing English accents. We found that the transformation degrades slightly when it is attempted across different dialects (or from British to American English). This means that the adversary would need to collect a population of voices with the same language and dialect to maximise the chances of success. While this might be straightforward for English (and in particular American English), it might not be as easily obtainable in other languages. In theory the attack should work against any language, provided the number of phonemes used for clustering is adjusted. Additionally our dataset is comprised of only male participants, however we expect the attack to achieve similar performance against female voices (given the availability of a population of female voices).

3.7 Conclusions

In this chapter, we propose our attack method to transform a source voice into a victim's voice to deceive speaker recognition systems. The transformation maps individual phonemes between the source and target voices and only requires knowledge of the number of language phonemes, a set of source voices (easily available online) and an audio sample of the victim speaking. Furthermore, we identify a metric for determining which voice among a group of voices is most likely to lead to a successful authentication.

We evaluate the attack under a set of scenarios that include different amounts and quality of victim audio and different systems. We test our attack on a GMM based system, as well as both the Azure Speaker Recognition APIs and the Apple iOS Siri voice assistant. On Azure, for verification, we show that 12 authentication attempts are sufficient to successfully impersonate victims in 40% up to 68% of cases, using just one minute of victim audio for training, even in noisy recordings conditions. For identification, the method achieves much higher success rates reaching over 50% on average with a single attempt. We demonstrate that high success rates can be obtained even when testing the attack over-the-air on Siri: 80% of victims can be impersonated within three attempts, which correspond to only 8 seconds of audio in total.

Compared to previous work, these findings reveal that limited quantity and quality of audio have only limited impacts on the overall success of this attack. Given the increasing availability of potential victim's audio, our analysis highlights the vulnerability of using voice as a biometric for access control in adversarial settings, suggesting that such weakness should be included in the design phase of such systems. In light of this threat, in the following chapters we turn our attention to voice privacy, to look for solutions that allow system providers and system users to protect their voice trait from being capture. This in turn may help reduce the audio available for an attacker to use in impersonation them.

Chapter 4

Voice Privacy when Preserving Speech Traits

Contents

In Chapter 3 we demonstrated an attack that allowed an adversary to impersonate a victim, with limited amounts of audio and of reduced quality, in doing so revealing the vulnerability of speaker recognition systems. In large part this vulnerability stems from the voice being *broadcast* in everyday settings, unlike other biometrics which are typically harder to obtain. Similarly the hardware required to collect voice data is simply a microphone, whereas many other biometrics, such as fingerprint, or electrocardiogram based metrics, require more advanced collection hardware.

As such we turn our attention to how voice data can be protected in this chapter. We do so by developing a system for implementing *voice privacy*, whilst preserving the traits of the original speech. We do this as part of the Voice Privacy Challenge, which is discussed further in Section 4.2.1.

4.1 Introduction

As the use of voice as an interface proliferates, combined with people generating more vocal content which is often widely shared online, it is becoming increasingly clear that solutions for anonymisation of individual voices are necessary. Voice cloning techniques have been rapidly advancing, with systems now able to generate realistic synthetic voices [87, 129], as is further explored later in Chapter 5. At the same time further works demonstrated that few voice samples are required to bypass voice authentication systems and clone users voices such as [10] and the attack demonstrated previously in Chapter 3.

Voice data, whether captured live or leaked from remote servers, not only constitutes personally identifiable information but can also contain user-sensitive information. Together with the introduction of new regulations, such as the General Data Protection Regulation in Europe, it has become increasingly important to develop techniques and methods to protect the privacy of voice data from adversaries.

Speaker anonymization techniques have been proposed to fulfil these protection requirements. These techniques process audio, so that the user-identifiable components of speech (i.e. those that link speech to user identity) have been removed, whilst retaining speech content and its other characteristics, such as tone and delivery.

The VoicePrivacy Challenge 2020 [148] (VPC) was one of the early efforts to provide a common ground for the speaker anonymization problem. The challenge established datasets and metrics to evaluate speaker anonymization methods. In this work we develop a system within the parameters of the VPC, by focusing on improving the anonymous identity generation mechanism of the x-vector baseline challenge system. The baseline system works on the principle of decomposing the audio into the identity component, x-vectors [137], and non-identifying components. The x-vectors can then be replaced with a pseudo x-vector, generated according to some strategy, with the audio then re-synthesised from these components. This chapter stems from our entry into the Voice Privacy Challenge 2020, as well as the subsequent journal paper that extended this work.

In this chapter we highlight how the VPC baseline anonymous identity generation methods leads to pseudo x-vectors which are largely similar to each other. These pseudo x-vectors represent the identity that the new voice will belong to once synthesised and thus similarity in these leads to less unique final voices.

We find this to be a consequence of averaging a large set of original x-vectors in the baseline technique. To overcome this problem, we instead train Gaussian Mixture Models (GMMs) on a reduced version of the x-vector space; we then can sample from these GMMs to generate new pseudo x-vectors.

We provide a general pipeline to help choose parameters to optimise the GMM performance. For this we make use of two relevant metrics which can be applied in x-vector space to estimate how well the GMM approximates the original x-vector distribution and its properties.

The anonymized voices are generated by re-synthesising components derived from the original audio with our newly generated pseudo x-vectors.

In this chapter we additionally show that the assumption of perfect separation between identity and speech content does not hold in the underlying pseudo-xvector anonymisation system. In fact, we find that anonymous voices are biased towards the original voice, highlighting a potential avenue for improvement to be addressed in future work.

4.2 System Setting

4.2.1 The VoicePrivacy Challenge 2020

The VoicePrivacy Challenge 2020 (VPC) [148] provides the setting for this chapter and defines specific goals, a selection of datasets, and a set of metrics for the evaluation and comparison of voice anonymization systems. The challenge seeks solutions for a scenario where speaker identity is hidden whilst still allowing all other downstream goals (e.g. speech recognition) to be achieved [147]. This is done by converting a speaker to a *pseudo-speaker* with a different voice: the new anonymous identity of the original speaker.

In order to accomplish downstream goals the following system requirements are given for the system: (a) to output a speech waveform, (b) to hide speaker identity as much as possible, (c) to distort other speech characteristics as little as possible, (d) to ensure that all trial utterances from a given speaker appear to be uttered by the same pseudo-speaker, while trial utterances from different speakers appear to be uttered by different pseudo-speakers.

The VPC specifies the evaluation dataset subsets for the evaluation of developed models: (i) the Librispeech [112] clean development and test sets, and (ii) the VCTK [168] development and test sets. We evaluate our work using the objective metrics proposed by the VPC, using the models for this trained by the VPC, namely Equal Error Rate (EER), log-likelihood-ratio cost function $\mathrm{C_{llr}}$ and the discrimination loss component of this, $\mathrm{C_{llr}^{min}}$, when analyzing the privacy performance of the system, and Word Error Rate (WER) for evaluating the speech recognition performance.

Several additional metrics were proposed as part of the VPC for evaluating speaker anonymization systems, the results of which we also evaluate in this work.

Specifically we also consider:

- Expected privacy disclosure at a population level, $\mathrm{D_{ECE}}$, and worst case privacy disclose for an individual, $\log_{10}(\mathrm{l})$, from the Zero Evidence Biometrics Recognition Assessment (ZEBRA) framework [108].
- Linkability, specifically the global linkability $D_{\leftrightarrow}^{\mathrm{sys}}$ [91], which examines the (differences in) mated and non-mated score distributions.
- Global de-identification De_{ID}, and global voice distinctiveness G_{VD}, both of which are derived from voice similarity matrices [109].

The VPC also stipulated which datasets may be used in training anonymization systems in order to ensure that results are comparable. As such we make use of these same datasets for training our system: VoxCeleb 1 and 2 [30,106], LibriSpeech train-clean-100 and train-other-500 and LibriTTS train-clean-100 and train-other-500 [173].

4.2.2 Usage Scenario

We outline two example usage scenarios, which motivate the use of a voice anonymisation system such as the one developed in this chapter. These scenarios both fit within the aims set out in the VPC.

Privacy Conscious Individual An individual wishes to share a recording of them speaking about a topic they are passionate about online. However, they do not wish for the speaker in the audio to be identifiable as they prefer to operate anonymously when discussing this topic. As such they wish to apply an anonymisation method to the audio that removes the identifying information, whilst retaining the prosody of their voice, so that listeners can hear the emotion involved. An example of a person who could have such desires would be a political dissident.

Risk Averse Organisation An organisation records the audio from their call centre, where sensitive information is often discussed. They are worried about the liability to them of holding this data, especially in the event that the data was lost or leaked somehow. However, they do not wish to delete all of the audio, as it has utility to the business after the call is finished. They use recorded calls for training new employees, as well as for quality control processes. To alleviate their concerns they wish to anonymise the audio that they store, so that they can continue to use it for training and review, but without the risk of compromising the privacy of the users of their service and the associated reputational risks of such an incident.

4.2.3 Threat Model

Throughout this chapter we consider the same external adversary, who wishes to infer the identity of the speaker of some anonymised audio. Our threat model is informed by that of the attacker model in the VPC.

Capabilities

The adversary has the ability to compare utterances with one another, evaluating their similarity to determine if they came from the same speaker. They have access to state of the art speaker identification systems for doing this. The adversary also only attempts to de-anonymise the audio using techniques that user speaker identification systems that operate on the audio. They do not try to re-identify the speaker by analysing the prosody, or word choices in the speech.

The adversary does not have the ability to use the anonymisation system – that is to the say that we do not consider an adaptive attacker. Whilst this introduces an unrealistic assumption for the real world, it is included as it is difficult to develop methods that simulate an adversary that has this ability. Furthermore, as the system was developed as part of the VPC, it is hard to develop an adaptive attacker than can be applied to compare differing systems.

Knowledge

The adversary has access to one or more anonymised utterances that they wish to de-anonymise. They have access to enrollment audio for the potential speakers of the audio. We investigate the scenario where this enrollment audio belongs to the original speaker, as well as when this enrollment audio is also anonymised (but to a different new identity). These correspond to the O-A and A-A settings used in our experiments.

4.3 System Design

4.3.1 Overview

Our system design follows the same approach as the x-vector baseline system used in the VPC [147] and is inspired by [43]. This system takes the audio to be anonymized and derives three components from it, the x-vector, the bottleneck features (BN), and the pitch information (F_0). The BN features are obtained by applying an Automatic Speech Recognition (ASR) acoustic model, which is a factorised time delay neural network (TDNN-F), with 40 MFCCs and a 100 dimension i-vector as input, producing output BN features of dimension 256. This ASR model is trained using Librispeech train-clean-100 and train-other-500.

The x-vector extractor is a TDNN, using 30 MFCCs as input features, outputting a 512 dimension speaker x-vector. It is trained using Voxceleb 1 and 2. The system assumes that these components decouple the speech content (BN and F_0) and the speaker identity x-vector.

Following this the x-vector is modified or replaced according to a generation technique; the modified x-vector is termed a pseudo or fake x-vector. The pseudo x-vector represents the new identity of the speaker and is used for all utterances intended to be spoken by that identity. Subsequently a speech synthesis module uses the F_0, BN features and pseudo-x-vector to generate mel spectrograms. This speech synthesis model is an autoregressive network, outputs Mel-filterbanks of dimension 80 and is trained using LibriTTS train-clean-100.

A Neural Source-Filter (NSF) model then processes these filterbanks, along with the F_0 and the pseudo x-vector, generating the anonymized audio. This NSF model is trained with LibriTTS train-clean-100.

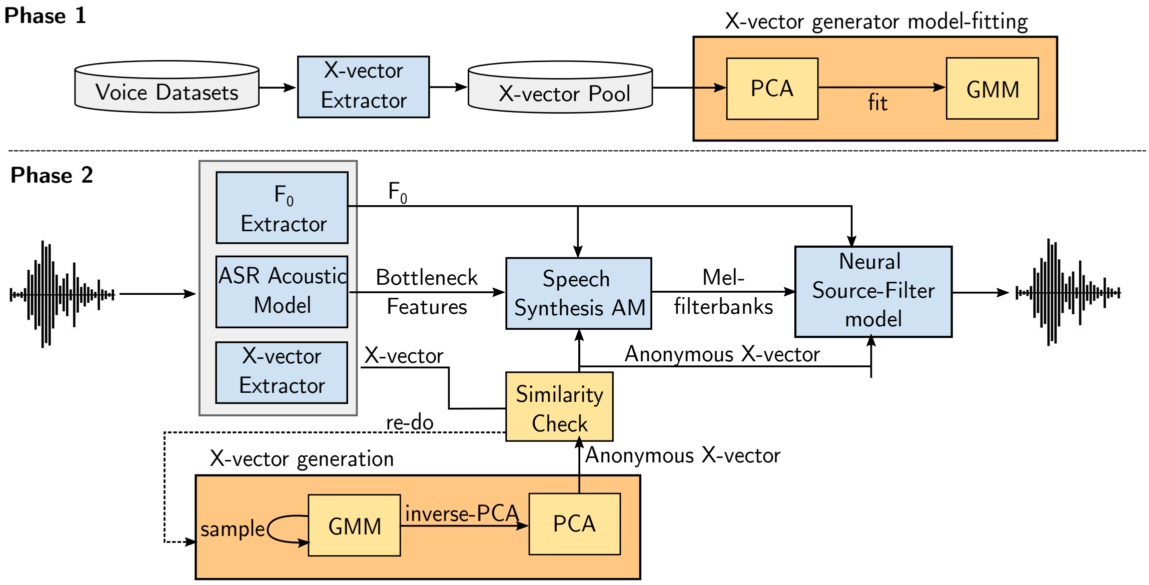


Figure 4.1: Voice Anonymisation system diagram. We replace the sub system for generating pseudo X-vector's with a a combination of PCA and GMM (shown in orange in the diagram).

A diagram showing the system can be seen in Figure 4.1. We use the models provided during the VPC, see [43, 147] for further details on the training of these models.

[43] proposed three techniques for generating fake X-vectors: (i) nearest speakers, (ii) random selection and (iii) range selection. The VPC x-vector baseline system uses a variant of the last of these techniques, selecting the 200 furthest away x-vectors from the original speaker and averaging a random selection of 100 within these to produce the new fake x-vector. The LibriTTS [173] train-other-500 dataset is used in the baseline for this pool, with 600 users in the male pool and 560 in the female pool.

4.3.2 Rationale

We examine the pseudo x-vectors that are created by the baseline generation technique. Figure 4.2, shows the cosine similarities of the x-vectors extracted from the original voices and pseudo x-vectors supplied to the later stages of the baseline system. We see that that the cross-similarity distribution between original voices and the pseudo x-vectors differs: pairs of pseudo x-vectors are more similar to one another than pairs of original voices.

This reduction in entropy leads to anonymized voices which are less distinct from one another, increasing the difficulty of distinguishing between anonymized voices.

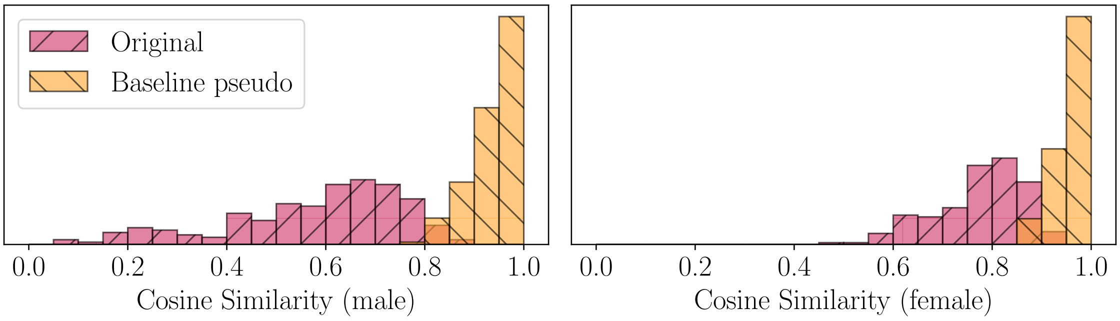


Figure 4.2: Distribution of cross-cosine similarities between pairs of x-vectors from original voices and from the baseline fakes. The baseline fake x-vectors do not follow the same distribution of cosine similarities as the original x-vectors: these fake x-vectors are much more similar to one another than x-vectors extracted from organic speakers.

Furthermore, this also means the x-vector space is being underutilised, meaning the total number of distinct anonymous voice available will be reduced compared to organic voices.

We postulate that the reduction in entropy and consequent reduced diversity of voices, occurs due to the averaging of several x-vectors in the pseudo x-vector generation process. Intuitively, similarly to what happens when sampling the mean of random samples of a normal distribution (which leads to a reduction in variance of a factor n, with n sample size), averaging subsets of 100 x-vectors will reduce the variance of the sampled x-vector means. In other words, the set of subset means will be more central in the complete x-vector space than the original population. While this can not be modelled exactly since the subsets are not sampled at random (but are biased depending on the current user), its effects are clear. Likewise, the alternative selection methods proposed by [43] and [138] suffer from the same problem, due to their use of averaging x-vectors.

Figure 4.2 also reveals that the distribution of cosine similarities for females differs significantly from that of males, with female voices having increased similarity compared to males. This performance may be due to an imbalance in the number of voices for males and females in the dataset used to train the xvector extractor (2912 females vs 4451 males). Alternatively it could be due to female voices being higher pitch and therefore containing less spectral information, resulting in increased difficulty in discriminating between them than male voices.

4.3.3 Method

We improve the x-vector generation in two steps. At first, we learn the properties of the 512-dimensional x-vector space by using principal component analysis (PCA) on a large x-vector dataset. Secondly, we fit a generative model on the PCA-reduced space, in order to sample from it. By using a generative model we can retain as much of the diversity of the original space as possible and avoid removing entropy with averaging operations. To generate a new pseudo x-vector using our method, we sample from the GMM in the PCA reduced space and then apply the PCA inverse transform.

As in the baseline, in the later stages of the anonymization a Speech Synthesis acoustic model is used to generate Mel-filterbank features, which are fed with the F0 and pseudo x-vector to a Neural source-filter model to generate audio. We train and reuse the models in the same way as the baseline, with the exception that we use the VoxCeleb1 [106] and VoxCeleb2 [30] datasets in our pool of speaker x-vectors, in addition to the LibriTTS [173] train-other-500 dataset. Figure 4.1 gives a full overview of how the system components fit together.

4.3.4 Determining Hyper-Parameters

4.3.4.1 Preliminaries

In order to choose the best parameters for the system, we focus on analyzing the performance of two key metrics: (i) the cross-similarity distribution between the fake and original x-vectors and the (ii) resulting differential entropy of the trained GMM. High match between the cross-similarity distributions of original and anonymous x-vectors indicates that the generated anonymous x-vectors retain the similarity properties that are expected in the original set. Differently, the differential entropy of the Gaussian mixture model is an indicator of how much information is retained by the mixture (akin to its discrete counterpart being a measure of how many bits are necessary to encode the information). In this case, GMM with high entropy are preferable, as they more closely approximate the underlying mixture distribution.

We measure the cross-similarity distribution with the Kolmogorov–Smirnov (KS) statistic:

$$D_{KS} = \sup_x |F(x) - O(x)|, \tag{4.1}$$

where F and O are the cumulative distributions of the empirical cross-similarity distributions among a set of X-vectors:

$$f(x) = \text{cossim}(x_i, x_j), \forall i, j \in x \tag{4.2}$$

O refers to the original X-vectors and F refers to fake (pseudo) X-vectors. We use the KS statistic for this as it is non parametric and can be easily applied to two empirical distributions.

The differential entropy of the GMM does not have a closed-form so we instead measure it by repeated Monte-Carlo sampling of the log-likelihood of the GMM-generated X-vectors:

$$\hat{H}(X) = \mathbb{E}_i[\log \sum_{k=1}^{K} \pi_k \frac{\exp(-\frac{1}{2}(x_i - \mu)^T \Sigma^{-1} (x_i - \mu))}{\sqrt{(2\pi)^k |\Sigma|}}], \tag{4.3}$$

with x_i the generated samples, μ the component means, Σ the covariance matrix and π_k the weight of the k-th component. As this way of estimating the differential entropy (hereafter entropy) may be imprecise for insufficiently large samples, we also report the mixture entropy estimators introduced by [77], which provide a tight estimation and are extremely fast to compute. See B for a more in-depth description of these estimators.

4.3.4.2 Setup

In the analysis, we vary the number of PCA and GMM components. We setup the evaluation as follows. Firstly we extract all the development and test X-vectors from VoxCeleb1, VoxCeleb2 (4,451 and 2,912 for male and female), and for each gender we perform a 50% train-test split, training the PCA+GMM models with only the training split. 2,000 samples are taken from the GMM and used to compute the entropy of Equation 4.3. We then apply the PCA inverse transform to obtain 512-dimensional pseudo x-vectors. To compute the KS statistic of Equation 4.1, we compute the cross-similarity among these pseudo x-vectors (obtaining F, Eq. 4.1) and we do the same among the testing part of the initial 50% split (obtaining O, Eq. 4.1). For the GMM we learn a diagonal covariance matrix on the PCA features, as these are de-correlated from one another. We set the maximum number of Expectation-Maximization iterations to 500 and the convergence tolerance to 10^{-15}. We increase these values linearly if the EM does not converge. Additionally it is easier to compute the entropy using a diagonal co-variance matrix, and the reduced dimensionality of the matrix makes parameter estimation easier.

As the GMM fitting may vary, we repeat the train-test split two times, and for each split we also repeat the training twice and average the results.

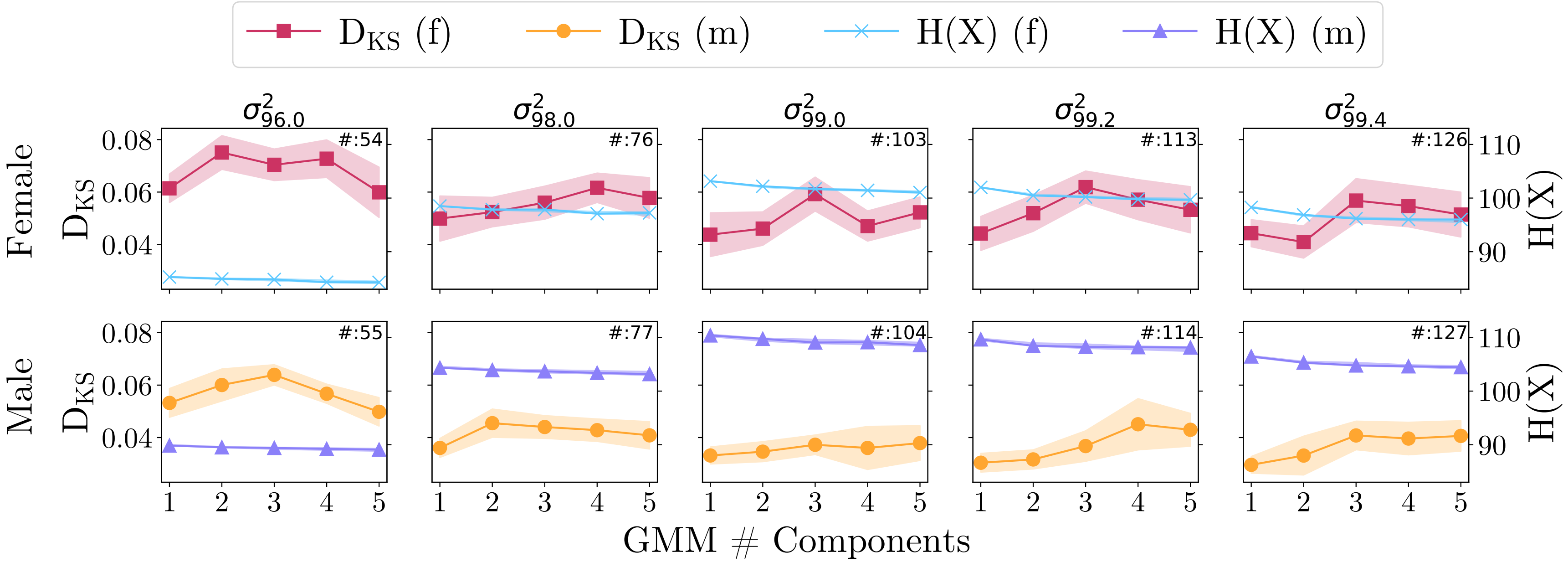


Figure 4.3: KS statistic (D_{KS}, Eq. 4.1) and entropy (H(X), Eq. 4.3) results across number of GMM components, amount of retained PCA variance (σ_p^2). Number of PCA components are given in the top right of each plot. Shaded areas for D_{KS} are 90% confidence intervals for D_{KS}. Shaded areas for H(X) are the lower and upper bounds measured with the estimators of Section B. The figure shows that increasing number of GMM components has a negligible effect on the two metrics.

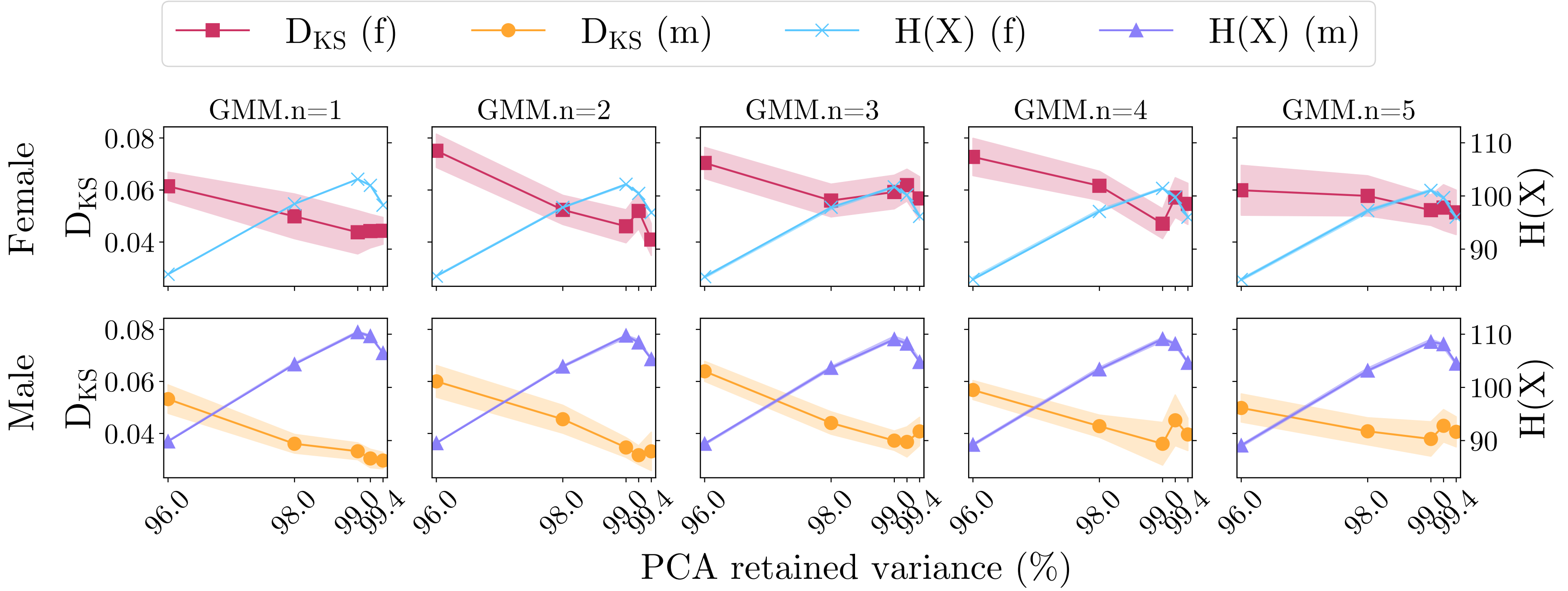


Figure 4.4: KS statistic (D_{KS}, Eq. 4.1) and entropy (H(X), Eq. 4.3) results across the amount of retained PCA variance (σ_p^2) and number of GMM components. Shaded areas for D_{KS} are 90% confidence intervals for D_{KS}. Shaded areas for H(X) are the lower and upper bounds measured with the estimators of B. The figure shows that the best entropy-distribution similarity trade-off is found at 99% retained variance.

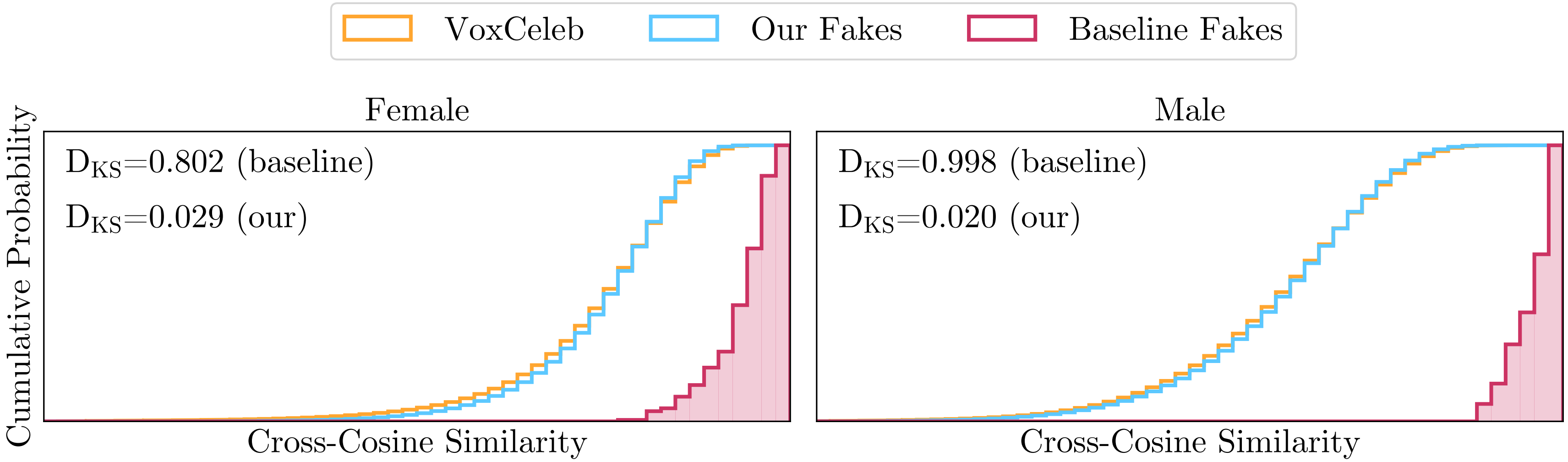


Figure 4.5: Example empirical cumulative distribution functions computed during the hyper-parameter search. The plots use one GMM component and 99% retained PCA variance. We do a 50% split on VoxCeleb data and use the first part to train our PCA+GMM and the second part to plot the VoxCeleb data series. We plot the baseline fake X-vectors computed with the VPC technique.

4.3.4.3 Results

We report in Figure 4.3 and 4.4 the resulting entropy and KS statistic for varying the number of GMM components and for varying PCA retained variance. Figure 4.3 shows that the number of GMM components does not significantly affect the result (either the similarity or the entropy), showing some fluctuations but no statistically significant improvement for increasing number of components. Instead, the amount of retained variance more directly affects the result: increasing it to 99% bring significant improvement in the similarity and also corresponds to the configuration which gives the maximal entropy. Figure 4.4 confirms the same insights, also highlighting how retaining more than 99% of variance leads to a slight degradation in entropy and no visible benefit in the distribution similarity. This can be explained with the fact that increasing the retained variance has diminishing beneficial returns, as the number of PCA extracted features has to grow significantly as we retain more variance. For example, for male x-vectors, going from 96% to 98% increases the number of features by 22, while going from 99.2% to 99.4% (a ten-time smaller increment) increases the same number by 13. Figure 4.4 shows that retaining more of these features is only beneficial up to 99%, after which point the co-variance matrix needs to include increasingly small elements which bring a reduction to the overall entropy.

For the remaining experiments we choose to use one GMM component and 99% of variance retained by the PCA transformation, which corresponds to 103 components for females and 104 for males. Using these parameters, we report in Figure 4.5 a comparison of the cross-cosine similarity distributions across various x-vectors subsets, pseudo and original. The figure shows that our generated pseudo x-vectors more closely match the expected similarity distribution found among VoxCeleb x-vectors compared to the baseline-generated anonymous x-vectors.

4.3.5 Forced Dissimilarity

As the GMM pseudo x-vectors are extracted randomly, these might occasionally be relatively close to the user's x-vector (this does not happen in the baseline as pseudo x-vectors are generated by selecting the n-furthest away x-vectors). This is detrimental to the quality of anonymization, so to avoid it we introduce a similarity check, termed *forced dissimilarity*, between the speaker's x-vector and the generated pseudo x-vector. For a pseudo x-vector X_p, we repeat the generation as long as X_p and the original x-vector X_o are too similar based on the θ_{FD} parameter:

$$\cos \operatorname{sim}(X_o, X_p) > \theta_{FD}.$$

We study the effects of this mitigation on the anonymization process in Section 4.4.3.

4.4 System Evaluation

In this section we evaluate the performance of our anonymization system, firstly by applying the VPC framework and extended metrics. Subsequently we also examine performance when using an alternate ASV system and under our forced dissimilarity measure. Finally we examine the resulting x-vectors from our produced audio, to glean further insight into the systems performance.

4.4.1 Evaluation with Voice Privacy Challenge Framework

4.4.1.1 Setup

The VPC evaluation framework uses two datasets for evaluation, with two sub splits of each: (i) the Librispeech clean development and test sets, and (ii) the VCTK development and test sets. We focus on the diff split of the VCTK dataset in our analysis, as performance on both sets is very similar and leads to lots of duplication of results.

The VPC evaluates speaker verifiability, using metrics derived from speaker verification scores, as well as word error rate. Two scenarios are studied for speaker verifiability, which correspond to the two different types of enrollment data our adversary may encounter under our threat model. Firstly original enrolment and anonymized trial (O-A), which examines the differences between the original voices and an anonymized version of them. In this case scores are computed between the clean audio and the anonymized audio of the same speaker for a target trial, with the anonymized audio of a different speaker used for a non-target trial.

Secondly anonymized enrolment and anonymized trial (A-A) is examined, where each of the enrolment and trial utterances are anonymized but to different identities. In this case a target trial is the anonymized enrolment and trial of the same original speaker, but anonymized to two different identities. Non-target trials are anonymized enrolments and anonymized trials from different speaker. In both cases the same identity is used for all utterances within that set (i.e. all the enrolment utterances are anonymized to voice A, all the trial utterances to voice B, where $A \neq B$. Speaker verifiability scores are calculated using an x-vector based system [137] trained using the LibriSpeech train-clean-360 dataset, using a probabilistic linear discriminant analysis (PLDA) backend.

We focus on EER and C_{llr}^{min}in our analysis, as these were utilised in the VPC and have been demonstrated to be robust in [91]. These are both computed from the set of scores between target and non-target utterances. Both the EER and C_{llr}^{min}minimise the discriminating power of the classifier against a dataset. Due to the anonymization applied to the audio, an EER of 50% and a $C_{llr}^{min}$of 1.00 are optimal, as target trial utterances are always anonymized compared to the enrolment utterance, and thus we hope to see that classifier unable to identify the two utterances as being spoken by the same (original) speaker.

Word error rates are calculated using a TDNN-F acoustic model with a trigram language model, based on the Kaldi recipe for Librispeech. Both evaluation systems are trained with the LibriSpeech train-clean-360 dataset. Further details can be found in the VPC evaluation plan [148].

4.4.1.2 Initial Framework

The full results for EER and $C_{llr}^{min}$are presented in Table 4.1, using the parameters determined previously in Section 4.3.4.3 (99% variance retained, one GMM component, one model per gender). The number of target and non-target trials for each of the datasets are given in Table 4.2.

For the O-A scenario, we experience a small performance degradation when compared to the baseline in most cases, with EER decreasing by up to 6.69% (female LibriSpeech test). The values of C_{llr}^{min}also feature a small drop, of a maximum of 0.04, however the values still remain generally close to 1, and as such the performance decrease is fairly limited. The degradation in results is more pronounced in females, with a C_{llr}^{min}decrease averaging 0.03, whereas for males the average decrease is 0.01. This could be because the baseline creates its pseudo x-vector by averaging x-vectors far away from the original, which in most cases will yield an x-vector that is also dissimilar from the original, acting as a dissimilarity constraint.

For the A-A scenario, we observe increases in the EER compared to the baseline for all settings, varying from an increase of 2% (Female VCTK Test) to 12.6% (Male LibriSpeech Dev). Similarly the C_{llr}^{min}improves across all data subsets, with increases from 0.04 to 0.13. We also note that the values of $C_{llr}^{min}$are close to a perfect score of 1 in many cases, indicating very strong anonymization performance and implying that two versions of the same voice anonymized are rarely confused with one another.

We observe a small increase in Word Error Rate (WER) across all of the datasets and dataset splits, when comparing the results from the large language model used in

Dataset	**Gender**	**Scenario**	**Development**		**Test**	
			EER (%)	$\mathbf{C_{llr}^{min}}$	**EER (%)**	$\mathbf{C_{llr}^{min}}$
LibriSpeech	F	O-O	8.7	0.30	7.7	0.18
		O-A	46.9(-3.3)	0.97(-0.03)	40.3(-6.9)	0.95(-0.04)
		A-A	45.0(+8.2)	0.97(+0.07)	44.2(+12.0)	0.97(+0.13)
	M	O-O	1.2	0.03	1.1	0.04
		O-A	53.3(-4.5)	0.99(-0.01)	48.1(-4.0)	1.00(-0.00)
		A-A	46.7(+12.6)	0.97(+0.10)	43.4(+6.7)	0.98(+0.07)
VCTK (diff)	F	O-O	2.9	0.10	4.9	0.17
		O-A	46.0(-4.0)	0.96(-0.03)	44.6(-3.4)	0.98(-0.02)
		A-A	35.3(+9.1)	0.87(+0.11)	33.7(+2.0)	0.89(+0.04)
	M	O-O	1.4	0.05	2.1	0.07
		O-A	53.0(-0.9)	1.00(-0.00)	47.6(-6.2)	0.99(-0.01)
		A-A	36.0(+5.1)	0.92(+0.08)	40.2(+9.2)	0.93(+0.09)

Table 4.1: Speaker verifiability results for the pretrained ASV_{eval} model. Results for our anonymization method with 1 GMM component and σ^2_{99} PCA, without forced dissimilarity. In parenthesis we report the difference with the baseline system.

Dataset	**Split**	**Trials**	**Female**	**Male**	**Total**
Librispeech	Dev.	Target	704	644	1348
		Non-target	14566	12796	27362
	Test	Target	548	449	997
		Non-target	11196	9457	20653
VCTK (Diff.)	Dev.	Target	1781	2015	3796
		Non-target	13219	12985	26204
	Test	Target	1944	1742	3686
		Non-target	13056	13258	26314

Table 4.2: Details of number of trials for each of the datasets and their respective splits.

Dataset	**Audio Type**	**Dev. WER (%)**	**Test WER (%)**
LibriSpeech	Original	3.83	4.14
	Anonymized	10.02 (+3.63)	7.09 (+0.36)
VCTK	Original	10.79	12.81
	Anonymized	16.3 (+0.91)	16.98 (+1.75)

Table 4.3: WER rates for original and anonymized voices on the datasets specified in the VPC. Results are produced using 1 GMM component and 99% variance retained. In parenthesis we report the difference with the baseline system.

Dataset	**Gen.**	**Scenario**	**ZEBRA**		**Linkability**
			$\mathbf{D}_{\mathbf{ECE}}$	$\mathbf{log}_{10}(\mathbf{l})$	$D_{\leftrightarrow}^{\mathrm{sys}}$
LibriSpeech	F	O-O	0.58	3.98(C)	0.90
		O-A	0.03(+0.03)	0.82(+0.51)(A)	0.15(+0.08)
		A-A	0.02(-0.09)	0.72(-1.77)(A)	0.09(-0.20)
	M	O-O	0.69	3.92(C)	0.96
		O-A	0.00(+0.00)	0.16(-0.12)(A)	0.06(-0.02)
		A-A	0.02(-0.05)	0.50(-1.90)(A)	0.11(-0.09)
VCTK (diff)	F	O-O	0.59	3.65(C)	0.88
		O-A	0.01(+0.01)	0.74(+0.61)(A)	0.07(+0.02)
		A-A	0.08(-0.03)	1.43(-0.44)(B)	0.24(-0.04)
	M	O-O	0.67	3.92(C)	0.95
		O-A	0.01(+0.01)	1.17(+1.17)(B)	0.06(-0.00)
		A-A	0.05(-0.06)	1.86(+0.62)(B)	0.14(-0.16)

Table 4.4: Results for the ZEBRA and Linkability metrics evaluated in the Voice Privacy Challenge for our system using one GMM component and 99% of variance retained. Presented results are for the test split of both datasets. Difference from the x-vector baseline is given in brackets.

the VPC evaluation, as shown in Table 4.3. Overall the increases in WER are fairly small and the results slightly worse than those of the VPC x-vector baseline.

4.4.1.3 Additional Metrics

In this section we calculate the values of the additional metrics proposed during the VPC: (i) ZEBRA Framework [108] (ii) Linkability [91] and (iii) Voice Similarity Matrices [109] Metrics. All of these metrics are computed using the existing scores output by the speaker verifiability model.

We focus on the results for the test split of both datasets.

ZEBRA framework. Table 4.4 shows the results for $\mathrm{D}_{\mathrm{ECE}}$ and $\log_{10}(\mathrm{l})$ computed with the ZEBRA framework. The expected privacy disclosure, $\mathrm{D}_{\mathrm{ECE}}$, gives a score for

Dataset	Split	De_{ID}		G_{VD}	
		M	F	M	F
LibriSpeech	Dev	0.99 (-0.01)	0.97 (-0.03)	-3.69 (+5.07)	-1.96 (+7.21)
	Test	0.99 (-0.01)	0.96 (-0.02)	-2.80 (+6.18)	-2.74 (+7.33)
VCTK	Dev	1.00 (-0.00)	0.97 (-0.02)	-2.85 (+9.82)	-2.53 (+6.28)
	Test	1.00 (-0.00)	0.98 (-0.01)	-2.93 (+8.80)	-2.92 (+7.36)

Table 4.5: De-Identification and Voice Distinctiveness (Gain) results derived from Voice Similarity Matrices. Results are calculated for our system using 1 GMM component and 99% variance captured by PCA. Difference from x-vector baseline shown in brackets

the average level of protection afforded to a population, with a score of 0 corresponding to *perfect privacy* and is thus optimal. The worst case privacy disclosure, $\log_{10}(l)$, gives a score for the protection afforded to the worst individual, again with 0 being optimal. We observe that our system performs slightly worse than the baseline in the O-A scenario for both metrics, although still retains an A grade for the worst case in all but one case. For the A-A scenario we improve over the x-vector baseline, with values for $\mathrm{D_{ECE}}$ becoming lower across all data subsets. The worst-case privacy disclosure, $\log_{10}(l)$, also improves in the A-A scenario, with the exception of the male VCTK (diff) dataset. Furthermore the VCTK (diff) dataset letter grades are B, implying an adversary would be incorrect once in every 10 to 100 attempts [108]. These results for $\log_{10}(l)$ suggest that our system does not perform equally well for all individuals, causing a high value for VCTK in particular due to poor anonymization of a specific (or several) individual(s).

Linkability. Table 4.4 shows the linkability, $D_{\leftrightarrow}^{\mathrm{sys}}$, results. The linkability measures the difference between the target and non-target score distributions and can capture anonymization problems not detected by metrics such as $\mathrm{C_{llr}^{min}}$. For linkability the optimal score is 0. These results display a similar pattern to the other population wide metrics. For the O-A scenario we have variable results, with increases in $D_{\leftrightarrow}^{\mathrm{sys}}$ for both female datasets and no change and a decrease for the male datasets. For the A-A scenario we see large decreases in $D_{\leftrightarrow}^{\mathrm{sys}}$, showing a clear reduction in linkability. These numbers still remain higher than those for O-A, but the performance disparity between the two scenarios is reduced.

Voice Similarity Matrix Metrics. Table 4.5 shows the De_{ID} and G_{VD} results calculated from voice similarity matrices for our datasets. Deidentification, De_{ID}, measures the ease with which speaker can be linked between original and anonymised, with scores of 1 being optimal. The results for De_{ID} show a slight degradation in performance from the baseline, which had close to perfect scores of 1 (> 0.99) for all combinations of dataset and split. Voice distinctiveness, G_{VD}, measures how distinctive individual voices are, with greater than 0 dB indicating more distinctive voices and less than 0 being less distinctive voices in the anonymized space. For G_{VD} we see an improvement for all metrics, with the absolute values being better for female, but the magnitude of changes overall being slightly larger for males. This increase in voice distinctiveness compared to the baseline clearly demonstrates that our x-vector generation technique produces more diverse anonymous voices than the original technique.

4.4.2 Alternate ASV Evaluation

Our proposed system utilizes x-vectors for replacing the identity component of the audio. The VPC evaluation framework and associated metrics also use a state-of-the-art x-vector extractor, which could cause masking of potential issues, if the anonymization properties were not maintained into other spaces. As such we validate that the anonymization results hold when also computed with an i-vector [36]-based speaker verification system.

Setup. We repeat the speaker verification experiments conducted in the VPC analysis, using the pre-trained i-vector model found in the Kaldi examples[1]. The features are 24 MFCCs with a frame length of 25ms, with an energy-based voice activity detection (VAD) system applied to determine which frames contain speech. The UBM is a 2048 component full-covariance GMM. The i-vector model extracts a 400 dimensional i-vector and is trained using the 100,000 longest utterances from the VoxCeleb 1 and 2 training datasets. A PLDA backend, which uses an LDA dimension of 200, is used for scoring the utterances, with the system achieving an EER of 5.3% on the VoxCeleb test datasets.

We extract i-vectors using this trained system for the original and anonymous audio for each of the datasets that we evaluated across the scenarios in section 4.4.1. This allows us to compare the x-vector and i-vector results directly.

We focus on the EER and $\mathrm{C}_{\mathrm{llr}}^{\mathrm{min}}$metrics and compare the results on the test splits of the LibriSpeech test and VCTK (diff) test datasets.

Results. Figure 4.6 shows the EER and $\mathrm{C}_{\mathrm{llr}}^{\mathrm{min}}$results for the x-vector and i-vector systems on the test splits for all of the anonymization scenarios. In most cases performance is similar for the x-vector and i-vector systems, with only small differences

---

[1] https://kaldi-asr.org/models/m7

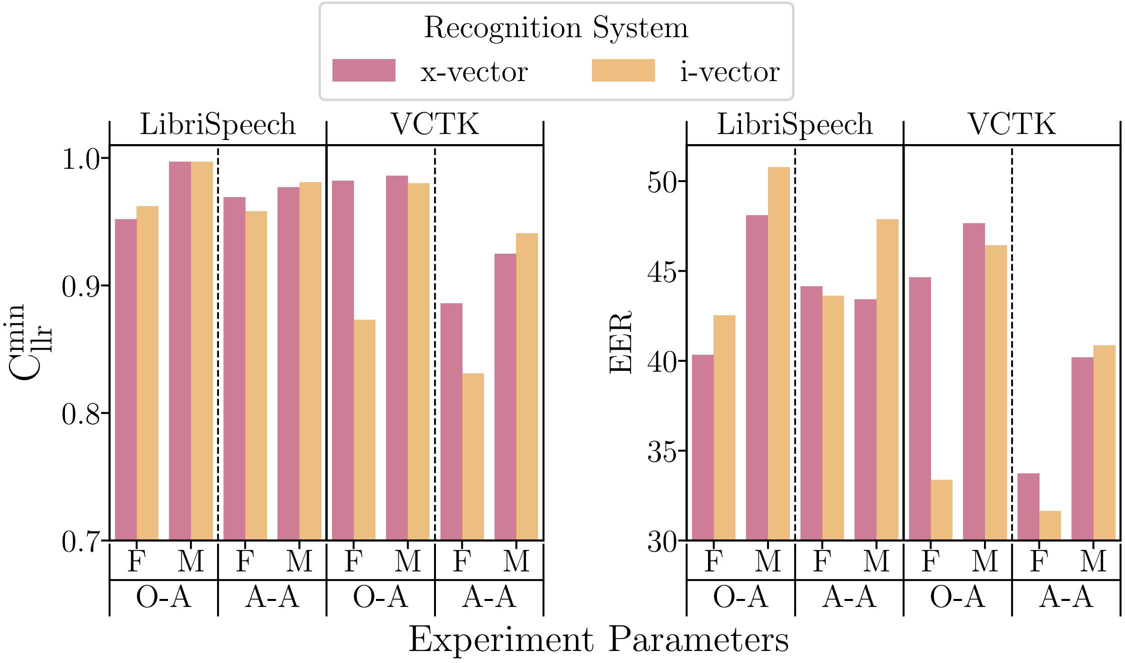


Figure 4.6: Alternate ASV System results for the test splits of both datasets on all scenarios, comparing the $C_{llr}^{min}$and EER when calculated using the VPC x-vector system and an alternate i-vector system. Note the y-axes does not start at 0.

observed between the two systems. In general the difference in performance is more pronounced on female voices than male voices. The largest degradation in performance come on the female split of the VCTK database, in both the O-A and A-A scenarios.

These large drops in EER and C_{llr}^{min}point to the potential for some overspecialisation in the x-vector space for female performance. Alternatively, it could be related to dataset imbalances between males and females, the effects of which were observed in examining the x-vector extractor in Section 4.3.2 and in the VPC results in Section 4.4.1.2. Overall, these results suggest that the anonymization is not limited only to the x-vector space.

4.4.3 Forced Dissimilarity

As mentioned previously the forced dissimilarity (FD) θ_{FD} parameter repeats the GMM sampling of a pseudo x-vector if the x-vector is too near to the original user's voice. However the specific value of θ_{FD} used may impact the anonymization performance of the system and as such we evaluate the system with varying θ_{FD} values.

Setup. We evaluate the effect of θ_{FD} using the VPC framework, again focusing on the results for $\mathrm{C}_{\mathrm{llr}}^{\mathrm{min}}$. We run the evaluation framework with $\theta_{FD} \in [0.2, 1]$ (1 corresponds to no similarity constraint).

Results. Figure 4.7 presents the $\mathrm{C}_{\mathrm{llr}}^{\mathrm{min}}$for both the O-A and A-A configurations, for the test splits of both datasets and for both genders. We observe that for the O-A scenario the values for $\mathrm{C}_{\mathrm{llr}}^{\mathrm{min}}$remain high and slightly increase with a smaller θ_{FD}. This result is as expected, as by forcing less similarity – and thus a bigger distance between the original voice (and thus the enrolment voice) and the target x-vector – the resultant voice also becomes increasingly distant from the original and thus the enrolment voice.

For the A-A scenario we observe a different effect, in that with smaller θ_{FD}, i.e. forcing more distance from the original voice, the $\mathrm{C}_{\mathrm{llr}}^{\mathrm{min}}$decreases. This is likely because the available area of the x-vector hyperspace is restricted by θ_{FD}, meaning that when anonymising the same voice twice, both the target x-vectors come from a region that becomes increasing small as θ_{FD} decreases, resulting in voices that are more similar. This is potentially problematic and shows that small values of θ_{FD} should not be used.

The effects on the overall WER for the test split of all of the datasets are seen in Figure 4.8. We observe that for both datasets, as θ_{FD} decreases we experience an increase in WER. This may be due to links remaining between the non-speaker identity components of the anonymization and the original speaker identity, resulting in a conflict between features when transforming to a very different x-vector and thus worse performance.

Overall our results suggest that values of θ_{FD} need to be chosen carefully, and small values should be avoided due to their negative impact on WER and the A-A scenario. If using FD we would recommend a value of 0.9, as this yields strong performance in O-A, A-A and WER, but gives the guarantee of avoiding a transformation to a very similar voice.

4.4.4 Examining Resultant X-Vectors

The overall architecture for the system relies on the assumption that the voice can be decoupled into the identifying components (described by the x-vector) and the speech content (described by the bottleneck features and F_0) [43]. In order to examine this assumption, we evaluate the resultant x-vectors produced by the system (i.e. the x-vectors extracted from the final audio). If the assumption holds, we expect to see

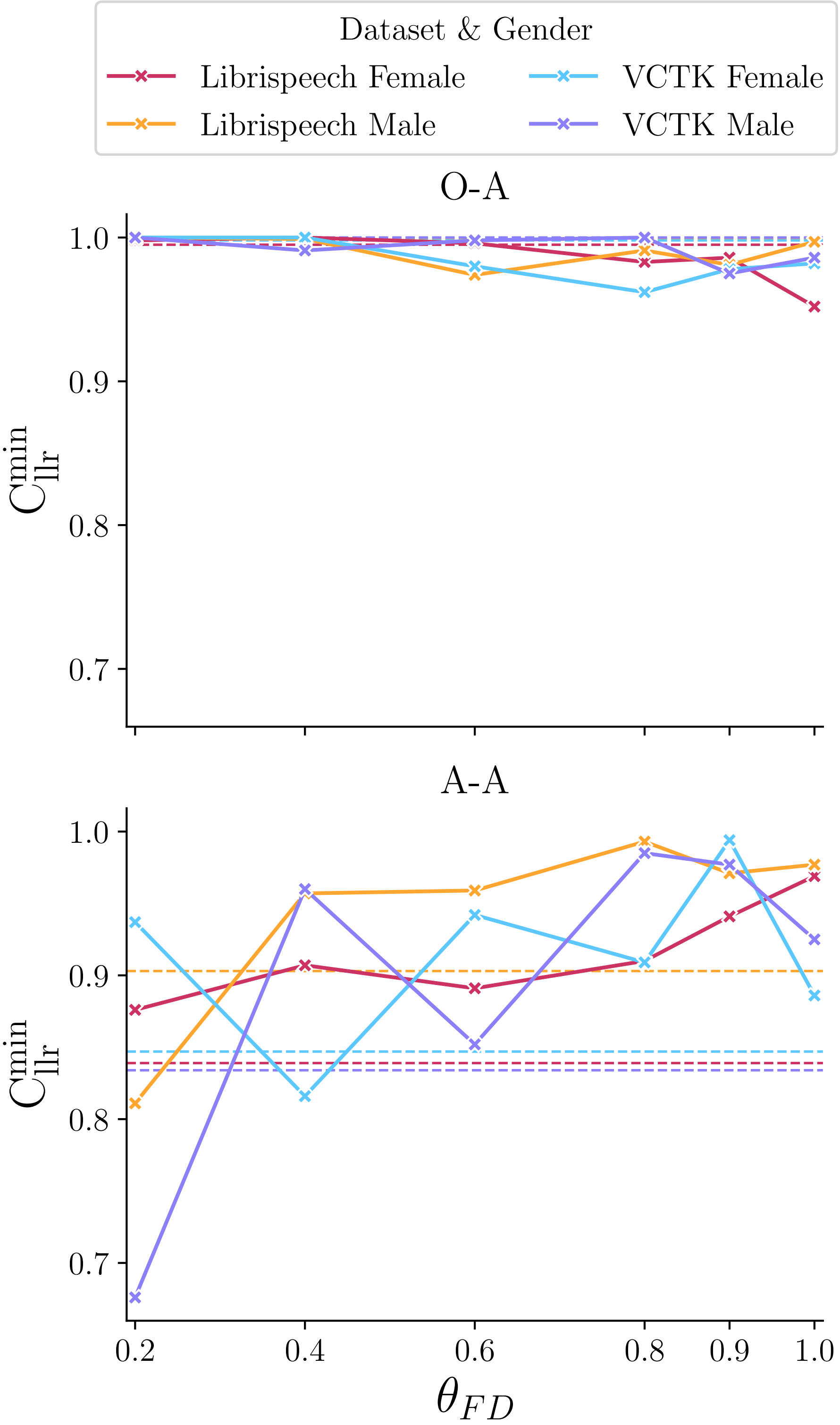


Figure 4.7: Plots showing $C_{\mathrm{llr}}^{\mathrm{min}}$values for the test split of the datasets for varied values of the forced dissimilarity parameter θ_{FD} for the model, for both scenarios in the VPC challenge. The values of the Baseline system are shown with the dashed lines.

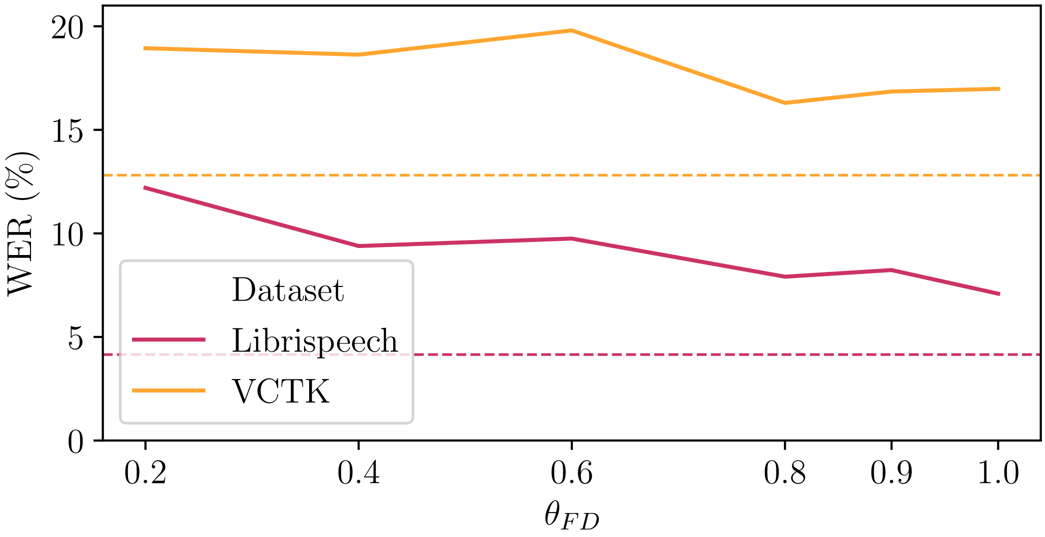


Figure 4.8: Plots showing WER as a function of the forced dissimilarity parameter θ_{FD}. WER values for the original datasets are marked with the dashed lines.

that the resultant x-vectors are as distant from the original voice as any other voice, and that they closely resemble the target x-vector supplied to the synthesis models.

Setup. We perform this analysis on the VCTK test dataset. Anonymized audio is generated for all of the enrolment utterances for each user, with the same pseudo target being used for each of these utterances. Resultant x-vectors are then extracted from this anonymized audio, using the same x-vector extractor as used for generating the dataset to train the GMM. We analyse the distances between the pseudo outcome and the pseudo target using the cosine distance between them, as well as comparing this to other sets of distances between original voices and the newly anonymized voices. The x-vectors used for computing these distances are extracted with the extractor used by our system and not the separate VPC evaluation x-vector extractor. We also compute two reference distributions, composed of the original voices compared with themselves and of the original voices compared with different voices.

Results. Figure 4.9 shows two sets of histogram distributions of the cosine similarity scores, for the anonymized audio. In the upper plot of Figure 4.9 we examine the x-vectors of the final audio post synthesis, with the target x-vector supplied in synthesis. We see that the distribution of these does not mirror that of a normal voice compared with itself and instead is shifted left-ward (i.e. less similar) and has a further spread distribution. This suggests that the synthesis algorithms do not result in audio that can be considered (in general) to be spoken by the same voice as the target vector,

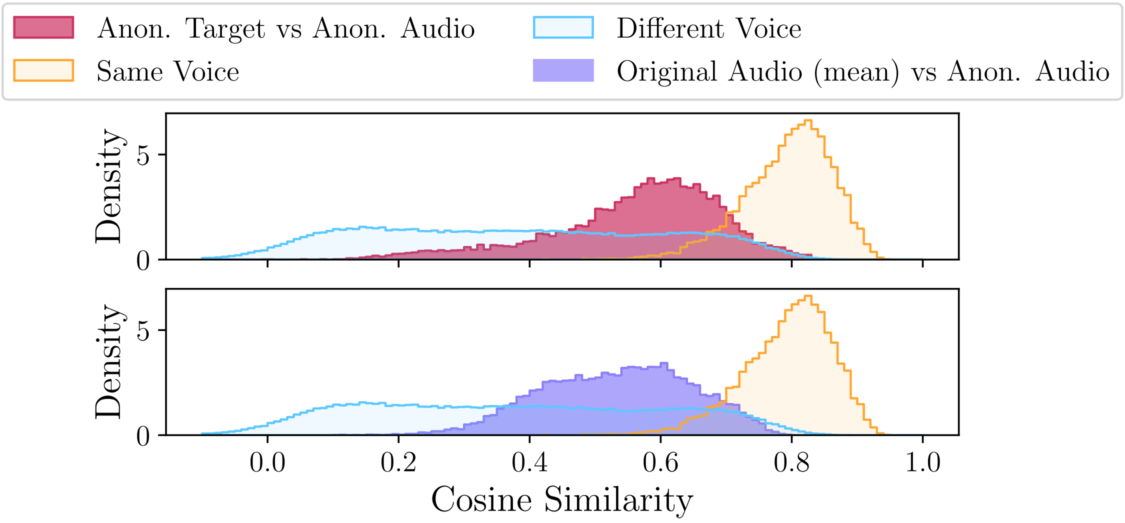


Figure 4.9: Similarity (cosine) comparison of resultant x-vectors with the target generated by the GMM. We observe that voices are more dissimilar than two copies of the same (original) voice, but not as dissimilar as if they were a different voice entirely.

degrading anonymization performance in the process.

The bottom plot, comparing the original audio's mean x-vector with the anonymized version of that voice, shows a distribution that does not match either of our reference distributions. In an optimal system, the similarity distribution should match the different voice distribution, however we observe that the distribution is shifted to the right and more condensed. This implies that the anonymized voices are more similar to their original voice than would be ideal, suggesting that the synthesis process must be retaining some bias toward the original voice.

Taken together, these results highlight deficiencies in the audio synthesis process, showing that it comes up short in recreating the target accurately and suggesting that the assumption that the non-x-vector features (BN and F_0) do not contain identifying information does not hold.

4.4.5 Subjective Naturalness

As part of the VPC the organisers computed subjective results for each of the submissions. As we have improved the parameters within our system since the VPC, we do not report the results provided by the challenge organisers here, but they can be found in [146]. Instead, we conduct Mean Opinion Score (MOS) tests to assess the subjective naturalness of the voices produced by our system.

Setup. We use the Amazon Mechanical Turk platform to conduct our MOS tests. We follow the ITU-T Recommendation P.808 [67] for conducting these listening tests, with each audio file being assessed on scale from 1 (poor) to 5 (excellent). As directed by the recommendation, we implement a qualification phase, in which we verify participants are native English speakers, are using headphones, are in a quiet environment and of normal hearing. Participant training is conducted with 5 samples that are selected to cover the range of sample qualities. Training samples are identical for all workers. Throughout the training and rating phases gold standard and trapping questions are used, as per the recommendation, to ensure participants are attentive to the task. Workers who fail to answer these correctly have their scores discarded.

We perform our MOS for 100 randomly selected anonymised audio files from the VCTK test dataset. We use the original audio, the baseline system, our proposed system, and our proposed system with $\theta_{FD} = 0.9$, giving a total of 400 audio files to be scored. We use the same random audio files for each set i.e. the original utterance, and 3 anonymised versions of it.

Workers are presented with sets of 12 samples at a time and paid for each set of 12 they complete. Each of these 12 samples contains at least one audio sample from each of the four conditions (Original + 3 anonymisation methods), to prevent bias introduced by workers who do not rate all samples. Following the p.808 recommendation we incentivise workers financially to rate at least half of the total samples. Each sample is rated by at least 6 distinct workers, with an average of 9.57 ratings per sample. We received ethical approval from our institution for this study, reference CS_C1A_21_010.

Results. The computed MOS can be seen in Table 4.6. We conduct statistical significance testing, after removing bias, pairwise between each of the audio sets using a Mann-Whitney U test, following the method given in [126]. We find that the differences between the MOS are statistically significant at the 1% level for all pairs of audio sets, except between the two sets based on the Proposed System. Full test statistics and p-values are reported in Appendix C

We observe that all of the audio anonymization systems have much lower opinion scores that the original audio. This highlights the need for improvement in the overall anonymization method.

We also observe that our proposed technique has a slightly worse MOS than the Baseline system, scoring 2.50 and 2.82 respectively. Whilst this difference is fairly small, it is statistically significant and suggests that the baseline method produces x-vectors that are more natural sounding than our method. This difference could be

Type	Total Ratings	MOS	Std. Dev.
Original	1039	4.432146	0.648822
Baseline	926	2.816415	0.800534
Proposed System	918	2.501089	0.860815
Proposed System w/ $\theta_{FD} = 0.9$	945	2.503704	0.826108

Table 4.6: Mean Opinion Scores for original audio and varied audio creation system. The differences between pairs of audio sets are statistically significant ($p < 0.01$) for all pairs except the two versions of the proposed system.

because the Gaussian space captured by the GMM is not guaranteed to contain only natural sounding voices, and thus can contain x-vectors that produce poor audio. The larger standard deviation also suggests this could be the case. An alternate hypothesis is that the averaging of the baseline produces xvectors that are more central in the hyperspace, which it is easier for the synthesis method to produce audio for.

4.5 Discussion

In this section, we discuss the overall anonymization capabilities of our system and possible concerns that need further investigation, as well as signposting directions for future work to investigate in order to improve this technique, as well as others based on NSF models.

4.5.1 Overall Anonymisation Assessment

Our experiments focused on two key scenarios throughout, one where the original voice is compared to the anonymized voice and one where two instances of the same voice anonymized are compared with one another. The x-vector selection technique we developed is intended to maximise performance on the second one of these scenarios, however the two are interlinked, with poor performance on one impacting the other, as well as the overall usefulness of the system.

Our experiments demonstrate that our x-vector selection technique improves the anonymization performance of the system, particularly in improving the diversity of the anonymized voices. We observe that values of $\mathrm{C}_{\mathrm{llr}}^{\mathrm{min}}$ approaches a perfect score in many scenarios, indicating strong anonymization, although worse performance was generally observed in the female voices analysed. This could be caused by the x-vector extractor's weaker performance on female voices, meaning the female voices occupy a smaller space. Potential causes of this lopsided x-vector extractor performance could be due to imbalanced datasets, or due to female voices containing less spectral information than male voices to begin with, particularly when a single extractor must handle all voices, covering a wider spectrum than a gender specific extractor. Further investigation will be required in order to better understand and mitigate this issue.

The results from the additional metrics developed for the voice privacy challenge also achieved strong results, although we observe some worse performance in similar scenarios as to the original metrics. In particular the results for the $\log_{10}(l)$ metric show that for some individuals performance may be poor. This could be an effect of the random pseudo x-vector selection from the GMM, or could be explained with our analysis of resultant x-vectors in Section 4.4.4.

The resultant x-vectors analysis in Section 4.4.4 also showed that the outcome utterances are not close enough to the x-vector target to be considered the same voice as it. Improvement in this aspect of the system is also likely to improve anonymization results further. These results also show that separation between identity and speech content is not perfect within the system. Whilst this finding does not appear to impact the metrics assessed with the VPC, it does imply that an adversary attempting de-anonymization gains some information about the original voice from measuring its distance to other voices, and thus improvements in the process would result in further privacy gains. One potential avenue of exploration could be to reduce the feature size of the xvectors and bottleneck features, to reduce the information that could be present in both of them.

4.5.2 Speech Quality Results

The speech quality of the anonymised audio was assessed by both WER of the resulting audio, as well as MOS tests to rate its subjective quality.

The WER results for both the baseline and our proposed technique were worse than for the unanonymised audio. Across all four datasets our proposed system performed worse (0.36% to 3.63%) than the baseline. In terms of MOS, our proposed system scored 2.5, compared to 2.8 for the Baseline and 4.43 for the original audio.

These sets of results highlight two problems that need further attention. Firstly, the x-vector anonymization and re-synthesis system that underpins both the baseline and our proposed system needs further improvement, to increase the naturalness of audio that is generated using it. Secondly, the worse performance of our proposed system in terms of speech quality needs further investigation, to determine why the x-vectors produced by our method lead to worse audio. One potential cause could be that the averaging method used by the baseline can not have extreme values, as the averaging method constrains the xvectors to values more central in the space. Our method does not have such a constraint and there is no guarantee that all the x-vectors that are modelled by the GMM-PCA space represent naturally sounding speakers, meaning that poorly performing speakers could be sampled from this space.

Further work could investigate optimising the x-vector sampling process to produce more natural speakers, as opposed to our focus of encouraging diversity in the produced x-vectors.

4.6 Conclusions

In this chapter we presented our technique for speaker anonymization, by utilising GMMs to generate pseudo x-vectors to transform voices to.

We demonstrate that our system performs particularly strongly when comparing two (differently) anonymised versions of the same voice, outperform the VPC baseline by a large margin. We also investigate the properties of the x-vectors taken from the produced audio. These experiments show that there remains space to further optimise the synthesis models, producing output audio that more closely resembles the target x-vectors, and thus differs more from the original voice, improving anonymization and privacy.

Finally we discuss future avenues to be explored in developing this system further, in particular highlighting the need for improved word error rates and better naturalness and intelligibility, if this system (or others based on similar synthesis models) are to be able to provide a useful solution for voice anonymization.

Whilst this system can be used to protect voices stored at rest by an end service provider, it would require the user of the voice system to trust the service provider to keep their voice safe. Some users may instead wish to have agency over the protection of their own voice trait, preferring a solution that allows them to protect their own voice. They may also wish to do so in a way that can be applied to audio as it is produced and that removes additional speech information that could be used to de-anonymise the audio. In the following chapter we develop and evaluate such a system.

Chapter 5

Enabling Voice Privacy for Users of Remote Speech Systems

Contents

In Chapter 4 we developed and evaluated a scheme for anonymising voice recordings. This scheme aimed to remove just the identifying information and the speech, and leave other speech traits untouched. However, retaining these other speech traits leaves a side channel through which an adversary may be able to de-anonymise audio samples. Furthermore, this scheme was developed to be applied to audio post-hoc and is more applicable to end service providers than individual users interacting with services. This leaves users have to depend on the end service provider to keep their voice data private, giving them minimal agency over the protection of their voice.

In this chapter we aim to improve this situation. We develop the AltVoice system, a first attempt in allowing users to anonymise their own voice data as they provide it to remote systems. In brief, the system uses a speech-to-text followed by a text-to-speech pipeline to remove all identifying information and replace the speaker identity. Reducing the audio to text ensures that all identifying information (save for word choices) is removed from the eventual audio.

5.1 Introduction

Systems increasingly utilise remote voice based services for many interactions with users. These remote systems have also expanded in their capabilities in recent times, now supporting voice authentication to unlock sensitive services (e.g [38,64]), promising smoother and more secure customer interactions by removing the need for users to remember passwords. Whilst this trend is not a new one, it has been accelerated by the Covid-19 pandemic, with people in many jurisdictions encouraged to avoid in-person interactions.

However, this increased use of voice interfaces comes with significant privacy risks for users. Leaks of voice data (e.g. recorded phone calls) not only expose potentially sensitive conversations, but also perpetually discloses the user's voice trait. Drawing a password-based comparison, while a password can be effortlessly changed after a leak of a password dataset, one's voice trait is unchangeable: once a user's voice is leaked, the secrecy of their voice trait may be compromised forever. This opens the door for various attacks: both identity linkage and even impersonation may be viable for an adversary who has obtained a sample of a victim's voice, as demonstrated by the attack proposed in Chapter 3. Furthermore, with the increased prevalence of public videos and audio on social media, the secrecy of one's trait might already be compromised and freely available online.

These concerns are exacerbated by quick advances in voice cloning technology, which shows that it is possible to accurately clone the voice of an individual with just a few seconds of victim audio [68]. If the cloned voice is accurate enough, it may be possible to use it directly to impersonate the victim, or to bypass automated voice authentication techniques [103, 151]. Furthermore these techniques improve at a fast pace, and the ability to impersonate a victim's voice is only going to become easier.

Recently work has begun on solutions to protect the privacy of users interacting with voice systems. Several systems have been proposed for service providers, allowing them to protect audio and voice prints they obtain [54, 102, 115, 167]. However, these systems rely on end users adequately protecting their voice information: when such voice information is disclosed or leaked, the systems cannot be safely used any longer. Furthermore if the user does not trust the provider to protect their voice, then they can not use the system.

More recently, the VoicePrivacy challenge [147] has taken place, in which participants designed systems that would allow a user to anonymise their voice. The challenge's aim was twofold, to hide speaker identity as much as possible while at the same time limiting the distortion of other speech characteristics to the minimum; part of the goal was to retain as much linguistic content from the original voice. A more in depth discussion of the VoicePrivacy challenge can be found in Chapter 4.2.1.

In this chapter, we tackle the challenge of protecting a user's voice secrecy by introducing the AltVoice system, which allows users to replace their own voice with generated unique voices on demand and as required. Using AltVoice, users can choose to not re-use their own voice across a multitude of services, but instead they can easily create new unique and secret voices at will, avoiding exposing their original voice with untrusted third-parties. Differently from previous systems, to create a new voice, AltVoice strips *all* identity information out of the user's voice, this way, it can grant fundamental privacy properties: revocability, unlinkability and noninvertibility. The proposed system makes use of Speech-to-Text and Text-to-Speech components, which reduce the voice signal to a sequence of words at the intermediate point, resulting in the maximal possible protection of the voice without changing the word content of the speech. We experimentally validate the performance of our proposed system both in terms of privacy protection and reliability of recognition performance when using authentication systems with a generated voice. To do this we consider two different attacker models, one aimed at compromising the privacy of a user via an identity linkage attack, with the other aimed at impersonating a user who enrolled with a AltVoice-generated voice. We examine the performance in authentication scenarios using a trained text-independent speaker recognition model based on Generalised End-to-End Loss (GE2E) [160]. Finally we examine the effects of differing the algorithm used to create new voice identities and outline the ways in which this system can be easily upgraded over time, as further improvements are made to speech to text and text to speech systems. Our results show the system successfully allows a user to assume a new identity and resists attacks under both of our threat models. However, the results also reveal that further work is needed in ensuring that generated identities are sufficiently unique and ensuring that the audio generated by the system is both natural sounding and intelligible (i.e. words are not mangled as they pass through the system).

The key contributions of this Chapter are:

1. Proposal and implementation of a system (AltVoice) for allowing users to interact with voice processing systems while hiding their voice trait and without cooperation from the end service provider.

2. Identification of six different methods to generate private voice identities given the AltVoice architecture, with each identity generation method relying on a user-known secret.

3. An evaluation of the above system, highlighting its trade-offs and detailing the extent to which the various individual components impact the overall system performance.

5.2 Background

In this section, we provide an overview of typical remote voice processing systems, as well as an overview of the elements used in our proposed system. A more in depth discussion of the components can be found in Chapter 2.

5.2.1 Remote Voice Processing Systems

Remote voice processing systems (VPS) have been common for a number of years, and encompass a large variety of services. Typically the remote aspect of the services means that the interaction occurs through a telephone call, although recently other mediums, such as VoIP programs may be used instead.

The most simple of remote systems are phone calls to call centres, such as those used to provide customer service. Often these calls are recorded and there is often an operator listening at the other end (or a machine processing inputs). Thus even the least technologically advanced of these systems have privacy concerns for a user, as their voice trait may be recorded and subsequently leaked. Furthermore users of phones may be unwitting participants in remote voice systems that perform surveillance on behalf of governments or other state entities [46].

These remote systems can also be augmented with additional features. For example, the phone operator may have software that automatically transcribes phone calls, either at the time or at a later date. Increasingly extra features are included for voice trait based authentication or identification, to avoid the need for users to answer cumbersome questions to confirm who they are. For example, in the United Kingdom major banks such as HSBC [64] and FirstDirect [38], telephone networks such as Vodafone [154], and Her Majesty's Revenue and Customs [63], use voice biometrics as part of their telephone banking service.

Our proposed system should ideally be useable for all types of remote system, including those with authentication. This requires that any private (or anonymous) voices used by our system can also be used for authentication. If this is not the case the system will still have utility and protect the privacy of users utilising remote speech systems without voice based authentication.

5.2.2 Speaker Authentication Systems

Speaker authentication and identification systems can be split into two categories: *text-independent* (TI) and *text-dependent* (TD). *Text-independent* systems operate on any utterance, whereas *text-dependent* systems require the same utterance to be spoken at enrolment and verification time. Usually this utterance is fixed for all users.

It is also important to distinguish between speaker authentication and identification. In authentication systems the user claims to be a specific member and then provides speech data which is used to prove the truth of the claim, to some degree of certainty. In an identification system an attempt is made to identify who speaks an utterance, but there is no claim of who is speaking beforehand (although a set of candidates may be provided for closed set identification) and we do not necessarily have a confidence threshold that must be met. In this chapter we mostly focus on speaker authentication.

State of the art systems for speaker authentication and identification are based on Deep Neural Networks (DNNs) [137,160], which take an audio utterance and produce an embedding vector from it, representing the speaker's identity. As such, the distance between vectors, usually measured by cosine distance, can be used to determine if two samples are spoken by the same user. State of the art systems achieve Equal Error Rates as low as 3.3% for TI and 2.38% for TD [160].

A full system also features an enrolment stage, in which several samples are fed to the system and the embeddings used to create a template for the user, usually by taking the mean embedding. Later, at verification time, the computed embedding from the sample utterance is compared to the claimed template. If the distance between template and sample embedding is below a pre-determined threshold it is accepted as spoken by that user, otherwise it is rejected.

Speaker identification systems also use these embeddings, determining the speaker to be the nearest template to a given utterance.

5.2.3 Speech-to-Text Systems

Speech-to-Text (STT) systems produce a transcription of audio that is input to them. Traditionally these systems have been built with large amounts of specialist knowledge, but more recently deep learning approaches have been applied to the problem, bringing performance improvements with it.

State of the art systems, such as Deepspeech [56], use DNNs, specifically Recurrent Neural Networks, to turn spoken audio into character level transcriptions. These DNNs are trained in an end to end fashion, allowing them to use large datasets and become increasingly robust to variations between speakers, background noise, and other artefacts that may be present in the audio.

Deepspeech (and other similar networks) produce character level transcriptions as their direct output. A language model is applied to these character level transcriptions to fix errors and ensure valid words are presented, as opposed to phonetic spellings of words.

The accuracy of STT systems is typically calculated using the Word Error Rate (WER), which calculates the number of errors that occur when comparing the transcription to reference the text. State of the art STT systems achieve WER rates below 10%, with Microsoft's speech recognition system achieving rates of 5.1% [166], a competing system by Google achieving 5.6% [28] and the latest release version of Deepspeech achieving 7.06% [100].

5.2.4 Text-to-Speech Systems

Text-to-Speech (TTS) systems transform text into audio that appears to be spoken by a real voice. Most work has historically focused on single speaker TTS, where a large dataset from a single speaker is used to create a model for synthesising any chosen text.

Recently there has been increased focus on multi-speaker TTS, where a system can synthesise audio from either one of many speakers, or any arbitrary speaker. These systems consist of a synthesizer and vocoder. The synthesizer takes an input sequence of phonemes and an embedding that represents the target user and outputs a set of log-mel spectrograms. The vocoder then takes these spectrograms and produces a waveform from this.

The quality of the final output voice depends on several factors, including the quality of the speaker encoder that is used to generate embeddings at training, the quality of the vocoder and the quality of the synthesis network. Mean Opinion Scores (MOS) are used to assess the quality of the produced audio, where listening tests are performed by humans who give a score for each audio file.

State of the art approaches to TTS achieve MOS of 4.22 for voice in the training data, compared to 4.67 for ground truth audio [68]. On speakers unseen in training the MOS is typically worse.

If a TTS system is used to create completely artificial voices i.e never seen, then identity generation is typically performed by random sampling between 0 and 1 and then normalising the resulting embedding. Approaches to Voice Privacy using other techniques that create synthetic embeddings have also proposed averaging embeddings from a pool [43,138] and sampling from Gaussian Mixture Models to generate realistic looking embeddings [152]. We examine how these embedding selection techniques can be generalised for use with a multi-speaker TTS system later in section 5.4.1.3.

5.3 System Requirements

Here we describe the scenario, give a brief outline of how AltVoice works and present the threat model.

5.3.1 Scenario

A user of a remote VPS that uses and stores the user's voice data wants to protect the privacy of their voice trait. Specifically, users are concerned that an adversary

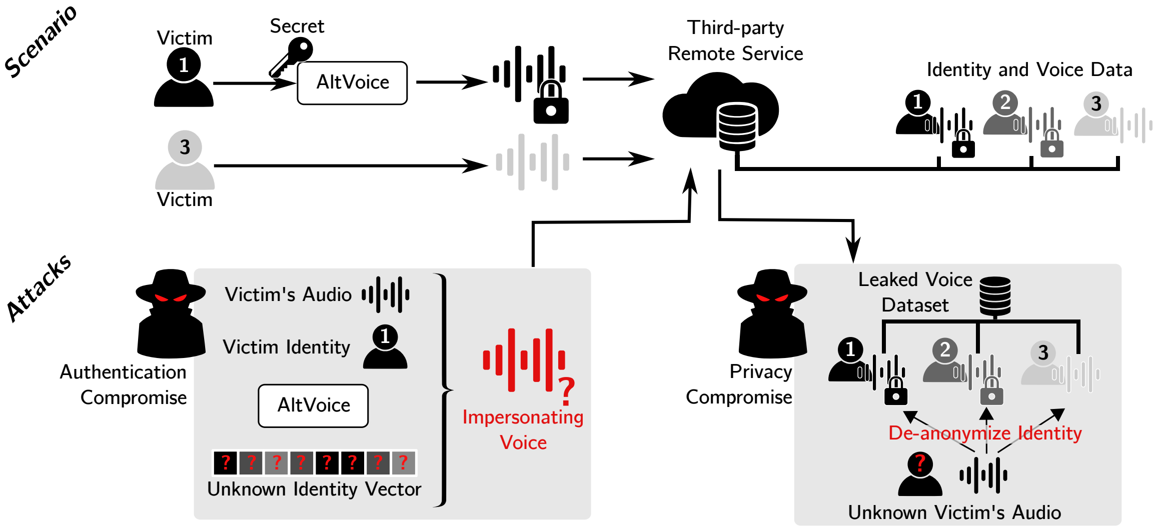


Figure 5.1: Scenario (top) and attacks (bottom). In the scenario, two users (victims) enrol into a remote voice processing service, which uses and stores users' voices and identity information. The first victim uses AltVoice with a secret to generate a private voice, the second victim uses their original voice. Two different attackers obtain a user's original voice and attempt two different attacks, either impersonating the victim or de-anonymising their identity. Using AltVoice protects the first victim from the two attackers.

could obtain their voice information, either by obtaining access to the remote system data or obtaining audio recordings of the user's voice elsewhere (e.g. audio from social media). Armed with the user's voice information, an adversary might be able to impersonate the user while interacting with the remote voice system, or they might be able to infer the user's identity by cross-referencing information from other sources. A representation of this scenario can be seen in Figure 5.1. Users would prefer to be able to use the voice-features provided by the VPS without having to disclose their unique own voice trait in the interactions.

This scenario encompasses situations such as phone banking, where the voice is used for authentication, as well as situations such as speaking to a customer service representative at a call centre, where the call audio may be recorded for later user.

5.3.2 AltVoice Design Goals

Requirements. To address the privacy concerns described in the previous section, a system has to *transform* user utterances in a way that *conceals* users' voice identity information, by substituting it with a generated identity. More specifically, a system must fulfil the following requirements:

- **R1:** Given a user utterance, the system produces a different utterance with identical word content, but different voice identity information.

- **R2:** The system does not require cooperation from third-party VPS that use voice information, only interaction with the end-user is required.

- **R3:** The identities of the system's generated voices are reproducible: given the same seed or secret, the system generates utterances with the same voice identity.

- **R4:** The system's produced utterances can not be linked to the original user's voice and vice-versa by examining the produced audio.

- **R5:** The diversity among system-generated voices resembles the diversity among natural voices. If diversity is not retained use-cases that require voice-uniqueness (e.g. recognition) may not have the same performance.

AltVoice Outline. In response to these requirements, we developed AltVoice. AltVoice strips voices of their identity information by transcribing an utterance into words and then synthesising the words into a new utterance emptied of the users' true voice trait. AltVoice generates fake voices based on a *user-known secret*, each secret corresponds to a fake voice identity. In Section 5.4, we argue how the system fulfils **R2** and **R3** by design. We use experiments with attack scenarios to validate **R4** and **R5** in Section 5.5.1 and 5.5.2. We also use automated speech-to-text transcribers and mean opinion scores to validate how AltVoice retains individual words present in utterances (**R1**) in Section 5.5.3 and 5.5.4.

5.3.3 Threat Model

In the threat scenario, a victim enrols into a VPS with an AltVoice-generated voice. We consider two attacks: (i) Voice de-anonimisation attacks and (ii) Voice impersonation attacks; a depiction of these attackers can be seen in Figure 5.1. We first introduce general attacker knowledge and capabilities and then detail each attacker.

General Attacker Capabilities. The adversary does not know the secret used by the victim with AltVoice to generate the VPS-enrolled voice. Nevertheless, adversaries have unlimited resources otherwise, specifically they have:

- Unlimited amounts of the victim's original audio.

- A copy of the anonymization system (AltVoice), to generate voices.

- A state-of-the-art voice identification system, which they can use to determine whether two utterances belong to the same individual.
- Knowledge that that the victim used AltVoice to enrol with the VPS (rather than the victim's original voice).

Privacy Attacker. In the privacy compromise attack, the attacker has obtained some of the victim's audio but they do not know the victim's identity. The attacker's goal is to de-anonymise the victim by cross-referencing the victim's audio against a voice dataset that contains links between users' identities and natural voices. In practice, the attacker checks whether the victim's audio matches voices in the voice dataset, when they find a match, they infer the victim's identity to be that associated with the matching voice in the dataset. We assume that the victim is in the voice dataset that the attacker has obtained.

We assume that the privacy attacker uses the voice data they have obtained to perform their attack and does not attempt to use other information, such as word choices or sentence lengths to infer the identity and deem such methods out of scope. Whilst stylometric approaches that use this information may help an attacker, it is hard to model them effectively and our proposed system is not designed to defend against them. Future work could examine methods that change the word content of utterances to defend against this style of attack.

Authentication System Attacker. In the authentication attack, the attacker has obtained the victim's audio and they know the victim's identity. The attacker's goal is to impersonate the user when interacting with a VPS which uses speaker recognition (where the victim is enrolled). In practice, the attacker can use AltVoice to generate a private voice utterance and attempt to log in to the VPS with the generated utterance.[1] Note that while we evaluate impersonation considering a single attacker's attempt, in practice a VPS typically allows multiple attempts before throttling, with the number of attempts depending on the VPS access control policies and is outside the scope of this work.

5.4 System Design

In order to meet the requirements set out in Section 5.3 we design the AltVoice system. The system aims to produce audio, conveying the same words as that being spoken by a user, but that appear (both to a speaker identification system and to a

---

[1] For this work, we consider text-independent speaker recognition is in place. We discuss text-dependency in Section 5.6.2.

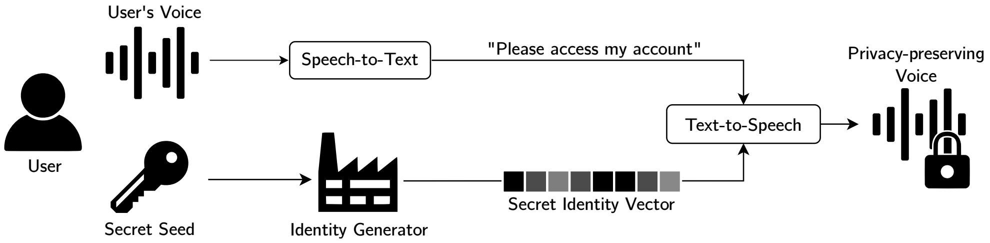


Figure 5.2: AltVoice system diagram showing the key components of the system and how they inter-operate with one another. AltVoice uses the user's voice and a user defined secret to generate a private voice which can be used with remote VPS.

listener) to be spoken by a different individual, in as close to real time as possible. In order to prevent any linkage between the initial identity and outcome identity of the audio, we design our system so that only a textual representation of the audio is present at the midpoint of the audio generation process.

We do this through sequential application of Speech-To-Text (STT) and Text-To-Speech (TTS) systems, converting the speech of the user into the text content of their speech, followed by turning this text back into speech spoken by a different speaker. The identity of this new speaker is provided by an identity generation system and we evaluate several candidate generation methods in this work. By reducing all the information from the original speech signal to just its text content at the midpoint of the system pipeline, we remove the maximal amount of speech information present, thus giving the strongest privacy guarantees possible short of changing the words spoken. An overview of the system can be seen in Figure 5.2.

In the remainder of this section we discuss the specific sub-systems we implemented to fulfil each part of the system in further detail.

5.4.1 Implementation

The AltVoice system is made up of several sub components. In this section we discuss the considerations that need to be made for each specific component, as well as detailing the specific implementations we selected for this version of AltVoice.

5.4.1.1 Speech to Text

The STT system forms the first part of the pipeline and its accuracy is critical for ensuring that the final spoken audio features the correct words. Errors that occur in this section will be propagated (and potentially worsened) through the rest of the model.

Another important consideration for the chosen model is its execution speed - if it is too slow then the lag between speech and transformed audio being produced will be increased. This is further impacted by the utterance length required to produce a transcription (some models require full sentences).

In our implementation of AltVoice, we utilise the Deepspeech system [56], which is freely available with pre-trained models [100]. Deepspeech has state of the art performance, both in terms of word error rate (WER) (As low as 7.06% on a benchmark audio set) and execution speed.

The Deepspeech system uses an RNN model to transform audio into a sequence of character-level transcriptions, which in turn are processed by a language model to help fix errors such as phonetic misspellings of words.

Customising the language model of Deepspeech can improve accuracy on domain specific tasks. In this example we evaluate AltVoice as a general purpose framework, and as such use the default language model, however in practical usage of AltVoice it may be beneficial to swap the language model to a domain specific one.

We augment the execution of the Deepspeech model by using voice activity detection (VAD) to detect pauses in speech, cutting the speech input and passing it to Deepspeech as often as possible. In doing so we can reduce the total system lag on producing output audio by not requiring the system to wait for another indicator that speech is complete. There is a potential trade off here, in that overaggressive VAD may impact the overall WER of the system. In our implementation of AltVoice we use the publicly available WebRTC [51] VAD.

5.4.1.2 Text to Speech

The Text to Speech (TTS) system turns the text and a speaker embedding, representing the new identity of the voice, into audio data.

Our overall multi-speaker TTS model is based on that of [68], which utilises a speaker encoder network to provide an embedding of a voice to the synthesizer, along with the text to be produced. The sequence of mel-spectrograms outputted by this is then passed to a vocoder, which produces output audio

In our implementation we use the open source Mozilla TTS implementation of Multi-speaker TTS[2]. From this library we use Tacotron 2 [129] for our spectrogram prediction network and use a fullband MelGAN [169] as our vocoder. The Tacotron 2 model is trained on the VCTK dataset [157], with the vocoder trained on the

---

[2] https://github.com/mozilla/TTS

LibriTTS dataset [172]. For training the Tacotron2 model a GE2E model trained on LibriSpeech dataset is used [111] as the speaker encoder network.

The choice of both of these components has a large impact on both the quality of the audio produced and the diversity of the voices produced. The Tacotron 2 model achieves MOS that are close to 4.526 vs 4.582 for ground truth speech, demonstrating its ability to create natural sounding speech. However, recent work has demonstrated that the datasets used in training impact the perceived quality of speech (via MOS testing), as well as the signal rate the speech was recorded at [31].

We perform MOS tests on our audio in section 5.5.3, to verify the performance of our system. This is especially important as previous examination of naturalness for fictitious speakers has been limited, with [68] demonstrating the technique on a set of 10 speakers generated using the random technique, which we also evaluate.

5.4.1.3 Voice Identity Generator

The Voice Identity Generator (VIG) provides the embedding to the TTS system which defines the output voice. Limited prior work exists on Voice Identity Generation for TTS, with demo examples of multi-speaker TTS just generating vectors using random numbers and then normalising the output vector.

In the voice privacy challenge several solutions, including the baseline, contained an element for voice identity generation. The work of Fang et al [43] examined potential techniques for voice identity generation within their x-vector anonymisation technique, used as the baseline for the voice privacy challenge, based on averaging embeddings from a subset of a pool of users, either using a random selection or (furthest) distance based measures. In Chapter 4 we proposed using a combination of a PCA coupled with a GMM to generate new embeddings which mirror the distribution of those found in the real world.

For our voice identity generator we examine six possible techniques:

Random Generation. Sample a value for each embedding feature from a normal distribution of mean 0 and standard deviation 1. The final new identity embedding is then normalised. This technique is based on that of Jia et al. [68] to create fictitious voices.

Random Generation in PCA Space. Conceptually similar to Random Generation, but instead of performing the random sampling in the embedding space it is conducted in a Principal Component Analysis (PCA) space. We fit the PCA on training data from applying the GE2E extractor used in the training of the TTS system to the VoxCeleb 1 and 2 Development Datasets. We create one PCA transform for each gender and fit the PCA so that it captures 95% of the variance in the data. For each component of the PCA space, we then determine the mean and standard deviation. When generating a new identity, we then sample from a normal distribution with the given mean and standard deviation for each component in PCA space, before performing an inverse PCA transformation to give an embedding in the original embedding space.

Mean Pool Subset. We follow the technique proposed in [43] for anonymised voice conversion in the X-vector space. We use a pool of all of extracted embeddings of the VoxCeleb 1 and 2 Development sets, averaging ten embeddings taken from the pool to give us our final new identity embedding. We create a separate pool for each gender.

PCA + Gaussian Mixture Model. For this method we re-use the technique for generating identities that we developed in Chapter 4, where new identities in x-vector space where generated for subsequent re-synthesis. We use a separate GMM for each gender, fit the PCA on 95% of the variance and use 20 components for the Gaussian Mixture Models fitted in the PCA space. We use the VoxCeleb 1 and 2 Development sets for training the PCA and GMM.

We evaluate each of these four techniques when examining the diversity of voices created by the AltVoice system, as it is the generation strategy, coupled with any biases introduced by the TTS which produce the overall voice identity and will be responsible for the diversity of the produce private voices.

Pool Selection. Using the pool of embeddings extracted from the VoxCeleb 1 and 2 Development sets, we select a single identity to use as the new anonymous identity for the user. In an optimal TTS system, this would be the same as cloning an existing voice from the pool. We use a different pool for each gender.

Training Selection. This is similar in concept to the pool section method, but the embedding is taken from the set of user embeddings used in training the TTS model. As the model has been trained on these embeddings, we would expect the TTS system to be able to produce higher quality audio for these embeddings.

5.4.2 Usage in Practice

In practice AltVoice would be implemented as a ‘Dialer’ application that allows a user to make phone calls.

For all of the identity generation methods a secret user-known seed is used to derive the embedding. This seed is used to instantiate a random number generator, which is then used either directly, for sampling from models, or selecting from sets of existing users. The application could manage the storage of these seeds (and thus the associated voices). By default it would generate a new identity for each different service that is contacted, but it would also be possible to configure the application to change identity when required by the user, thus changing the voice for different contexts with the same (outbound) phone number, or to maintain the same identity across multiple calls to different service providers. In cases where multiple voices are used for a single service, it would be necessary for the user to press a button to change the voice.

Finally in practice a user may not wish to provide input to the system as voice and could instead provide text input direct to the TTS component of the system. This would reduce the complexity of the system and yield higher accuracy as mistakes could no longer occur in the STT component.

5.5 Experimental Evaluation and Results

In this section we evaluate AltVoice, focusing on the requirements specified in Section 5.3. We first analyse the diversity of anonymized voices, followed by the resilience of AltVoice to attacks and then look at how AltVoice-generated utterances maintain word content.

5.5.1 Diversity of Anonymized Voices

The diversity of anonymized voices is crucial when considering the authentication system compromise threat model: too little diversity and users will have less security than normal voices as attackers would be able to authenticate with a random voice. Poor diversity will lead to worse performance in an authentication system, failing to meet R5.

Experiment Setup. To test voice diversity, we use a state-of-the-art text-independent speaker recognition method based upon the GE2E system proposed in [160]. GE2E extracts multi-dimensional identity embeddings from voices: we expect that pairs of identity embeddings obtained from AltVoice-generated voices are as dissimilar as pairs of identity embeddings obtained from natural (or *organic*) voices. For this analysis we bypass the STT part of AltVoice and we generate AltVoice voices directly from text files, using sentences from the Mozilla Common Voice dataset in the English language [101]. For each of the six identity generation methods introduced in Section 5.4, we produce 500 voice identities and generate 30 spoken sentences for each of these. For each identity, we use the identity embeddings extracted from 10 out

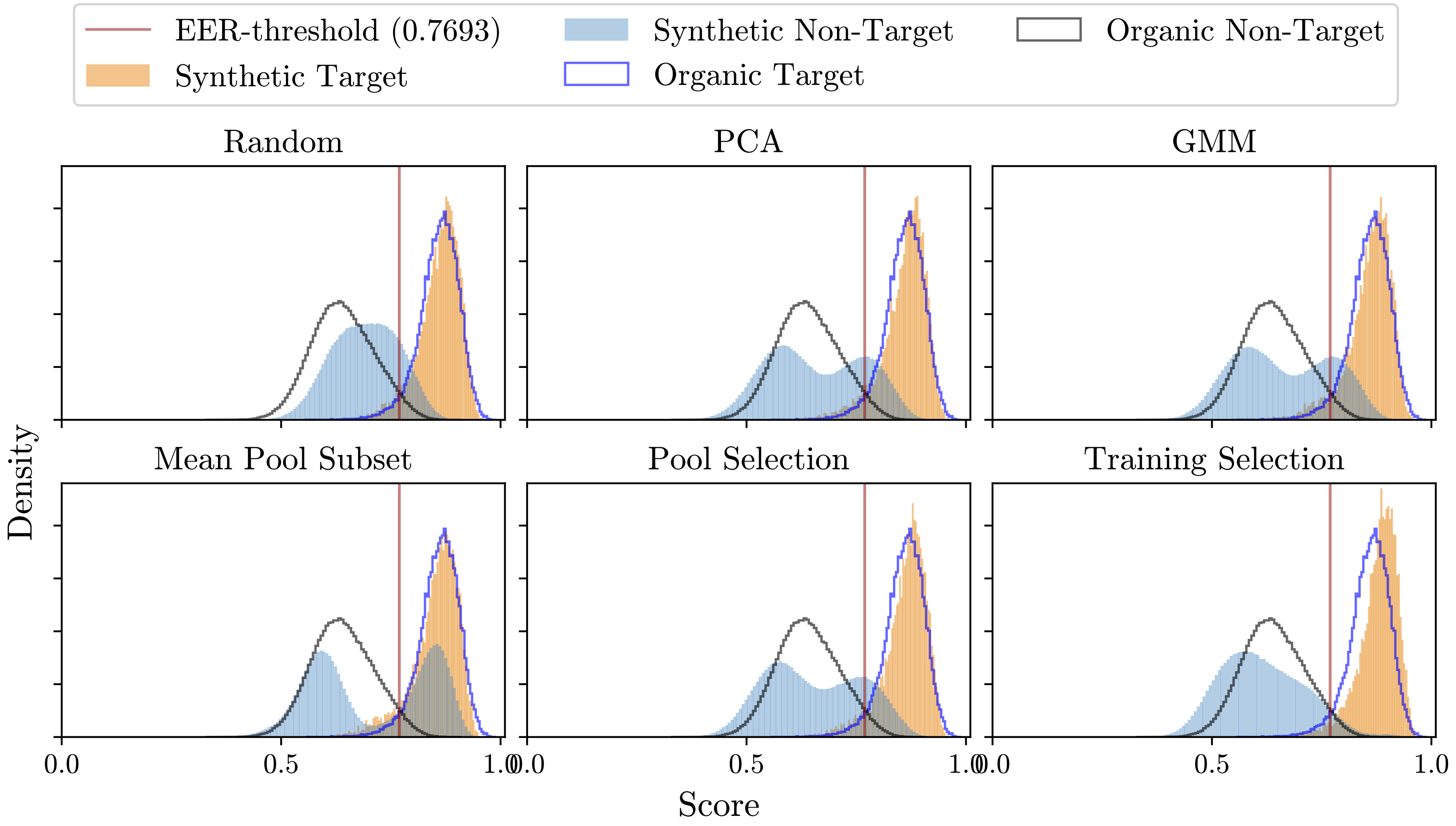


Figure 5.3: Distribution of pairwise cosine similarity scores between naturally occurring voices (organic) and voices produced by AltVoice for each of the identity generation methods. Target means that the pair compares a identity template with an utterance belonging to the template owner, non-target means the opposite. The EER-threshold is computed using naturally occurring voices.

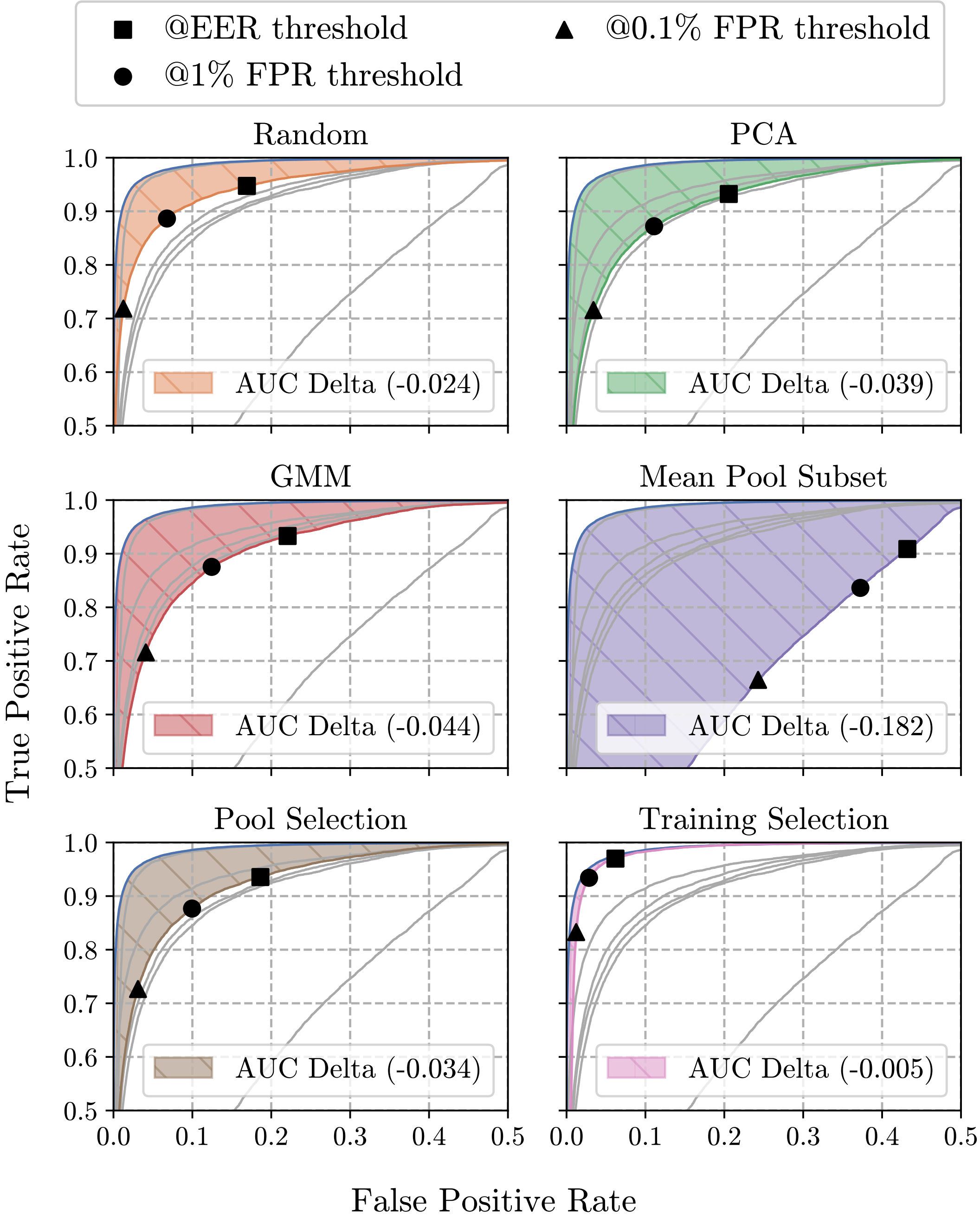


Figure 5.4: ROC Curves obtained by comparing target and non-target vector identity pairs, for each identity generation method. The solid blue line reports the performance among naturally occurring voices, which leads to an equal error rate (EER) of 3.84%. Markers on the curves report the performance trade-off for various ways of selecting a test-time recognition threshold based on the performance among natural voices: either at EER level, at 1% FPR or at 0.1% FPR. Shaded areas report the method AUC difference compared to the AUC obtained using natural voices. In this chart a true positive is two audio samples with the same target identity matching, whereas a false positive is two samples with different identities matching.

of the 30 sentences to enroll an identity template (by taking the average of them), we reserve the other 20 utterances to be used as positive and negative trials. We compare the identity templates with the remaining 20×500 utterances, by selecting sets of *target* and *non-target* pairs (i.e. target when the utterance belongs to the same identity as the template, non-target otherwise) and storing their distance scores using cosine similarity [160]. Finally, we use the distribution of cosine similarities to compare natural and AltVoice-generated voice diversity. We use the Voxceleb 1 and 2 test datasets [106] for the natural voices. This dataset contains audio data that is scraped from videos of celebrities speaking, containing over 1 million utterances from more than 6000 celebrities.

Results - Similarity Distributions. Figure 5.3 shows the distributions of distance scores for the six identity generation methods. The distribution of target pairs i.e. same identity, are similar to those for naturally occurring voices. We see that these are shifted slightly right than the original audio, an effect particularly noticeable for the training identity generation method, suggesting utterances from the same identity have less variation for AltVoice-generated voices than for naturally occurring voices.

Comparing non-target pairs, we see a disparity between the naturally occurring distribution and those for the identity generation methods. For the PCA, GMM and Mean Pool Subset models we see two peaks, corresponding to each gender. This suggests that these methods create voices for each of the genders that are more similar to one another than naturally occurring voices of that gender. For the Random method we see the distribution is shifted to the right, implying that the voices are more similar than naturally occurring ones.

The pool selection method is dual peaked, implying that the underlying TTS network is responsible for some of the increased similarity between voices of the same gender, as opposed to this being a consequence of just the identity generation method. The training selection method shows a peak to the left of the baseline, but with a wider distribution. This suggests some overfitting may be occurring, due the leftward shift of the peak. Similarly it may be that voices of the same gender are more similar, causing the long tailed distribution.

Results - ROC curves. Figure 5.4 report receiver operating characteristic curves obtained with the pairwise similarity scores. Figure 5.4 shows that the performance for each of the methods i.e. the area under the curve (AUC), is worse than the AUC computed among naturally occurring voices (blue solid line in Figure 5.4); the gap between the natural AUC and the AltVoice-generated AUC is reported in each plot for each method. The voices in Training methods performs best, with a small difference of 0.005 AUC from the result obtained with naturally occurring voices. It should be noted however that we are only limited to 96 voices used in training, so in this case there is a 1/96 chance a voice generated by a user would be the same as another voice generated by another use.

5.5.2 Threat Model Attacks

Here, we study the performance of AltVoice against two attacks: a privacy compromise attack and an authentication compromise attack.
Audio Used. To evaluate the attacks, we re-use the same audio generated in the previous section: we generate 500 voice identities for each of our proposed identity generation methods and for each private voice we create 30 spoken utterances. As before, we use source sentences obtained from the Mozilla Common Voice dataset in the English language. We use the speakers in the VoxCeleb 1 and 2 test datasets as naturally occurring voices [30, 106].

5.5.2.1 Privacy Compromise

Experiment Setup. To conduct the privacy compromise attack, the attacker has obtained a sample of a private voice and wishes to identify the speaker the voice belongs to, matching against known candidate voices.

To simulate the adversary's knowledge of a voice dataset, we select a set of 20 speakers from VoxCeleb test set, for each of them we compute an identity template available to the adversary. Then, we use the same 20 speakers with AltVoice, generating a private voice for each speaker and we produce one utterance for each of them: the adversary needs to match private voices with speaker identities. To do the matching, the adversary computes the cosine similarity between the utterance embedding and the embedding of the user template, selecting the most likely speaker as the nearest voice. Cosine similarity is used as the similarity measure when training the GE2E network and is therefore the best method for an attacker to determine the similarity of two embeddings.

We conduct 100 rounds of the experiment for each identity generation method. Throughout the attack, the adversary only considers voices that are the same gender as the original voice (in the case of the random identity generation method gender is ignored).
Results. We report the results of the attack in Table 5.1. We find that conducting the privacy compromise attack on normal voices (without the AltVoice protection), the

	Generation Method	Success Rate (%)	95% CI (pm)
	Baseline	97.585	0.002
AltVoice	GMM	10.355	0.618
	Random	7.713	0.557
	PCA	8.445	0.460
	Mean Pool Subset	8.826	0.370
	Pool Selection	9.063	0.335
	Training Selection	9.276	0.310

Table 5.1: Privacy attack results. The table shows how the attacker can successfully de-anonymise a non-protected victim (baseline) with high success, while when using AltVoice with the various identity generation methods the attack success rate decreases.

adversary identifies the speaker with a success rate of 97.6%. This is as expected, as the encoder network used is highly accurate, meaning it is almost always the case that the nearest speaker to the sample is the correct one. Applying AltVoice, the attack success rates are reduced significantly. For the methods with gender, the mean success rate is just below 10%, which is the expected outcome from guessing randomly out of 20 identities to choose from. The reason for mean success rate being just below 10% could be due to the synthetic voices not being as spread as naturally occurring ones, as seen in the experiments in Section 5.5.1, meaning that the same incorrect guesses are made for large parts of the voice spectrum. The random generation method achieves a 7% performance, slightly higher than the 5% from purely random guessing, again likely caused by the random voices having a different distribution than normal voices.

5.5.2.2 Authentication Compromise

Experiment Setup.

To conduct the authentication compromise setup, we consider two scenarios, one in which the victim has enrolled with a VPS with their natural voice to act as a baseline and another where they have used a private AltVoice voice. We re-use the same speaker verification system used in Section 5.5.1. We follow a similar procedure to evaluate both scenarios:

1. Select a victim from the set of possible speakers (Voxceleb 1 and 2 test datasets).
2. Create the victim's natural template from 10 utterances, or create a private voice for the victim and a template for this with 10 AltVoice-generated utterances.

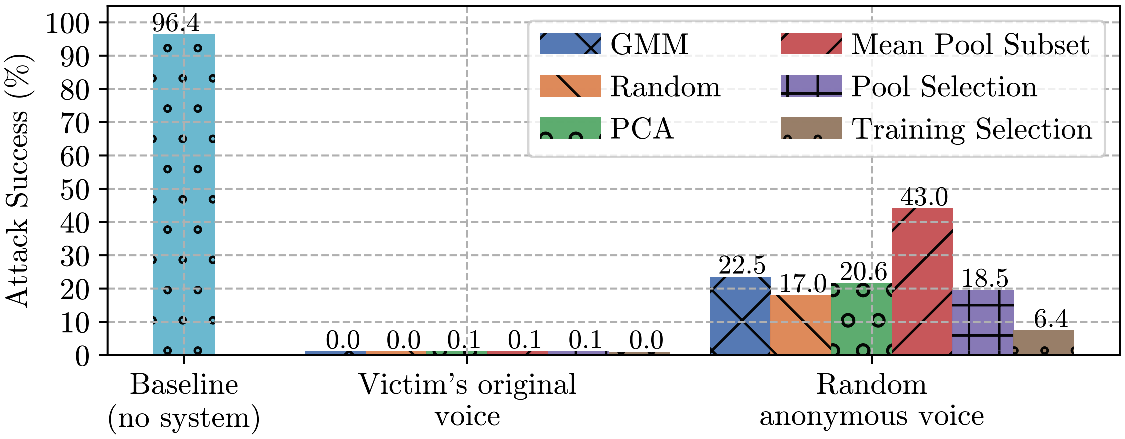


Figure 5.5: Adversary success rates under our authentication attack. Baseline refers to the attack success rate when no AltVoice is in place (voices are unprotected). Victim's original voice refers to the attack success rate with AltVoice in place and the adversary using the victim's original voice for impersonation. Random anonymous voice refers to the attack success rate with AltVoice in place and the adversary using a randomly generated AltVoice voice for impersonation.

3. Adversary presents an utterance in an attempt to access the system, using one of two strategies: (i) Use a natural utterance belonging to the victim or (ii) Use a AltVoice-generated utterance with the same identity generation as the victim (but different secret seed).

We treat a successful attack as one where the similarity score exceeds the threshold value at EER using the Voxceleb 1 and 2 test dataset, i.e. an attack is successful when the adversary successfully impersonates the user. We conduct 10000 trials for each of the scenarios.

Results. Figure 5.5 shows a comparison of the results for the baseline and both of the attack strategies employed by the adversary. In the Baseline scenario, we find that an attacker is successful 96.4% of the time. This is intuitively the case, as if an attacker has perfect copies of a voice, they should be able to impersonate that user with a success rate the same as the system's true positive rate.

Testing the victim's original voice against a private rate we get a success rate of 0.1% for all identity generation methods. This shows that the private voices generated by our system are only related to the original voice by their textual content and thus the identity embeddings are fully independent.

With an attacker trying random anonymous voice we see varied results across the attack techniques. The Mean Pool Subset method performs worst, with the attacker

	Identity Generation Method	MOS	CI (±, 95%)
	Baseline	4.230	0.054
AltVoice	Random	3.015	0.067
	GMM	2.514	0.059
	PCA	2.548	0.058
	Mean Pool Subset	2.438	0.054
	Pool Selection	2.495	0.052
	Training Selection	2.607	0.062

Table 5.2: Mean Opinion Scores for the baseline audio and each identity generation scheme. All AltVoice-generated audio performs worse than the Baseline.

selecting a random voice that is closer than the threshold 43.0% of the time. Selecting a voice with the Training identity generation performs strongest, with the attacker randomly selecting a colliding voice 6.4% of the time.

5.5.3 Perception of Audio Quality

While the (perceived) quality of generated audio is not important for automated systems (beyond the WER), it is crucial for communicating with humans. If the generated voice sounds obviously synthetic, the human may terminate the call or refuse to engage in sensitive transactions (e.g. in phone banking). Similarly to previous works, we perform Mean Opinion Score (MOS) tests to measure the quality of the voices that are produced by our technique.

Experimental Setup. In order to obtain MOS, we use a crowdsourcing approach on Amazon Mechanical Turk. As in Chapter 4, we follow the ITU-T recommendation P.808 [67] for this.

Following the completion of the qualification, setup, and training phases, we present a total of 700 audio samples – 100 baseline clean audio files from the LibriSpeech dataset and the same set of files fed through the AltVoice system with a new private identity randomly generated for each file. These samples are presented in sets of 12 and workers are compensated for each set they complete. Each set includes samples from all of the seven conditions (Baseline + 6 generation methods) in order to prevent workers who don't rate all 700 samples from biasing the results. In line with P.808 recommendations, we incentivise workers to complete at least 50% of total samples. Overall, each sample is rated by at least 8 distinct workers. We received ethical approval from our institution to perform this experiment, reference SSD/CUREC1A CS_C1A_21_010.

Results. Our MOS results are presented in Table 5.2. All AltVoice generated audio achieves worse MOS scores than the Baseline audio from organic speakers. Of the identity generation techniques Random performs best with an MOS of 3.015. Our MOS results are in contrast with the higher (>4) MOS scores reported in the TTS paper [68], suggesting that such systems will need further improvement before they can be integrated out-of-the-box into more complex pipelines such as AltVoice. Scores of 2.5 on a MOS scale mean that in most cases it would be obvious that the audio is not from a natural (human) speaker.

5.5.4 Word Error Rates

We evaluate the word error rates of AltVoice when applied in an end-to-end manner, to determine how much word information is lost. There are two potential points at which errors may be introduced into the system: (i) when initially performing speech to text and (ii) when synthesising speech from text. The model used for performing STT has a WER rate of 7.06% on clean test data, giving this as an empirical upper bound on performance.

Experiment Setup. We apply AltVoice to a set of 25 randomly selected audio files from the LibriSpeech test-clean dataset, transforming each file to the same private identity. This is repeated 100 times, for each of the 6 proposed identity generation methods.

Each generated audio file is transcribed using both the Google Cloud Speech API and Deepspeech. We also transcribe the original 25 files, to give a baseline for comparison. We evaluate the differences in this transcribed audio using Word Error Rate (WER) and Word Information Lost (WIL). Often only WER is used, however WIL is preferred for speech use, as it measures the proportion of word information communicated [99]. Note that by definition WIL can never be higher than WER for a given set of sentences.

Results. The results for each of the techniques can be seen in Table 5.3. We see that all the identity generation methods have a large degradation from the baseline in both WER and WIL. This degradation would be discernible to listeners and could lead to sentences with lost meaning or confusing words embedded in them.

There is some difference in both of the metrics for the various techniques. Most notably we see that performance is worst for the random identity generation method. This could be a result of there being some relationships between the values in the identity embedding, which are lost when all values are set randomly, resulting in identities that the system struggles to produce audio for. Across the other methods

System	Deepspeech		Google Cloud	
Metric	WER [CI]	WIL [CI]	WER [CI]	WIL [CI]
Baseline	9.74%	16.82%	9.74%	16.82%
Random	42.41% [40.84, 43.98]	57.54% [55.84, 59.24]	47.92% [46.49, 49.35]	67.11% [65.64, 68.59]
GMM	24.48% [23.50, 25.47]	36.65% [35.38, 37.92]	30.27% [29.04, 31.51]	46.46% [44.86, 48.05]
PCA	25.42% [24.48, 26.37]	37.88% [36.68, 39.08]	31.57% [30.37, 32.77]	48.11% [46.56, 49.66]
Mean Pool Subset	22.93% [22.15, 23.70]	34.72% [33.70, 35.74]	28.67% [27.58, 29.76]	44.34% [42.89, 45.80]
Pool Selection	24.33% [23.42, 25.23]	36.52% [35.34, 37.69]	29.88% [28.68, 31.08]	46.01% [44.41, 47.61]
Training Selection	28.99% [27.86, 30.12]	42.47% [41.11, 43.82]	35.80% [34.28, 37.31]	53.30% [51.45, 55.14]

Table 5.3: ASR Recognition Rates for 1,000 Audio files generated by our system from the LibriSpeech test-clean dataset. Transcriptions are produced with a local version of Deepspeech 0.9.3 and on Google Cloud Speech-to-Text Service in May 2021.

we see broadly similar results, with the Mean Pool Subset method performing best. This could be because averaging many embeddings results in an embedding central in the overall space, making it fairly neutral and easier to synthesise quality audio for.

Interestingly the Training Selection method performs worse than the Pool Selection method, despite the training identities having been seen before.

5.6 Discussion

In this section we discuss the identity generation performance, system limitations, and the ways in which we expect the individual components to improve over time, highlighting directions for future work. We also discuss the general implications of the existence of this system and in particular new considerations for designers of remote voice based systems as a result of the existence of private voices.

5.6.1 Identity Generation Performance

The results from each of the experiments paint a mixed picture for the identity generation techniques we evaluate, with no technique being a strong performer across all categories.

Generally the techniques that create structured (i.e not purely random) synthetic embeddings from existing models performed poorly in the diversity and attack sections, with the homogeneity of the embeddings they create meaning that an adversary can generate a fairly similar voice with few attempts. The random generation scheme, produces more diverse voices, with better MOS performance, but at the expense of much higher WER and WIL scores.

Both the Pool Selection and Training Selection identity generation methods produce more diverse voices and perform better under both attack scenarios. The training selection method in particular performs very strongly under both attack threat models, as well as performing second best for MOS, although has a slightly higher performance cost for WER and WIL than the synthetic techniques.

Going forward further work is needed on generating good synthetic identities. Alternatively further improvements to the models performance on the voices seen in training could be fruitful, as given a sufficiently large set of training voices selecting from this achieves the security goals of the system.

5.6.2 Limitations of Proposed System

Through the experiments we evaluated in this chapter, we demonstrated that improvements to identity generation methods, coupled with improvements to multi speaker TTS are needed to ensure that the space of producible voices is as wide as natural voices.

We expect that these improvements are likely to come with time, as computational power increases and increasingly advanced neural models for multi speaker TTS are developed. Comparing progress from just a few years ago, when high quality TTS required large databases of audio for that individual, to now where a few seconds is all that is required to clone a voice, it does not seem unlikely that techniques will continue to improve in this area.

Further progress is also needed in audio intelligibility and understanding. The best identity generation methods caused an increase in WER of approximately 15% over the baseline and MOS scores were degraded from 4.23 to around 2.5 (2 is rated as poor and 3 as fair). Again this is an area that has been rapidly improving in recent years with neural systems.

Performance improvements in this area are likely to come from both improvements in the STT and the TTS systems.

Furthermore in this work we only used the default Deepspeech system for STT, and adaption of this to specific tasks should improve the WER. It's also worth noting that the dataset used for our WER analysis is from applying the system in an end to end manner on the LibriSpeech dataset, which was produced by people reading books aloud. These books were from Project Gutenberg and as such are generally over 95 years old [53], meaning the text in them is likely to be different from the kind of speech used today.

For practical usage of the system there is also the difficulty of input lag. By using VAD this lag can be minimised to the length of speaking without a pause, but it can not be eliminated. Similarly there is also some lag introduced by the processing pipeline and although each of the models operates with a real-time factor of below 1, we have not experimentally analysed the effects of input lag on the overall usability of the system, which we leave to future work.

Finally the system is limited due to the deterministic nature of the audio that it produces – given identical text and identity embedding, the exact same audio will be produced. In this work we only consider text-independent speaker recognition, but text-dependant is often used, where a specific phrase must be spoken. However, with the AltVoice system the same phrase will always be identical, which may lead to systems preventing access due to detecting an overly similar audio sample i.e. a replay attack. Further work is needed to evaluate methods to alleviate this, such as by applying a small perturbation to the identity embedding to yield slightly different audio.

5.6.3 Limitations of MOS

In our experiments (as well as in experiments in Chapter 4) we utilise MOS to assess the quality of the audio generated by our system. As a metric MOS utilises scores provided by listeners, who assess the quality of the speech they hear on a scale of 1 to 5. MOS has become a de-facto method of measuring the quality of audio, but it also has weaknesses that limit the ability to compare measures across experiments or settings. In particular the specific pool of listeners used could impact the scores, meaning that results obtained from one pool at one point in time may not be comparable with those obtained from another pool.

Furthermore we utilise Mechanical Turk for our pool of listeners. This is highly attractive to researchers, as it simplifies many parts of the process of conducting such a study, such as recruitment and payment of participants. However the participants are not necessarily audio experts and perform the task in an unsupervised manner, meaning researchers rely on the participants performing the task correctly and following all steps. We employ standard quality control measures as per the ITU-T recommendation [67], such as trapping questions and calibration audio in order to prevent instances of obviously incorrect study participation. However these additions can not force workers to judge audio properly and we can not be certain that they are rating the audio to the best of their ability. These concerns, coupled with the Mechanical Turk workers desire to operate as quickly as possible, mean that MOS scores are better interpreted as a comparison between methods and not used for comparing techniques across different studies.

In the future alternative metrics, or improvements to MOS, would benefit the comparability of audio naturalness between studies. Potential avenues to achieving this could be through developing algorithms or machine learning methods to assess audio quality or through improving the remote study methods further to improve rating quality. Many researchers conducting these experiments already use the same framework for MOS testing, which could be further extended to include a qualification for participants to be able to participate in studies. The framework could then only publish participation requests to qualified users, ensuring a more consistent and higher quality pool of participants across experiments. An authority figure would likely be required to maintain the list of approved participants and ensure new participants are added once they have met the requirements.

5.6.4 Additional Secrets

One of the principle attractions of voice based authentication is that the system user is no longer required to possess an explicit secret to access the service, with this secret instead being derived from the users voice. Our proposed system reneges on this, once again requiring a user-known secret in order to create the correct voice identity. Whilst this may appear to be a step back, in a situation where the original voice trait – and thus the secret within it – has been compromised, it is necessary to introduce a new secret for voice authentication to continue to be used. Unlike many biometrics, voice data is actively shared in many contexts and as such the likelihood of a voice trait being exposed is high for most individuals. AltVoice therefore allows such individuals to use remote voice based authentication systems with reduced fear of being the victim of an impersonation attack.

5.6.5 Longer Term Implications

The existence of a system such as AltVoice shifts the requirements of remote based speech systems. In particular, the ability to change your voice will require remote speech systems to offer the ability to re-enrol if the voice for that remote system is compromised (*revocability*, see [107]). This would bring remote voice services in line with traditional text based password systems, where the password can later be changed by the user.

The compatibility of this system with existing remote systems is also likely to be a point of tension. In particular previous works have investigated being able to detect synthetically generated audio via liveness detection or spoofing detection e.g. [4, 14, 74, 144, 165, 176]. Such works include the ASVSpoof challenge, which has undergone several iterations, with systems achieving EERs below 1% in the 2019 edition [144]. As such audio produced using the proposed system may at this stage be caught by such a system.

However, for a user who has already had their voice trait stolen and perhaps audio of them uttering specific keyphrases needed to access a given system, the ability to change their voice would be a benefit to them. The alternative is relying on an adversary to not generate a convincing enough fake or to not have the correct phrase to play. In particular the ASVSpoof challenge does not contain samples where an attacker has logical access to the system and recordings of the original speaker, as would be the case with a remote replay attack (such as playing a sample directly down a phone). Such samples should by definition be the same as regular speech down a phone and thus would not be distinguishable from other samples. AltVoice protects against an attacker conducting this attack, as the attacker will not be able to replay a sample of the victim's voice.

Thus a tradeoff arises, between synthetic voices being allowed and people having the benefits of this, such as no voice reuse and cancelability against systems attempting to prevent synthetic voice usage, leaving people potentially vulnerable to adversaries who can bypass this. One possible solution is to enable spoofing detection checks by default, but allow users to disable these if they wish to use alternate voice systems.

5.7 Conclusion

In this chapter we propose AltVoice, a system enabling users of remote voice system to both protect their privacy and use such systems without having to rely on the confidentiality of their possibly exposed voice trait. The system reduces spoken audio to its textual content, before re-synthesising it with a newly generated identity embedding, resulting in audio that appears to be spoken by a different voice.

We examine several identity generation methods within the proposed system, evaluating them on the diversity of voice created, word error rates, and perceived audio quality. We also examine them under two realistic threat models, for a privacy compromise attack and an authentication compromise attack. Our results demonstrate that the new audio maintains its identity and resists attacks under both threat models better than the status quo – where a user's voice trait has already been exposed.

By outlining the system's trade-offs, we highlight how improvements are needed for such a system. In particular we find that re-using state-of-the-art TTS, STT and speaker recognition components in AltVoice leads to private voices with relatively high word error rates and limited perceived audio quality. We expect many improvements to occur over time as the components mature and deep neural network research continues to provide performance gains.

Our system also raises new questions for the designers of remote voice systems, in particular as to how their systems may need rethinking to support the changing of voices over time, as well as how they can support systems such as the one proposed to facilitate better privacy for users and better security for users with exposed voice traits.

Chapter 6

Moving Lab Experiments Online

Contents

In Chapter 3, when developing the attack against speaker recognition systems, one key component of the research was collecting the audio database required to perform the attack. Despite being a relatively small collection of audio, with 20 participants reading audio from a script for a duration of about 30 minutes in total, the logistical effort and time involved was large. For each participant approximately an hour was required for setting up and facilitating the recording. Further time was also required to publish the advert across various channels, screen participants and schedule their collection time. Furthermore there is also overhead in the lost time between participant sessions.

Separately to this, recruiting for in person studies on campus leads to a biased sample, with participants (generally) being members of the University and young compared to the population at large, as well as further biases that may be present within the student body.

This led to conversations with collaborators in the research group on how this could be improved, leading to the development of a system for collecting (some) biometric datasets remotely. In this Chapter we discuss this development process, the experiments, and the analysis we performed on the data collection method, before providing useful advice for those seeking to collect datasets in this way in the future.

6.1 Introduction

When working with biometrics, including voice, it is often necessary to collect datasets in order to conduct research. Historically these datasets have generally been collected in person, such as with collecting the voice dataset in Chapter 3. This data collection was particularly time intensive, with each of the 20 participants being required to read from a series of scripts in a quiet space, whilst accompanied by myself in order to record their audio data, taking up to an hour per participant including the informed consent process and all recording sessions.

As such we set out to find an alternative method to collect longitudinal datasets, that was less time intensive for researchers, whilst also making it easier to conduct studies with measurements conducted over longer time periods.

Crowdsourcing has been embraced in parts of academia and widely used in research tasks, being deployed to either collect data about workers (e.g. demographic surveys [113]), to collect data about some stimulus (e.g. image tagging [84]) or to collect data about workers reaction to stimuli (e.g. perceived audio quality [131]). Commonly, these tasks are very short in duration and workers therefore attempt to make up for the small payment per task with volume [57]. In Chapters 4 and 5 we used crowdsourcing with the Amazon Mechanical Turk platform to conduct experiments where participants were asked to score the naturalness of audio we generated, a standard method for conducting these experiments. However there is nothing to prevent them from being used for other tasks and in particular to be used for collecting longitudinal datasets.

In this chapter we develop a system for collecting such datasets. We first developed a prototype of the system for collecting smartphone camera Photoplethysmography (PPG) data, before subsequently deploying an enhanced version of the system for collecting a large touch dynamics dataset. We evaluate the successes of this system, and propose some design recommendations for the development of similar systems in the future. Finally we discuss the nature of informed consent in such studies and the impacts of this on the applicability of this technique to future experiments.

6.2 Crowdsourcing Overview

Crowdsourcing uses a crowd of people to achieve some goal or task and in particular small repeatable tasks. The most popular platform for doing this is Amazon Mechanical Turk (MTurk), which facilitates interaction between the requester, who provides the task to be done and the worker, who completes a Human Intelligence Task (HIT). Workers on MTurk are paid for each HIT they complete by the requester, according to the payment schedule that they set, with no minimum wage enforced by the platform. Workers are free to select the tasks they perform, as long as they meet any requirements set by the requester, such as a minimum previous HIT acceptance rate, or completion of a qualification task.

6.2.1 Typical MTurk Usage

Several studies of both the MTurk platform and MTurk workers themselves have been conducted to gain insight into how the MTurk platform is used.

On the platform the majority of tasks completed (61%) are microtasks [62]. These tasks pay \$0.10 or less, can be completed quickly and are generally repetitive. Examples of such tasks include image classification [78] or relevance judgements [7].

In [62] tasks were observed for a week, with 37% being imaged content tasks, 26% audio transcription, 13% content classification, 7% information collection and 13% completing surveys. Our longitudinal studies do not fit into any of these categories, and is not a microtask, instead requiring longer continued engagement of specific workers. As such the task is likely to be novel to workers, leading to an increased chance of workers self-selecting away from our task [95].

6.2.2 Longitudinal Studies on MTurk

Previous longitudinal studies have been conducted on MTurk, with these generally being repeat surveys at pre-defined time intervals. The TurkPrime platform supports longitudinal surveys, notifying workers of new surveys with emails [86]. Using an email based notification strategy such as this yields participant re-response rates of 75% after 2 months, decreasing to 47% after 13 months, for a survey based task [34]. Previous works have used notifications to encourage continuous participation in experiments and in particular in experiments with high engagement frequency [12,135]. Notifications can also help establish habitual interaction with smartphones [110] and can be used to influence how often users access an app, in particular when offering a prize or electronic coupon [66].

Longitudinal studies have also been conducted to investigate information about workers over a period of time, such as that of Weinshel et al., who used a browser extension to monitor worker's exposure to website trackers over a period of a week, book-ending their study with surveys which prompted the install of a browser extension and a review of the data the extension had observed [162]. The key difference with our type of study is that interaction once the browser is installed with the study is passive until the completion of the post study survey.

In contrast to previous longitudinal studies, we generally wish to collect data from participants daily and with a desire for reasonable certainty in participants continuing to complete the task each day. Thus our tasks are both time critical – unlike most longitudinal studies – and require the same workers participating, unlike most micro task studies. As such, we design and implement our custom solution to conduct these high frequency longitudinal studies, as described later in Section 6.4.2.

For micro-tasks that are repeated over long periods of time it has been demonstrated that task accuracy and success are both stable long term, with workers self-selecting out of tasks that they believe they perform poorly on [59]. As such we can be confident that worker quality is unlikely to decline throughout the duration of our experiment, particularly as fatigue conditions are unlikely to accumulate significantly with the task only being completed once per day.

6.2.3 Worker Recruitment & Retention

For many tasks on MTurk it is useful to retain workers so that a batch of tasks can be completed in a timely manner, with individual workers completing multiple tasks of the same type. Typically the number of Human Intelligence Tasks (HITs) completed from a batch by the same worker follows a long tail distribution, with most workers completing few HITs, with a few workers completing most of the work, broadly aligning with the Pareto rule [37, 59]. For our studies retention is more than just useful: it is essential. The experiments we wish to conduct with the biometric datasets we obtain require a large volume of samples so that we can train models to recognise them accurately. Furthermore we are also interested in the *time stability* of the features. Considering PPG as a biometric, it is conceivable that ten measurements taken over the course of one session may have very similar values. However, suppose these measurements are instead taken over several sessions on subsequent days. We may find that artefacts introduced by natural variation, both organic but in response to different external factors from day to day may yield different measurements. As such we wish to train our models to be robust to these natural variations over time, and to be *time stable*, which we can not do if all measurements occur in one session.

Difallah et al. investigate the impact of different pricing schemes on different types of HIT, showing that pricing scheme impacts user retention for workers completing the same HIT repeatedly, with bonus payments for volume of HITs completed yielding the highest retention rates [37]. The payment schemes studied here can not be applied directly to our problem, but they help inspire the payment schemes we investigate.

Studies on task abandonment also show that insufficient monetary reward is the strongest driver of abandoning a task [55], with workers abandoning early if the monetary reward seems poor, with payment the strongest motivating factor for workers [71, 125]. Workers also help each other using external platforms for mutual aid to flag good and bad requesters [54, 95], a further factor in ensuring that we pay workers appropriately, to ensure those using such services see our tasks as being worth completing.

The impact of rejection on workers pay has also been investigated [95], showing that workers are wary of rejected work, due to the loss of pay from wasted hours, as well as the reduced opportunities for future work due to requesters mandating that workers have a low rejection percentage to complete their tasks. As such workers select tasks to minimise the risk of rejection and impact on their pay. Similarly it has been demonstrated that longer instructions reduce task recruitment [163]. This represents a difficulty for us, as the uniqueness (relative to the bulk) of our prospective tasks, combined with our differing payment scheme may be perceived as a risk by workers and will necessitate longer instructions.

6.3 Candidate PPG Study

Our initial candidate for conducting a remote biometrics study was to collect a smartphone camera based photoplethysmogram (PPG) dataset. A PPG is an optically obtained plethsymogram, which is used to detect blood volume changes in tissue [128].

We intended to use this dataset to examine the possibility of using camera collected PPG for authentication. We were particularly interested in evaluating the time stability of the signal derived from the camera. It is known that the operation of the heart changes with time, but furthermore session artefacts, such as placement of the finger on the camera, skin temperature, or even if the participant has recently moved about, could impact the signal that we are able to obtain from the camera.

6.3.1 Capture Application

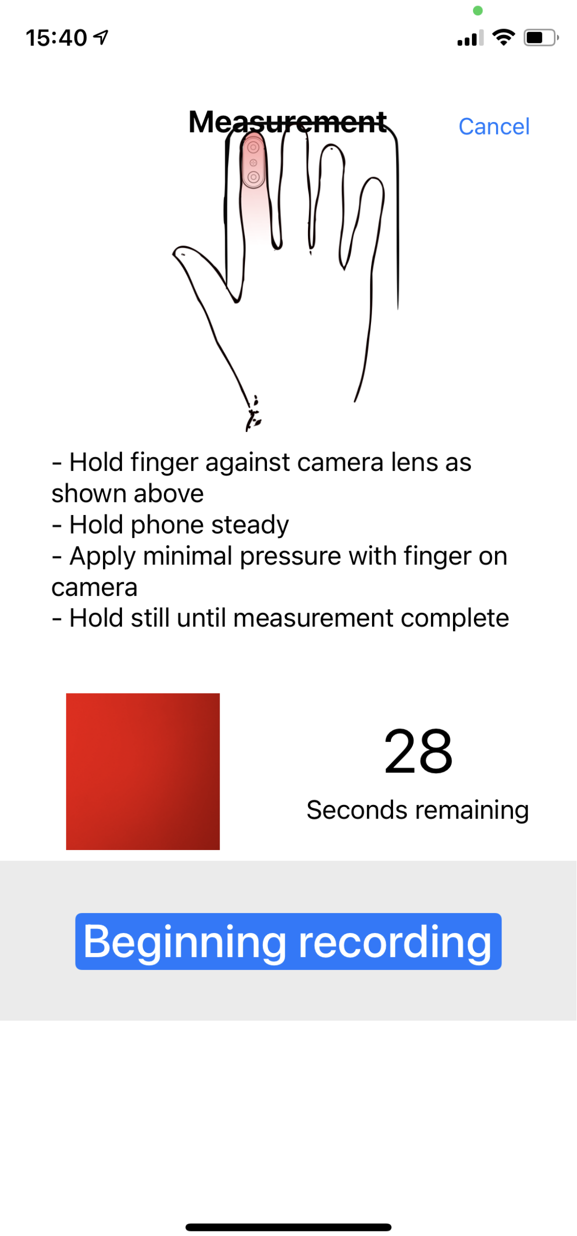
We first developed an iOS application to capture the PPG signal, a screenshot of which can be seen in Figure 6.1. The application records video at 120 frames per second from the device camera whilst shining the torch. The participant places their finger over the camera and records a 30 second long video, which captures the movement of blood within the finger. The application also monitors the accelerometer of the phone, stopping and restarting the recording if too much movement is detected. The video is then compressed and resized so that it can be uploaded to a remote server.

6.3.2 Suitability

We conducted an initial in person study with 15 participants, who each participated in 6 to 11 sessions, to both complete initial research into the biometric and to assess the suitability of the biometric for remote capture. As part of this study we verbally instructed participants on how to place their finger over the camera lens and demonstrated the use of the application. Without this verbal instruction it was not instantly clear to participants on how to perform the measurement and further improvements to the application to do this would have been necessary.

From this initial study it soon became apparent that PPG measurements collected in this manner were sensitive to the movement of participants, as well as highly sensitive to the exact placement of the finger over the camera. With the lab study we could correct this, however remotely this would be difficult to do and could potentially impact data quality.

Furthermore it was difficult to ensure that participants would enter legitimate data if using the app remotely. In order to proceed further we would need to develop



- Hold finger against camera lens as shown above
- Hold phone steady
- Apply minimal pressure with finger on camera
- Hold still until measurement complete

Figure 6.1: Screenshot of the application developed to capture a PPG using a smartphone camera. In the screenshot the application is in use, with the participants finger over the camera in the manner shown in the diagram. The red square shows a live feed of what the camera sees, so that the user is aware when the finger is fully covering the camera.

algorithms that could detect potentially malicious inputs, such as placing the phone on a red object, as well as for benign but incorrect inputs, such as when a participant temporarily removed their finger.

Ultimately we decided that the PPG biometric was not suitable for collection in a remote manner due to this combination of environmental sensitivity and difficulty in ensuring quality data. Consequently we did not proceed with remote collection of this dataset. The data collected was utilised for a smaller scale study into PPG biometrics using smartphone cameras [89].

6.3.3 Research Ethics

Before the data collection, we obtained informed consent from each of the participants. Our data collection process and experiment took place with ethical approval from the university: reference SSD/CUREC1A_CS_C1A_19_032.

6.4 Touch Dynamics Study

After finding PPG unsuitable for this kind of remote collection study, we instead looked for other biometrics that may be suitable. Touch dynamics seemed like an obvious candidate, as the experiment naturally occurs on a mobile device, the application can be designed to ensure data quality, and the biometric itself is less sensitive to external environmental changes than PPG.

As such we set out to collect a behavioural biometrics dataset with daily measurements over a period of 31 and 7 days.

Behavioural biometrics – the use of distinctive human behaviour for identification or authentication – have become a popular way to establish user identity without requiring explicit action (such as entering a PIN). On mobile devices, touch dynamics involve using distinctive characteristics of touchscreen interactions such as the start and stop coordinates of swipes, gesture speed and touch pressure [47].

Traditionally, studies on behavioural biometrics have been conducted in controlled lab environments [5, 47, 104, 124] due to the relative ease of setup and the physical presence of researchers to spot irregularities. However, an ideal study requires a high number of participants to complete a specified phone-based task repeatedly and regularly over an extended period of time. In addition, the makeup of participants should largely reflect the general population in terms of gender and, for touch dynamics, handedness. In addition, restrictions related to the COVID-19 pandemic further impede in-person experiments, with safety requirements ruling out many potential experiments for the time being. The onset of the pandemic coincided with the development of our experiment framework, providing further motivation to pursue the remote collection.

Consequently, we undertook a longitudinal study with recurring daily measurements on touch dynamics using the Amazon Mechanical Turk platform. As noted in Section 6.2 using MTurk in this way is unconventional and as far as we know studies of this kind have not been performed previously on the platform. In the following sections we discuss the design of our study and how we optimised the design to retain users throughout.

6.4.1 Touch Study Design

We conducted our remote participant study with the purpose of creating a large datasets for behavioural touch based biometrics on smart phone devices. The study was approved by the University ethical review process with reference SSD/CUREC1A CS_C1A_19_013.

In using crowdsourced work to conduct our study, we had specific aims that we hoped to achieve throughout our experiment:

A1: Collect measurements from workers daily, with few skipped days.

A2: Engage workers for the full duration of the study, minimising drop out rates.

In this section we outline the daily task that workers were asked to complete, as well as our system design through which we hope to achieve our two aims given above.

6.4.1.1 Task Design

Our biometrics measurement task aimed to capture touch based activity from users whilst they completed two separate tasks, scrolling through a news feed to find a specific article based on information given (similar in design to a Facebook news feed) and whilst counting the number of objects in a set of images, requiring repeat swipe gestures[1].

The two tasks were implemented in an iOS application and would be presented to the user sequentially once they opened the application and began a measurement session. We chose iOS as we need to standardise the device models used in our study for the eventual biometric data to be useful and the number of iOS device models

---

[1] The iOS application was built by the lead author of the Touch Dynamics paper [49].

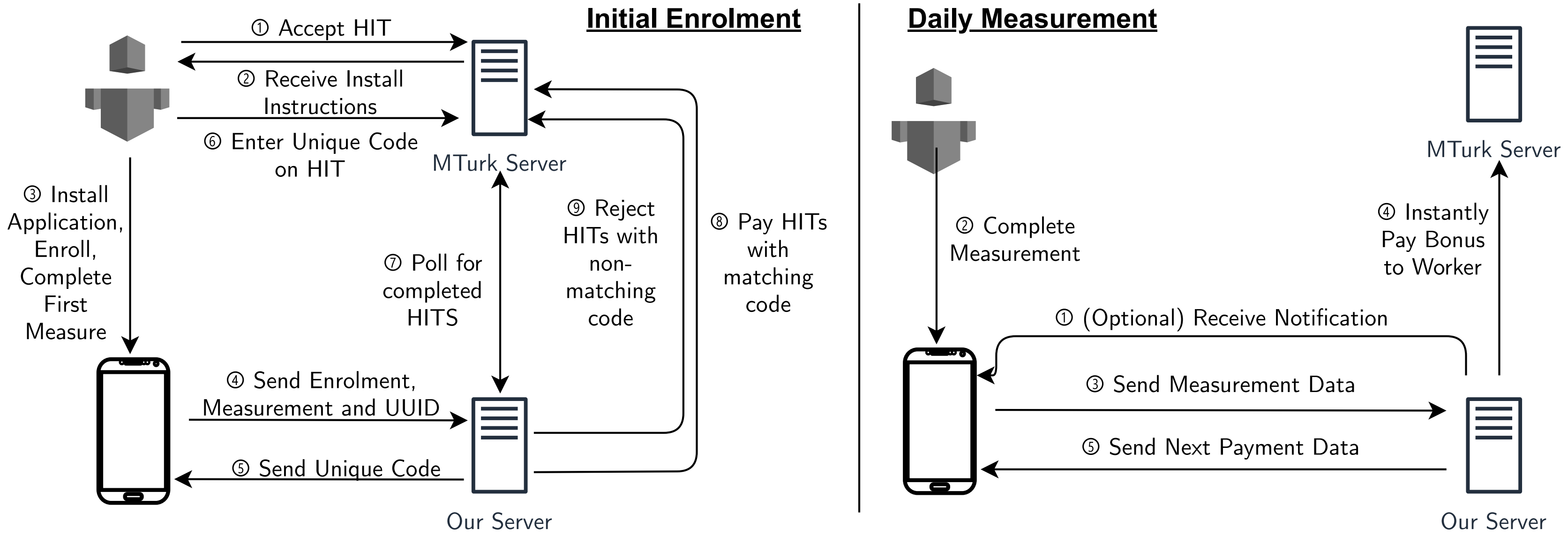

Initial Enrolment
① Accept HIT
② Receive Install Instructions
⑥ Enter Unique Code on HIT
MTurk Server
③ Install Application, Enroll, Complete First Measure
⑦ Poll for completed HITS
⑨ Reject HITs with non-matching code
⑧ Pay HITs with matching code
④ Send Enrolment, Measurement and UUID
⑤ Send Unique Code
Our Server
Daily Measurement
MTurk Server
② Complete Measurement
④ Instantly Pay Bonus to Worker
① (Optional) Receive Notification
③ Send Measurement Data
⑤ Send Next Payment Data
Our Server

Figure 6.2: System diagram for our experiment platform. The left hand side of the figure shows the first enrolment task, whilst the right hand side shows the system operation for the remainder of the experiment period.

available is much smaller than the number of Android device models. In a similar lab based study the same device would be used by all participants.

When outside of a measurement session the application showed them the reward for the next completion of the task, as well as the total value of rewards that could still be obtained for completions of the task in the study period.

Workers were asked to complete both the scrolling and swiping tasks once per day, in a process that takes approximately 4 minutes (median completion time 4 minutes 1.44 seconds) and for the full 31 day (or 7 day) duration of the study period.

6.4.1.2 Reminder Notifications

Previous longitudinal studies with infrequent request intervals have made use of email to notify workers of new tasks to complete [34, 86]. However, these prior works had the benefit of being able to allow workers several days to complete tasks, whereas we sought to collect measurements daily for the study duration. As a consequence we elected to use reminder notifications to improve user retention and encourage participants to complete the task each day, in the hopes of fulfilling both of our aims. These reminder notifications are push notifications to the participant's iOS device, and sent by the server that manages receiving data and processing payments.

During the enrolment process we ask workers to enable notifications. Upon submission of a successful measurement we schedule a reminder notification for the worker the subsequent day at 9 a.m. in their local timezone. An additional notification is sent at 7 p.m. the same day if the worker has not submitted a measurement before this time. We evaluate the effectiveness of these reminder notifications through our post task survey, as well as examining the times at which measurements were completed.

6.4.1.3 Data Quality Assurance

Quality data is a concern for any task on MTurk, but is particularly so in our case due to future studies being conducted with the data and most other touch biometric works using supervised data collection methods. For other tasks requesters use techniques such as obtaining multiple responses which can be combined later for classification problems, or including verification questions that the user must answer correctly to ensure they are human and attentive to the task [92]. We employ a variant of the second of these methods, by validating the answers to each of the sub-tasks completed.

For the news feed scrolling sub-task if an incorrect article is selected then another round of the sub-task must be completed, until 5 rounds are answered correctly. Similarly in the object counting sub-task if the incorrect number of each object is given then another round must be completed. The correct article location is varied randomly for each round of the sub-task, with the number of objects, type of object, and number of images varied for each round of the object-counting sub task.

6.4.2 Mechanical Turk Design

In this subsection we describe the design of the MTurk related system components.

6.4.2.1 Eligible Workers

Our initial study recruitment HIT was available to all MTurk workers. Whilst it is common to restrict studies to master workers, or those with a high HIT acceptance rates, we felt this was unnecessary for our task, as the barriers to entry, namely installing the app and completing the game for the measurement, would act as a filter on workers enrolling who were not motivated to complete the task. Workers self-select to avoid risky tasks [95], and thus we were concerned that restricting participation to master workers may yield poor results, as our experiment differed enough from the platform norm that they may not see it as a worthwhile opportunity.

By ensuring the measurement task was designed robustly, and by requiring input from the touch screen, we believe that performing the tasks in the intended manner is the most efficient way to complete them.

6.4.2.2 Infrastructure Design

We designed our study infrastructure to complete all interactions with MTurk automatically, so that the study could run itself with minimal human oversight. We conduct the completion of repeat tasks in a different way to previous longitudinal studies, in that only one HIT is completed by each worker, which we term the enrolment HIT. After this HIT all additional work completed by the worker is paid using the MTurk bonus system. This allows for faster payments (bonuses are paid instantly on task completion) and allows us to use out of band methods, specifically device notifications, for communicating with the workers and soliciting further work. Furthermore the instant bonus payments help re-assure the workers that they will be paid for extra measurements they complete. In the remainder of this section we detail the on-boarding process for workers who join our study and the daily process completed by the workers in our study.

6.4.2.3 On-boarding Process

Recruitment was conducted by publishing a HIT to MTurk, which workers could claim and subsequently complete. The initial task required workers to download the Testflight[2] application onto an iOS device. We use the Testflight mechanism for loading apps as this allows more control over who can install the application than an App Store listing, which is important is the application is useless without also completing the HIT on MTurk as enrolment into the study requires both.

We limit sign-ups to select iOS devices to ensure similar devices are used for the biometric measurements, so that we can remove any device related effects when examining biometric traits at a later date. This requirement was specified in the HIT title and description. Whilst this could be added as a qualification to the HIT (Primary Mobile Device - iPhone), it is unclear if all workers with access to an iPhone would have enabled this task requirement. In addition, the primary mobile device qualification does not allow requesters to specify a list of supported models as our study requires.

After installing our application the workers completed an enrolment process through the application. This required them to read the informed consent documents, which were also included with the HIT. Subsequently they then had to agree to a set of conditions based on providing consent, by toggling a switch for each condition. The worker then provides us with basic demographic information about them, including country of origin, dominant hand, height and weight.

Lastly the worker completes the study task for the first time, receiving a verification code on completion. The worker enters this code on the HIT page on MTurk. Our backend infrastructure validates this code, approves the HIT if it is correct, and associates the worker ID with the device that was given this code. This verification step allows us to remove spam-like completions, or workers who attempted to guess the code and not complete the task. Some workers do this as they hope that the task may not be approved or rejected before the end of the time out, leading to them being paid automatically, or perhaps in the hope that the application is mis-configured and will accept any text for the verification code.

A flow diagram of the initial on-boarding is shown in Figure 6.2.

---

[2] https://developer.apple.com/testflight/

Measures Completed	**Pay Received ($)**			**Equivalent Hourly Pay ($/h)**		
	LC	**HC**	**HI**	**LC**	**HC**	**HI**
1	1.00	1.00	1.00	7.50	7.50	7.50
11	9.80	12.30	7.25	12.19	15.30	9.02
21	18.60	23.60	18.50	12.61	16.00	12.55
31	27.40	34.90	34.75	12.77	16.23	16.20

Table 6.1: Payment values for participants in the study. Equivalent hourly pay increases with participation duration. Time spent working was calculated by determining the median time taken to complete the daily task (241.44 seconds) and allowing 8 minutes for on-boarding.

6.4.2.4 Daily Measurements

Each day workers are required to complete the same pair of tasks. They can either do this unprompted, or can do it after being sent a reminder notification, the details of which were presented in Section 6.4.1.2. On completing the two tasks the collected measurements are uploaded to our server. The server responds by updating the in app values showing future potential earnings and sends a pay bonus instruction to MTurk, to instantly pay the worker a bonus for the specified amount. By instantly paying bonuses and not requiring any manual approval, we hope to increase our workers confidence that they will be paid for this work completed outside of the normal HIT–payment structure.

The web server schedules the notifications for the next day at this point, assuming the participant has not finished the study.

6.4.2.5 Payment Schemes

We utilized three different payment schemes within our study. Two of the payment schemes, low constant (LC) and high constant (HC) paid the user a fixed amount for each measurement: $0.88 and $1.13 respectively. The third scheme, high increasing (HI), paid users a variable amount, with each submitted measurement's value being determined by the formula:

$$\text{Payment (cents)} = 40 + 5 \times (\text{\# Submitted Measurements - 1}) \tag{6.1}$$

Regardless of payment scheme, all participants were paid a fixed value of $1.00 for the initial HIT for installing the application and completing the first measurement. Table 6.1 shows the expected payment totals for each of the payment schemes with differing study completion percentages, as well as equivalent hourly pay calculated by examining the average completion times of workers on each of our tasks. We note that the equivalent hourly pay for all of our tasks is a minimum of $7.50 (when only the initial measure is completed), well above previously reported median worker hourly pay ($3.18/h, calculated only on time completing tasks) and above the requester average of $11.58/h [57] for those who participate in the majority of the study. Whilst it is possible to pay lower rates to the participants, they are professional workers and not participating due to their love of science. It would also be unethical for us to purposefully pay them small amounts, even if it can be done.

We evaluate the impact of each of these payments schemes on overall participant engagement later in Section 6.5.2. All workers who participated in the 7 day version of the study were assigned to the Low Constant payment scheme.

6.4.2.6 Soliciting Worker Feedback

At the conclusion of our study we posted a survey HIT onto MTurk, to gather insights from the workers who performed work for us. The survey asked questions specifically about the role of notifications, the value of payments, and how comfortable they were with completing studies like this using MTurk.

Unlike the other subtasks, we did not send a notification to workers to complete this as many workers had already uninstalled the app following study completion. We targeted our users through a qualification and offered a relatively high payment of $1.00 to entice the workers to complete the survey. We examine responses from this survey in several of the subsequent sections.

6.5 Touch Dynamics Study Outcomes

In this section we analyse the feedback from our workers and compare this with the activities of the workers within the experiment. This allows us to gain an insight into the effectiveness of the mechanics we implemented to increase worker engagement, and to assess the suitability of MTurk for such experiments.

6.5.1 Study Participants Statistics

For the 31 day study, we enrolled a total of 187 participants (102 female, 82 male, 3 other) with the majority of users (153) from the US and the remaining 32 split over 16 other countries. Of these, 44 were allocated to the HI payment scheme, 54 to HC and the remaining 89 to the LC scheme. While we initially recruited equal numbers of users into the HI and HC schemes, varying rates of rejected submissions (i.e. invalid

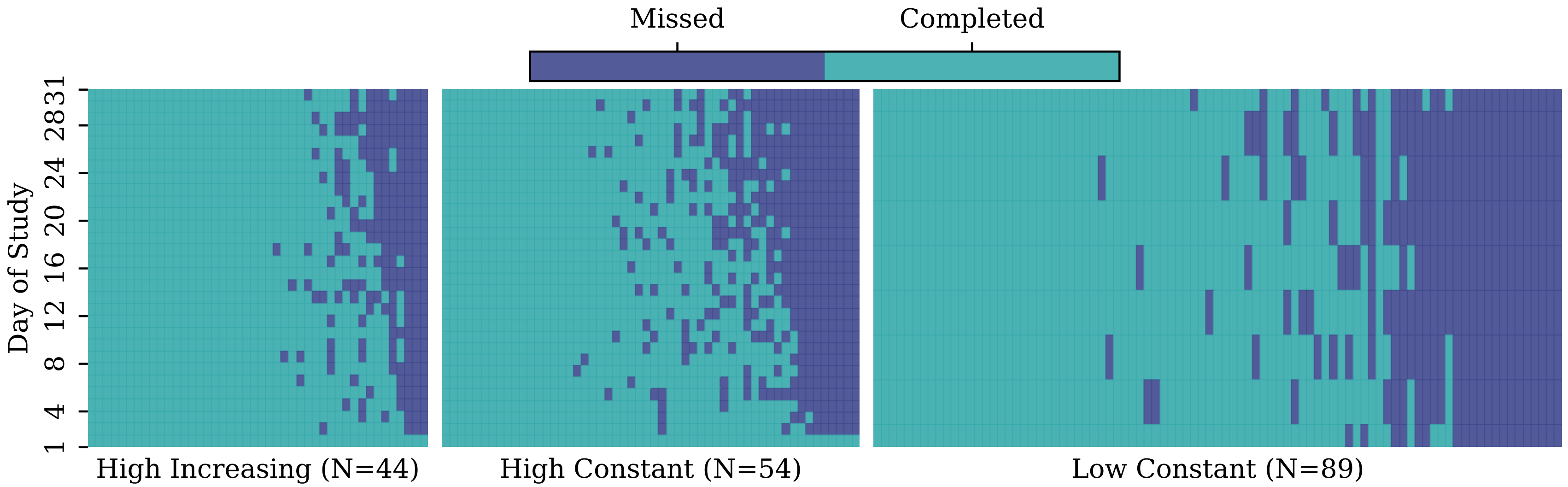


Figure 6.3: Heatmap showing measure completion by day for the participants enrolled in the 31 day study, split by payment scheme. Each vertical line represents a single worker's progress through the study.

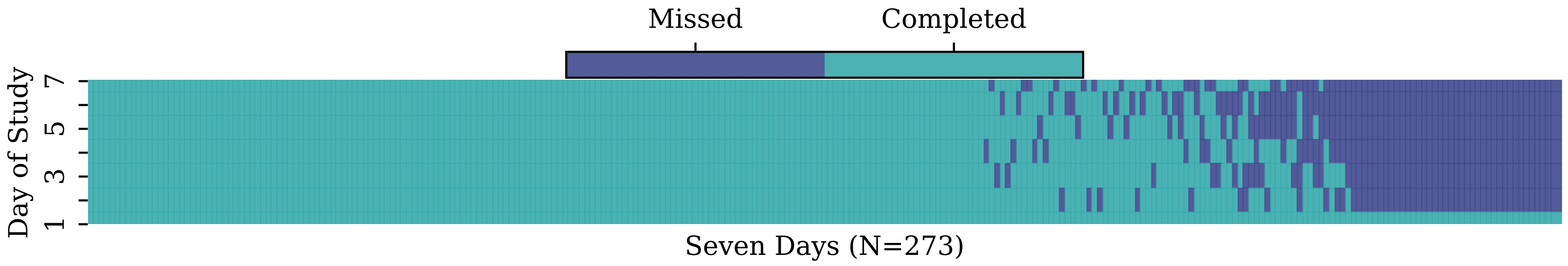


Figure 6.4: Heatmap showing measure completion by day for the participants enrolled in the 7 day study. Each vertical line represents a single worker's progress through the study.

verification codes) led to different numbers of users actually enrolled into the study. These invalid codes could be either due to workers attempting to game the system (i.e. entering random codes in the hope of auto-approval) or workers accepting the HIT unaware of the restriction to iOS devices and submitting random codes instead of returning the HIT. For the 7 day version of the study we enrolled a total of 273 participants who were all paid according to the LC scheme.

A further 26 participants also downloaded the application and completed the demographic questionnaire, but never completed the enrolment into the study by completing the first measurement and submitting the generated code on MTurk. Of these, 15 participants completed the demographic questionnaire but did no measurement and 11 completed the questionnaire and measurement but did not submit the code to MTurk. Without both these steps the participants can't be linked to the study, meaning they can't be paid for the parts of the task they completed. It could be that the design of the task or instructions were not clear enough to be understood by these participants.

In our analysis we primarily discuss the outcomes of the participants who completed the 31 day version of the study, as this gives a longer period of time to examine the effects of our choices and impacts of the study.

Considering the 31 day study, overall most participants stayed engaged throughout, with the heatmap in Figure 6.3 showing the proportion of participants who completed the study remained high until its conclusion, although with a gradual decline in participation throughout the period observed. Within the 31 day study 36.8% of participants completed a measurement every day, with 68.4% completing a measurement on more than 75% of days.

The heatmap reveals certain trends in the participants who missed measurements, with vertical blue lines showing that often once a participant started missing measurements they were unlikely to re-engage with the study, including several users who exhibited this pattern after the initial sign up measurement, which can be seen as the rightmost entries on each of the sub-heatmaps. We see a similar pattern for the seven day version of the study, shown in Figure 6.4. Thirty-nine participants only completed the first measurement, representing 14.3% of participants, a higher proportion than for the 31 day study or any of its respective payment schemes.

This could be explained by participants who are trying to complete the task as fast as possible performing the on-boarding process, completing the measurement and subsequently removing the application, without realising the potential future earnings.

This may be due to a lack of task clarity, as the unusual design of the task (vs. other typical MTurk HITs) means workers can't rely on previous knowledge to complete it. They may have assumed subsequent HITs would be posted for subsequent measurements or may not have understood that the app could be revisited for subsequent daily measurements, leading to them removing it after submitting the HIT completion code. Lack of clarity has been suggested in previous works as a cause of task abandonment [55].

We also note some users show sporadic patterns, where they fail to complete the task on some days, but re-engage subsequently. However, often missing a measurement proved terminal, with 46 of the 118 participants in the 31 day experiment (39%) who missed a measurement finishing the experiment with a run of three or more missed measurements.

6.5.2 Varied Payment Schedules

We evaluate the payment schedules used in our 31 day experiment to determine any impacts they have on completion rate, as well as payment satisfaction compared with completion rate and with payment schedule.

We first investigate the patterns in completion between the three payment schedules, investigating if the differences in: i) participants who only completed 1 measure, ii) mean measures completed (excluding those who dropped out after one) and iii) completing all measurements, are statistically significant between the pairs of payment schemes.

P-values for these comparisons are reported in Table 6.2. It is clear that there is no significant difference between payment schemes on the number of users who complete only one measurement. This suggests that not continuing with the experiment may not be related to payment and is more likely to be failing to understand the daily nature of repeat tasks through the application.

Examining the total measures completed per user, we find that with a two tailed test there is no significant difference between the payment schemes. We conducted a one tailed test to examine if the increasing scheme leads to more measures and if the higher constant value has an impact, but we find no significant difference at the 5% confidence level. Further experiments are necessary to further investigate this effect, particularly with a larger sample size to enable better confidence in the values.

Examining the proportion of participants who completed the task daily, we find that there is a statistically significant difference showing the HI scheme has a higher

Variable	**Test**	**Scheme 1**	**Scheme 2**	**p-value**	**5% sig.**	**10% sig.**
Drop Out After First	Two Proportion Z-Test	HI	HC	0.464	✗	✗
		HI	LC	0.572	✗	✗
		HC	LC	0.133	✗	✗
Measures Completed	Two-Tailed t-Test	HI	HC	0.131	✗	✗
		HI	LC	0.122	✗	✗
		HC	LC	0.870	✗	✗
Measures Completed	One-Tailed t-Test Scheme 1 > Scheme 2	HI	HC	0.065	✗	✓
		HI	LC	0.061	✗	✓
		HC	LC	0.565	✗	✗
Completed Every Day	Two Proportion Z-Test Scheme 1 > Scheme 2	HI	HC	0.019	✓	✓
		HI	LC	0.014	✓	✓
		HC	LC	0.554	✗	✗

Table 6.2: Statistical significance test results for varying payment schedules. Measures completed tests are on the number of measures completed per user. Drop out after first are based on proportion of users who only complete one measurement and never return. Number of samples are HI:44, HC:54 and LC:89.

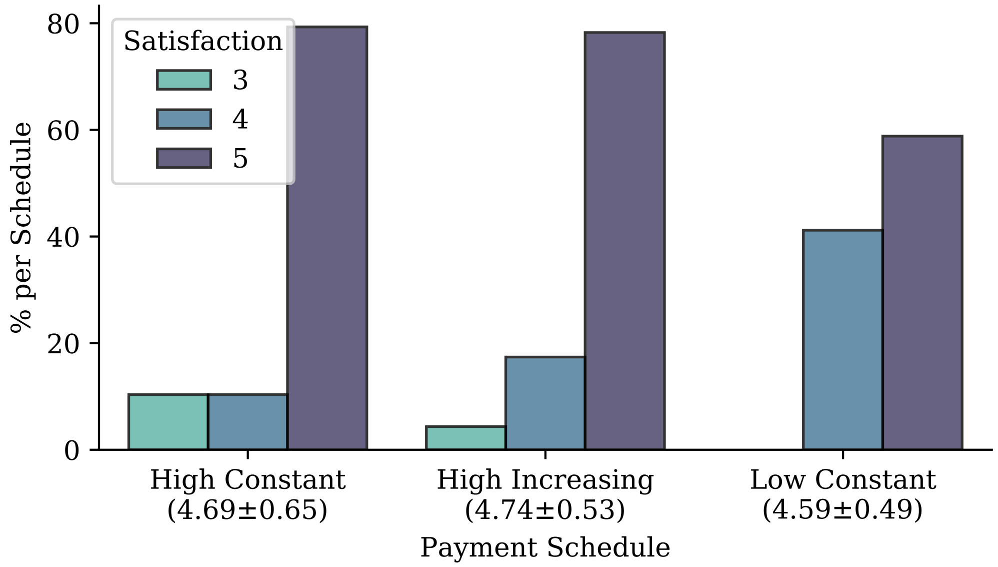


Figure 6.5: Respondent's payment satisfaction, split by payment schedule.

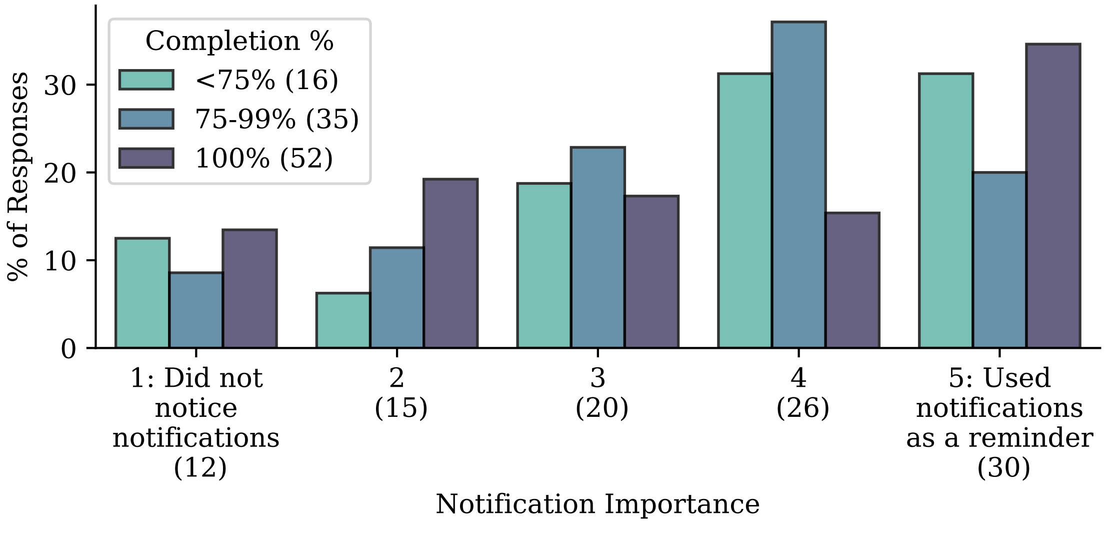


Figure 6.6: Self-reported importance of notifications, broken down by completion rate for the 31 day participants.

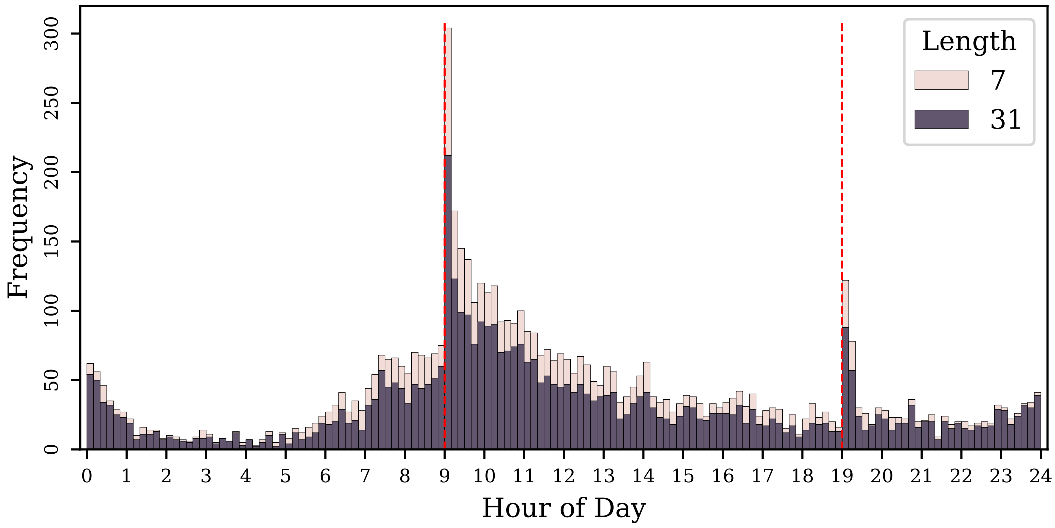


Figure 6.7: Submission times in user's timezone for both the 31 and 7 day experiment lengths. Spikes can be observed directly following reminder notifications at 9 a.m. and 7 p.m (shown by red dashed lines). Bins are spaced at ten minute intervals.

proportion than the two constant schemes. This supports our theory that by participants observing the increasing payments over time – and seeing that their future earnings are dependent on completing it daily – leads to increased motivation to complete the task fully and not miss sessions, as this would impact future earnings.

We also investigate the impact on workers payment satisfaction with respect to their payment schedule through our post-study survey. Figure 6.5 shows the payment satisfaction depending on the payment schedule. We observed mean scores of 4.74, 4.69 and 4.59 for the HI, HC and LC schedules, with none of the differences being statistically significant.

An overall explanation for the lack of significant differences here could be that all of our payment schemes are notably higher than the worker average (and above US minimum wage). Consequently most ratings given are high and workers appear satisfied with completing the study on any of our payment schemes. Further experiments could be conducted to determine lower wage levels where an impact is seen, however it is not ethical to pay workers poorly, so care must be taken in the design of these. The value of determining a minimum bound for pay is also ethically questionable, as it will facilitate the further exploitation of workers with low wages.

6.5.3 Daily Completion Reminders

In our post study survey participants were asked “How important were the daily notifications in reminding you to complete the task?” and answered on a scale from 1 (Completed the task without noticing the notifications) to 5 (Used the notifications as a reminder to complete the daily task). Figure 6.6 shows that 54% of respondents who completed in the 31 day version of the study rated the importance of notifications high (4 or 5) but that the self-reported importance of notifications had no statistically significant effect on actual completion rates. We also looked to see if there was any difference between those who did not enable notifications when registering and those who did, however due to a low sample size not having notifications (22 workers) we could not draw any conclusions.

To further evaluate the actual rather than perceived importance, we also recorded the completion time of each submitted sample (see Figure 6.7). The data shows spikes in the ten minute windows following notification dispatch (9 a.m. and 7 p.m.) followed by a trailing off effect after these notifications. While it could be argued that both are popular times to complete these tasks in general, the size of the spike in specific 10-minute windows strongly suggests a significant impact of the notification timing on prompting workers to complete the task at these times.

6.5.4 Worker Quality and Performance

Our task was sufficiently different from most MTurk tasks that we expected our workers to have limited familiarity with it at first, but to gain confidence and competence in completing the task as they completed it each day. We examine the existence of this trend by studying the completion times for each time of game to evaluate the speed of completing the task. We also examined the number of rounds completed to see if the workers make fewer mistakes over time. When evaluating completion speed we remove outliers, in this case defined as times that are 5 times the inter quartile range above and below the upper and lower quartile respectively. In practice this only removes high values, as times can not be below 0. A total of 254/25358 swipe rounds and 229/22937 scroll rounds were removed due to this. These outliers are fairly extreme values and presumably result from workers pausing work on a round and resuming later.

Figure 6.8 shows the mean round time for each of the two tasks for the 31 day participants. For both tasks we see that workers’ speed increases over time, with the scroll task exhibiting a bigger overall improvement. We also observe that the scroll

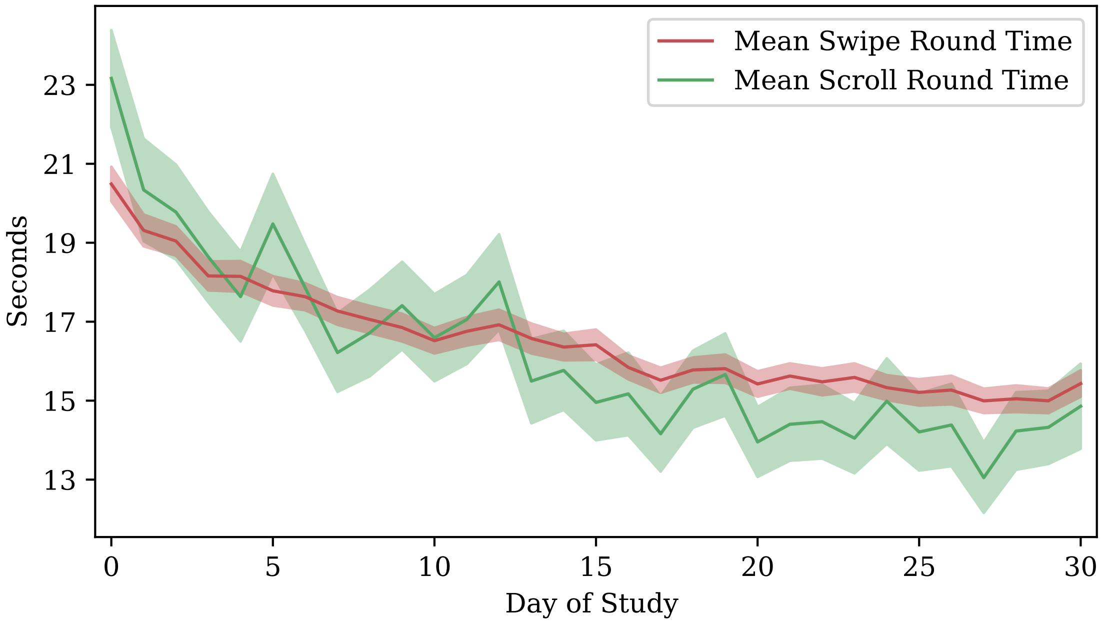


Figure 6.8: Mean round time for each of the games included in the task. We see that for both tasks round times decrease over the course of the experiment, as worker familiarity and skill increases. The effect is higher on the scroll game. Shaded regions show 95% confidence intervals.

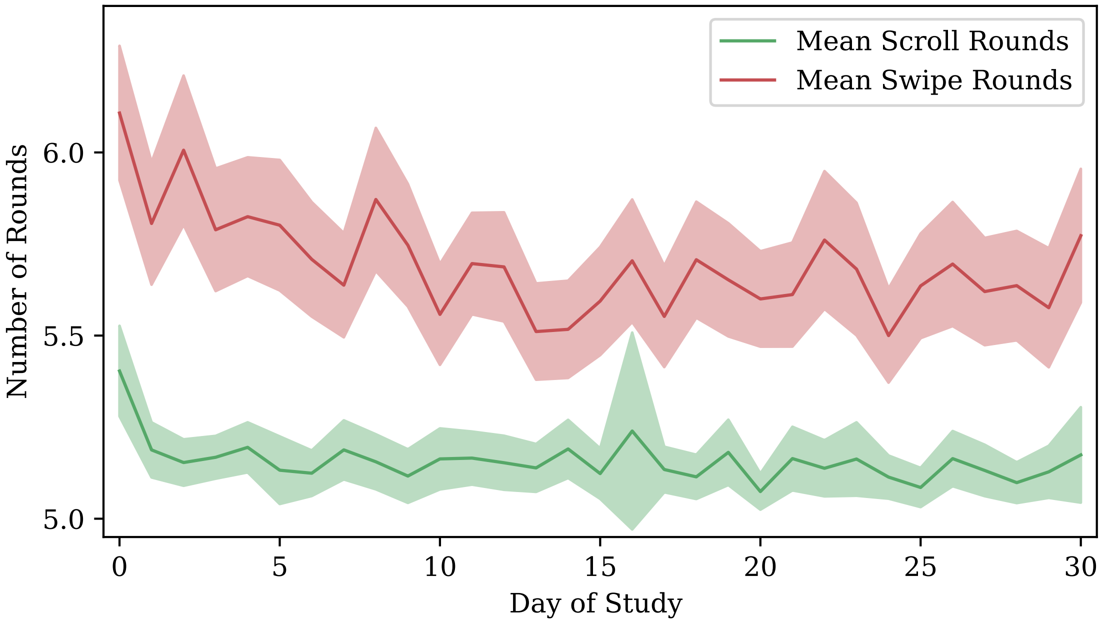


Figure 6.9: Mean number of rounds required to complete each of the two games in the task. 95% confidence intervals indicated by the shaded area. There is a trend indicating in a familiarity/learning effect that resulting in less errors for both tasks.

means fluctuate more than the swipe means and have larger confidence intervals. This is due to the swipe game always featuring the same number of images and swipes, whereas the scroll game has a variable number of scrolls, meaning some rounds will be shorter than others.

Figure 6.9 shows that for both experiments the number of rounds to complete the task decreases over time (a new round is undertaken if a worker answers the previous round incorrectly). This effect is more variable than the round times and is less clear overall than the time reduction. This is likely because the workers make few mistakes anyway (the games are fairly easy) leaving limited room for improvement. In particular the swipe game shows limited overall improvement, perhaps because counting objects is more likely to cause mistakes than identifying the correct article.

For both time and number of rounds we see a clear drop between the first and second measurements, showing that workers quickly grasp the tasks and recall them for their next measurement. In general the learning effect is also more concentrated at the start of the tasks, with the improvements between subsequent days diminishing as the experiment progresses.

6.6 Remote studies and Informed Consent

Academic research involving human participants is subject to IRB approval at most institutions and researchers are typically required to follow a formal ethics procedure involving participants giving informed consent. We performed this process at our institution, with reference: SSD/CUREC1A CS_C1A_19_013. As part of this process, we provided a project information sheet in the HIT and within the app along with the app-based consent form. As part of the consent form, participants indicate to have read and understood the information sheet and that they have received answers to any questions they may have had.

As this method of collecting remote data was new to us, we wished to examine the extent to which participants were truly informed. To do so, during the study's exit survey, we asked participants what they thought the purpose of the study was[3]. The study purpose was clearly stated in the first paragraph of the information sheet. We perform this for 31 day and 7 day workers, and point out that the time interval between the 31 day workers completing the post study survey is greater than that of the 7 day workers. Figures 6.10 and 6.11 show the responses given by participants for

---

[3] By 'the study' we mean the touch authentication study that workers completed, as opposed to this subsequent study of worker behaviour.

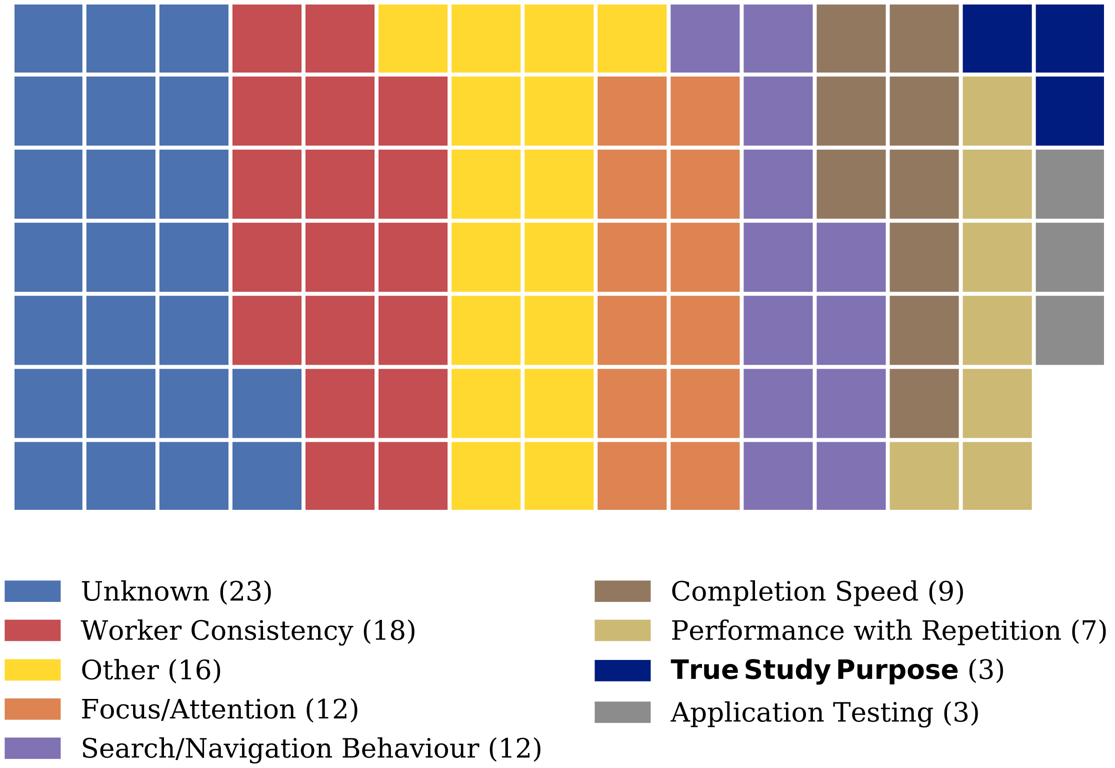


Figure 6.10: Categorised responses given by participants in the 31 day study when asked about the study purpose. Total number of respondents is 103.

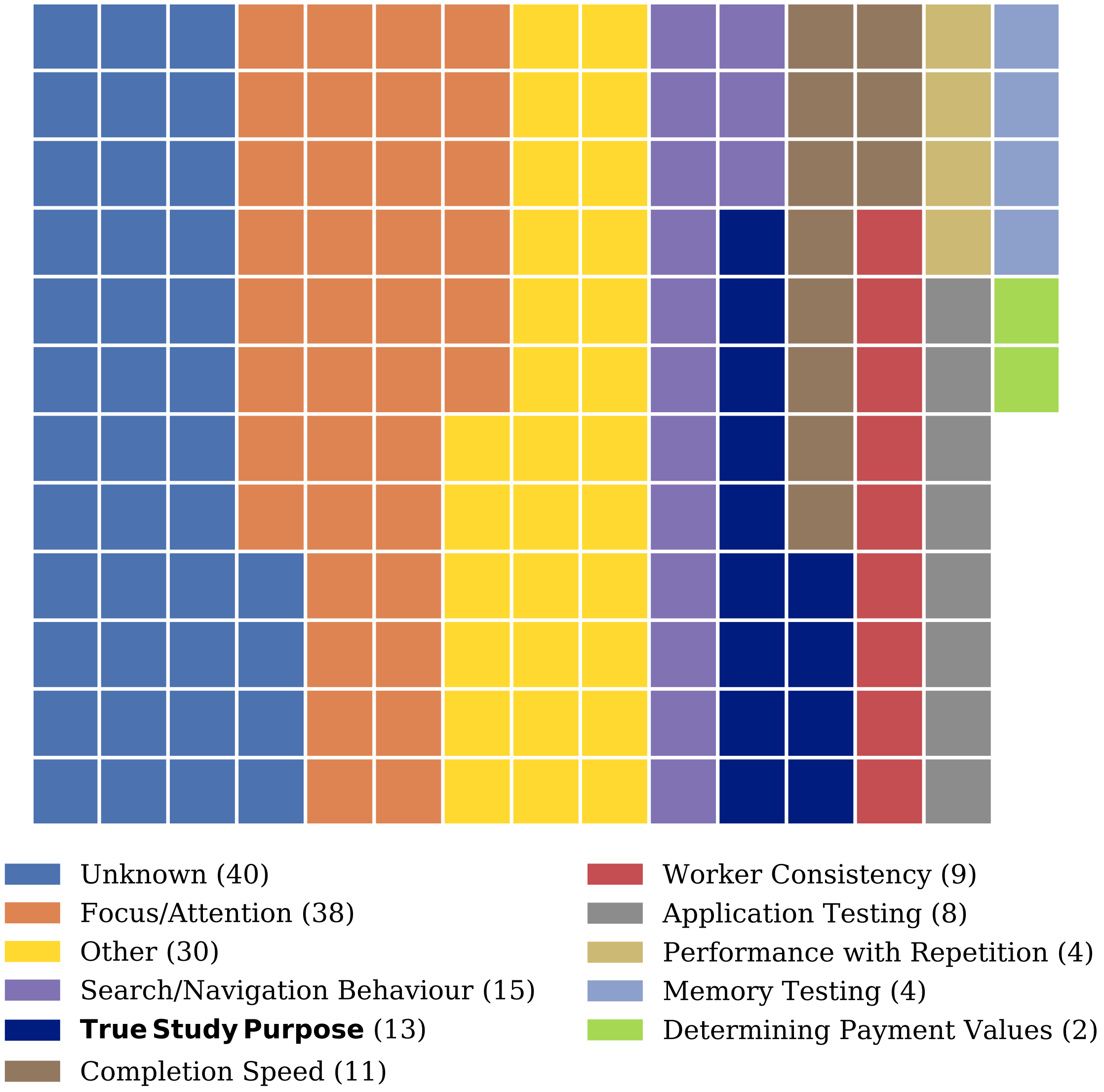


Figure 6.11: Categorised responses given by participants in the 7 day study when asked about the study purpose. Total number of respondents is 174.

the 31 day and 7 day studies respectively. Only 3 respondents' (2.91%) answers from the 31 day study related to the true study purpose. The remaining users either stated that they did not know or gave various guesses based on the tasks (such as attention span, consistency or improvement over time). In the 7 day study 13 respondents (7.47%) gave answers relating to the true study purpose.

While the time gap between enrolment in the study (at which point the information sheet is presented) is at least a week (and over a month for the longer study), the survey results nevertheless suggest that only a small fraction of participants read, understood and retained the information in the project information sheet. This result highlights a severe disconnect between the realities of MTurk and academic due process. Due to the relatively small payment for the average HIT, workers are primed to complete them quickly with minimal delay in order to achieve a good hourly rate of payment. Naturally, this emphasis on speed is opposed to giving informed consent as workers are unlikely to consider reading information sheets worth their time.

While it could be argued that the present study is relatively low-risk as it does not involve collecting sensitive data, this fact would arguably be unknown to participants without reading the information sheet as HITs do not undergo any review process before going live. An additional factor could be the high fraction of HITs posted by companies, that are not required to follow this process, leading to users not being conditioned to seeing consent forms in general.

Our results show that the traditional academic process of informed consent is not fit for purpose on MTurk and does not lead to truly informed consent for the vast majority of participants. For low-risk experiments (such as ours) it could be argued that it is sufficient to give participants the *opportunity* to thoroughly inform themselves without strong verification that they really did but this is less true for riskier studies (e.g. those involving sensitive information). While typical high-risk experiments (such as medical trials) are often limited to the physical domain and rare on MTurk, it is still important to consider options for obtaining truly informed consent. Standard approaches to verify that workers have completed a task properly, such as quiz or trapping questions, could be applied to the information sheets. However these add complexity and significantly increase time spent on the process. This is particularly true for common tasks of short duration as the time spent on consent would heavily outweigh the actual participation time. One caveat to this is that we did not require our workers to be "Master workers" or to meet a work completion threshold, and as such it may be that introducing requirements on worker quality would change this outcome.

6.7 Future Design Suggestions

In light of our experimental results and our experiences with both the PPG and touch based studies, we use this section to distil some design suggestions for those looking to build similar systems in the future.

Notifications for Retention

Our MTurk results suggest that the notifications helped many of our workers stay engaged in the experiment. We observed large spikes in tasks being completed immediately following a notification and user feedback suggested they were useful for many workers. Thus we would suggest using a notification scheme for any future app based studies. For a non-app based study we would suggest either using email notifications, or browser notifications.

Finally we would also suggest improving over our system by sending notifications to re-engage users who have been active but then miss a measurement. In our system this led to no more notifications being sent, but our results suggests that otherwise active workers may miss a measurement and then no longer engage. Sending another notification the day after missed measurements may help re-engage these lost workers.

Design Payment Schemes Carefully

Our payment scheme experiments showed that our increasing payment scheme led to a significantly higher proportion of users completing the entire study, whilst we found no significant difference between the payment scheme and mean number of submitted measures. As such we would recommend using an increasing scheme, but can not recommend an overall payment level. In our study we paid workers significantly more than the average, as we felt that paying them wages below minimum wage is ethically wrong. It may be that workers would perform the task just as well on lower overall wages and that the improvements from increasing the payment with each measure would persist at lower overall payment levels.

Take care with informed consent

On surveying our workers post-study, worryingly few of them still remembered the aim of the study. Whilst it may be that this would improve by filtering worker quality on sign up more aggressively, it may be that the combination of longer duration tasks than normal and workers operating quickly, led to few of them being truly informed when they consented at task sign-up. Thus we would suggest that researchers should verify whether participants' consent was truly informed and that the degree of this verification should be proportionate to the risk level of the experiment.

Automate as much as possible

In our system we automate the initial HIT approval, the notification sending, and all bonus payments each time a measurement is submitted. High levels of automating make it much easier to scale the number of participants in the study, allowing the study to take place without daily involvement of those running the study. In turn this reduces the time burden on the researchers conducting the study. Anecdotally several workers told us in our post study survey that they appreciated the speed with which bonus payments were sent on task completion, and it was mentioned by a reviewer on `turkerview.com` as something that they liked and appreciated this feature.

Design your task with work quality in mind

As in other work previously conducted on MTurk, it is important to ensure the quality of the data submitted through the platform. We decided that it was not possible to do this with a reasonable level of effort for our PPG dataset collection, however it was possible for our touch dynamics collection.

In this case the games required touch input on a mobile device to be completed, making any automatic completion system harder to use. Furthermore workers needed to get the answers to each task round correct in order to proceed. When designing a similar repetitive study, we would suggest employing similar designs, in particular because it is not possible to rescind the bonus payments once they are sent, so workers completing fraudulent work will be able to get paid for completing the task if they are improperly designed.

Part of the motivation for this work was in the hope of using the same method for other biometrics in the future and in particular for collecting voice datasets. For collecting a voice dataset work quality could be ensured through several mechanisms, such as running speech to text on the audio to ensure a minimum quality threshold is met. More advanced mechanisms could also be implemented, such as implementing the measurement capture session to include some speaking and some transcription of audio. In this way the crowd could also be used to transcribe the other crowd members work, with multiple transcriptions being collected for each sample to ensure they are correct, with bonus payments paid once an audio file has been correctly transcribed enough times by others.

6.8 Summary

Overall, MTurk is a viable platform for long-term studies that require daily completion. The participant retention rate is high, with 37% of users completing all 31 samples and 68% completing at least 75% in the 31 day study. We would recommend the platform for conducting studies similar to ours and think that by designing the task well, with a focus on designing for worker retention and engagement, high quality datasets can be obtained.

Examining payment schedules, we observed no statistically significant relationship between payment schedule and payment satisfaction or satisfaction and average completion rate. Likewise there was no statistically significant difference between payment schemes on the number of measurements completed, or the number who only completed the initial measurement.

However, we did find that our payment schedule which increases the payment with each measurement had a statistically significant effect on the proportion of workers who completed all measurements. As such we would recommend future studies implement similar mechanisms in their payment schedules if workers completing the full study is of importance.

Twice-daily push notifications were a useful tool to improve retention, with our data showing large spikes in submissions shortly after reminders notifications are sent.

The demographics obtained in our study without explicit targeting is largely comparable to previous work, showing a roughly equal gender split and a largely US-based set of participants. Our exit survey revealed that only 2.9% of participants read, understood and retained the contents of the participant information sheet in the 31 day study, with 7.5% in the 7 day study, suggesting that the standard academic approach for obtaining informed consent may not be compatible with workers' focus on completing HITs as fast as possible. One counter point to this is that it could be argued that the participants do not need to remember the purpose after the consent is given, and as such did not commit it to memory after deciding to consent.

The general feedback received from workers was positive and our results show that MTurk is a highly viable tool to augment or replace lab studies in fields going beyond the traditional scope of the platform.

Chapter 7

Summary

Contents

7.1 Summary

This thesis aimed to firstly demonstrate the vulnerability of users of voice processing systems to impersonation attacks and in particular draw attention to the amounts and quality of audio data that is required to perform a successful attack. We develop an attack against speaker recognition systems that shows adversaries can impersonate users with limited amounts of data and even if this data is not perfect quality. This motivates the work of the subsequent chapters, where we focus our attention on bringing privacy to voice processing systems.

We first investigate voice privacy from a service providers perspective, by developing and evaluating a system that they can use to protect the audio data that they have stored. This occurs largely within the bounds of the Voice Privacy Challenge 2020, a common set of datasets and protocols for evaluating voice privacy systems for those looking to protect stored data. We highlight problems in the baseline solution in the diversity of voices it produces and seek to address this in our proposed solution. Our results show an improved diversity in the resulting voices and that the anonymization remains if a different comparison system is used.

Following this we investigate voice privacy from an individual standpoint, developing AltVoice, a system to allow a user to replace their voice with a synthetic replacement voice. This system uses a combination of Speech to Text and Text-to-Speech, along with an identity generation component, to allow users to replace their spoken utterances with utterances that appear to be spoken by a different identity. We evaluate the implications of this system on privacy for individuals and their protection from authentication attacks and privacy compromise attacks. Our results highlight the potential of our system, but also identify the components that need further improvement, in particular improving the Text to Speech system's generalisability to unseen identities, as well as the naturalness of speech it creates when compared to human voices.

Finally the thesis explores the collection of biometric datasets using a remote mechanism. The dataset collection process is one of the more cumbersome and restrictive parts of biometric research. Developing research methods that scale better is an important component in collecting larger datasets, which in turn can facilitate the application of deep learning techniques. In this chapter we develop a framework for conducting longitudinal studies with many participants remotely. We evaluate its suitability on a pilot study for PPG, before deploying it for touch dynamics. Our results show that this method can successfully be used for biometric dataset collection if the experiment is designed appropriately.

7.2 Future Work

In each of the chapters of the thesis we give directions for future work related to that specific chapter. Here we instead signpost more general areas of future work that need to be addressed going forwards, in order to ensure voice processing systems work for the users of these systems and that their privacy is protected.

Improve Identity Generation Methods

In Chapters 4 and 5 we proposed several methods for generating identities for anonymous speakers to be used in conjunction with voice privacy systems. These generation methods are essential in voice privacy schemes and need sufficient diversity, distinctiveness and realism (compared to natural voices) in order to be suitable for use as alternate identities. The work in Chapter 4 improves on existing identity generation methods, but still yields a system that can not reflect the diversity of natural voices. Thus further work is required on these identity generation methods in order for voice privacy systems to become usable in the real world.

Reduce Identity Confusion in Speaker Recognition Systems

In Chapter 3 we demonstrate an attack against speaker recognition systems and show that with a pool of voices the performance of the attack can be improved significantly. This stems from the overall high confusion rates between voices in speaker recognition systems, where the best performing systems have equal error rates approaching 3% in text independent settings and 1% in text independent settings [160]. These numbers also mean that given a pair of voices, there is a 3% chance that they are confused with one another for a pair of utterances. This high confusion between voices causes problems that propagates across other voice based systems. For example, this high confusion makes it more difficult to develop speech synthesis systems that can generate different voices and also makes it harder to generate unique identities.

Thus attention needs to be given to improving the ability of speaker recognition systems to discriminate between voices, the benefits of which will be felt across most aspects of the voice security research space. At its current performance level voice biometrics are not secure enough to be used as a standalone method, without further checks for important functions and additional voice liveness methods to prevent attacks, especially in the remote contexts that they are a natural fit for.

The ability to improve voice identification algorithms further rests upon the assumption that the current best results are not due to the (lack of) entropy in human voices and that there is more differentiating information extractable from speech. Other biometrics, such as fingerprint, achieve error rates that are orders of magnitude lower than voice currently does. In the case of fingerprint, features derived from a direct measurement of the individual's trait are used, whereas with voice the measurement is of the output of a human's vocal system. From this measurement underlying features that can be used to separate individual's identities are extracted, with the state of the art features being computed by neural networks trained in an end-to-end fashion. Future advances may not necessarily lie in improving the existing neural network methods. Instead, more effort in capturing the unique aspects of each human's vocal tract, in a similar manner to how other biometrics are used, may yield better results.

7.3 Final Conclusions

As the use of voice interfaces continues to grow in the modern world, it is important to keep a watchful eye on the implications of this to society. Speaker recognition in particular has the ability to add custom functionality to systems and is attractive for authentication due to its relative simplicity for end users.

However caution is needed in the use of these systems. We demonstrate in this thesis that speaker recognition systems are vulnerable to attacks from audio manipulated by adversaries, even when they only have limited information about the victim's voice. When coupled with the large amounts of voice data available freely for many individuals, this raises serious concerns about the ability to impersonate others.

As such care must be taken when storing recordings of audio by providers of voice processing systems. In Chapter 4 we explore a mechanisms that can be applied to stored voice data to protect the original voice traits of those who produced the data. We followed this in Chapter 5 by developing a system for voice system users to protect their own voice.

Ultimately this thesis highlights the difficult problems when considering (remote) speaker recognition security. Speaker recognition can not yet rely on the uniqueness of the voice alone and instead depends on further methods, such as liveness checks, or additional information exchange. Unless more entropy can be extracted from individual speech recordings, it may never be possible to use voice to provide levels of security comparable with other biometrics. Furthermore automated liveness checks are likely not possible against perfect remote replay attacks. As such users should take steps to protect themselves by using systems such as that proposed in Chapter 5.

Those implementing speaker recognition systems need to be aware of the sensitivity of the voice data they hold. System creators should use anonymisation systems such as the one proposed to protect the voice data they hold, so that users can not be re-identified. Given the many practical attacks and as of yet insufficient countermeasures, the field has a long way to go before voice can be used securely as a biometric in remote access scenarios.

Bibliography

[1] Hadi Abdullah, Washington Garcia, Christian Peeters, Patrick Traynor, Kevin R. B. Butler, and Joseph Wilson. Practical hidden voice attacks against speech and speaker recognition systems. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society, 2019. URL: `https://www.ndss-symposium.org/ndss-paper/practical-hidden-voice-attacks-against-speech-and-speaker-recognition-systems/`.

[2] Hadi Abdullah, Muhammad Sajidur Rahman, Washington Garcia, Kevin Warren, Anurag Swarnim Yadav, Tom Shrimpton, and Patrick Traynor. Hear "no evil", see "kenansville"*: Efficient and transferable black-box attacks on speech recognition and voice identification systems. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 712–729. IEEE, 2021.

[3] Mohamed Abou-Zleikha, Zheng-Hua Tan, Mads Græsbøll Christensen, and Søren Holdt Jensen. A discriminative approach for speaker selection in speaker de-identification systems. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 2102–2106, 2015. `doi:10.1109/EUSIPCO.2015.7362755`.

[4] Muhammad Ejaz Ahmed, Il Youp Kwak, Jun Ho Huh, Iljoo Kim, Taekkyung Oh, and Hyoungshick Kim. Void: A fast and light voice liveness detection system. *Proceedings of the 29th USENIX Security Symposium*, (1):2685–2702, 2020.

[5] Shatha J Alghamdi and Lamiaa A Elrefaei. Dynamic authentication of smartphone users based on touchscreen gestures. *Arabian journal for science and engineering*, 43(2):789–810, 2018.

[6] Kevin Allix, Tegawendé F Bissyandé, Jacques Klein, and Yves Le Traon. Are your training datasets yet relevant? In *International Symposium on Engineering Secure Software and Systems*, pages 51–67. Springer, 2015.

[7] Omar Alonso and Ricardo Baeza-Yates. Design and Implementation of Relevance Assessments Using Crowdsourcing. In Paul Clough, Colum Foley, Cathal Gurrin, Gareth J F Jones, Wessel Kraaij, Hyowon Lee, and Vanessa Mudoch, editors, *Advances in Information Retrieval*, pages 153–164, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[8] Apple Siri Team. Hey Siri: An On-device DNN-powered Voice Trigger for Apple's Personal Assistant. 2017. URL: `https://machinelearning.apple.com/2017/10/01/hey-siri.html`.

[9] Apple Siri Team. Personalized Hey Siri. [Accessed 2019-07-08], 2018. URL: `https://machinelearning.apple.com/2018/04/16/personalized-hey-siri.html`.

[10] Sercan Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. Neural voice cloning with a few samples. In *Advances in Neural Information Processing Systems*, pages 10019–10029, 2018.

[11] Fahimeh Bahmaninezhad, Chunlei Zhang, and John Hansen. Convolutional Neural Network Based Speaker De-Identification. 2016(June):255–260, 2018. `doi:10.21437/odyssey.2018-36`.

[12] Niranjan Bidargaddi, Timothy Pituch, Haitham Maaieh, Camille Short, and Victor Strecher. Predicting which type of push notification content motivates users to engage in a self-monitoring app. *Preventive medicine reports*, 11:267–273, 2018.

[13] Frédéric Bimbot, Jean-François Bonastre, Corinne Fredouille, Guillaume Gravier, Ivan Magrin-Chagnolleau, Sylvain Meignier, Teva Merlin, Javier Ortega-García, Dijana Petrovska-Delacrétaz, and Douglas A Reynolds. A tutorial on text-independent speaker verification. *EURASIP Journal on Advances in Signal Processing*, (4):101962, 2004.

[14] Logan Blue, Hadi Abdullah, Luis Vargas, and Patrick Traynor. 2MA: Verifying Voice Commands via Two Microphone Authentication. In *Proceedings of the 13th on Asia Conference on Computer and Communications Security*, pages 89–100. ACM, 2018.

[15] Logan Blue, Hadi Abdullah, Luis Vargas, and Patrick Traynor. 2ma: Verifying voice commands via two microphone authentication. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 89–100, 2018.

[16] Logan Blue, Luis Vargas, and Patrick Traynor. Hello, is it me you're looking for?: Differentiating between human and electronic speakers for voice interface security. In *Proceedings of the 11th Conference on Security & Privacy in Wireless and Mobile Networks*, pages 123–133. ACM, 2018.

[17] D Blumeyer. Relative frequencies of english phonemes, 2012. [Accessed 2019-04-27]. URL: `https://cmloegcmluin.wordpress.com/2012/11/10/relative-frequencies-of-english-phonemes/`.

[18] T. Boult. Robust distance measures for face-recognition supporting revocable biometric tokens. *FGR 2006: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, 2006:560–566, 2006. `doi:10.1109/FGR.2006.94`.

[19] Stewart Brand and Chris Anderson. Mammoth resurrected, geoengineering and other thoughts from a futurist. TED2017, 2017. URL: `https://www.ted.com/talks/stewart_brand_and_chris_anderson_mammoths_resurrected_geoengineering_and_other_thoughts_from_a_futurist`.

[20] Brené Brown. The power of vulnerability. TEDxHouston, 2010. URL: `https://www.ted.com/talks/brene_brown_on_vulnerability`.

[21] David Cameron. Pm's speech at olympic press conference. 2012. URL: `https://www.gov.uk/government/speeches/pms-speech-at-olympics-press-conference`.

[22] Tim Capes, Paul Coles, Alistair Conkie, Ladan Golipour, Abie Hadjitarkhani, Qiong Hu, Nancy Huddleston, Melvyn Hunt, Jiangchuan Li, Matthias Neeracher, Kishore Prahallad, Tuomo Raitio, Ramya Rasipuram, Greg Townsend, Becci Williamson, David Winarsky, Zhizheng Wu, and Hepeng Zhang. Siri on-device deep learning-guided unit selection text-To-speech system. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2017-Augus:4011–4015, 2017. `doi:10.21437/Interspeech.2017-1798`.

[23] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. Hidden voice commands. In *Proceedings of the 25th USENIX Security Symposium*, pages 513–530, 2016.

[24] Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *IEEE Security and Privacy Workshops*, pages 1–7. IEEE, 2018.

[25] Fadi Chehadé and Bryn Freedman. What everyday citizens can do to claim power on the internet. TED Salon: Verizon, 2018. URL: `https://www.ted.com/talks/fadi_chehade_what_everyday_citizens_can_do_to_claim_power_on_the_internet`.

[26] Nanxin Chen, Yanmin Qian, Heinrich Dinkel, Bo Chen, and Kai Yu. Robust deep feature for spoofing detection - the SJTU system for asvspoof 2015 challenge. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2015-January(61222208):2097–2101, 2015.

[27] Si Chen, Kui Ren, Sixu Piao, Cong Wang, Qian Wang, Jian Weng, Lu Su, and Aziz Mohaisen. You can hear but you cannot steal: Defending against voice impersonation attacks on smartphones. In *Proceedings of the 37th International Conference on Distributed Computing Systems*, pages 183–195. IEEE, 2017.

[28] Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Ekaterina Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, and Michiel Bacchiani. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 4774–4778. IEEE, 2018. `doi:10.1109/ICASSP.2018.8462105`.

[29] Ju-Chieh Chou and Hung-yi Lee. One-shot voice conversion by separating speaker and content representations with instance normalization. In Gernot Kubin and Zdravko Kacic, editors, *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 664–668. ISCA, 2019. `doi:10.21437/Interspeech.2019-2663`.

[30] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. In *Proc. Interspeech 2018*, pages 1086–1090, 2018. URL: `http://dx.doi.org/10.21437/Interspeech.2018-1929`, `doi:10.21437/Interspeech.2018-1929`.

[31] Erica Cooper, Xin Wang, Yi Zhao, Yusuke Yasuda, and Junichi Yamagishi. Pretraining strategies, waveform model choice, and acoustic configurations for multi-speaker end-to-end speech synthesis, 2020. `arXiv:2011.04839`.

[32] Richard V. Cox, Donald E. Bock, Keith B. Bauer, James D. Johnston, and James H. Synder. Analog Voice Privacy System. *AT&T Technical Journal*, 66(1):119–131, 1987. `doi:10.1002/j.1538-7305.1987.tb00480.x`.

[33] Amy Cuddy. Your body language may shape who you are. TEDGlobal 2012, 2012. URL: `https://www.ted.com/talks/amy_cuddy_your_body_language_shapes_who_you_are`.

[34] Timothy M. Daly and Rajan Nataraajan. Swapping bricks for clicks: Crowdsourcing longitudinal data on Amazon Turk. *Journal of Business Research*, 68(12):2603–2609, 2015. URL: `http://dx.doi.org/10.1016/j.jbusres.2015.05.001`, `doi:10.1016/j.jbusres.2015.05.001`.

[35] Phillip L. De Leon, Michael Pucher, Junichi Yamagishi, Inma Hernaez, and Ibon Saratxaga. Evaluation of speaker verification security and detection of HMM-based synthetic speech. *Transactions on Audio, Speech and Language Processing*, 2012.

[36] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011. `doi:10.1109/TASL.2010.2064307`.

[37] Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, Philippe Cudr, and Philippe Cudré-Mauroux. Scaling-up the Crowd: Micro-Task Pricing Schemes for Worker Retention and Latency Improvement. *Second AAAI Conference on Human Computation and Crowdsourcing*, (Hcomp):50–58, 2014.

[38] First Direct. First Direct phone banking - voice ID security. URL: `https://www1.firstdirect.com/banking/ways-to-bank/telephone-banking/`.

[39] Simon Eberz, Kasper B. Rasmussen, Vincent Lenders, and Ivan Martinovic. Evaluating Behavioral Biometrics for Continuous Authentication. In *Proceedings of the 12th Asia Conference on Computer and Communications Security*, pages 386–399, 2017.

[40] Daniel P. W. Ellis. PLP and RASTA (and MFCC, and inversion) in Matlab, 2005. [Accessed 2019-07-08]. URL: `http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/`.

[41] Serife Kucur Ergünay, Elie Khoury, Alexandros Lazaridis, and Sebastien Marcel. On the vulnerability of speaker verification to realistic voice spoofing. In *Proceedings of the 7th International Conference on Biometrics Theory, Applications and Systems*, pages 1–6. IEEE, 2015.

[42] Nicholas Evans, Tomi Kinnunen, and Junichi Yamagishi. Spoofing and countermeasures for automatic speaker verification. *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 925–929, 2013.

[43] Fuming Fang, Xin Wang, Junichi Yamagishi, Isao Echizen, Massimiliano Todisco, Nicholas Evans, and Jean-Francois Bonastre. Speaker Anonymization Using X-vector and Neural Waveform Models. pages 3–8, 2019. URL: `http://arxiv.org/abs/1905.13561`, `arXiv:1905.13561`.

[44] Fuming Fang, Junichi Yamagishi, Isao Echizen, Md Sahidullah, and Tomi Kinnunen. Transforming acoustic characteristics to deceive playback spoofing countermeasures of speaker verification systems. *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, (ASVspoof):1–9, 2018. URL: `http://arxiv.org/abs/1809.04274`, `arXiv:1809.04274`.

[45] Gunnar Fant. *Acoustic theory of speech production: with calculations based on X-ray studies of Russian articulations*. Number 2. Walter de Gruyter, 1970.

[46] Electronic Frontier Foundation. NSA spying. https://www.eff.org/nsa-spying, 2021.

[47] Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE transactions on information forensics and security*, 8(1):136–148, 2012.

[48] Mark Gales and Steve Young. The Application of Hidden Markov Models in Speech Recognition. *Foundations and Trends® in Signal Processing*, 1(3):195–304, 2007. `doi:10.1561/2000000004`.

[49] Martin Georgiev, Simon Eberz, Henry Turner, Giulio Lovisotto, and Ivan Martinovic. Common evaluation pitfalls in touch-based authentication systems. *arXiv preprint arXiv:2201.10606*, 2022.

[50] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. *NIPS Proceedings*, page iii, 2014. URL: `https://linkinghub.elsevier.com/retrieve/pii/B9780408001090500018`, `arXiv:arXiv:1011.1669v3`, `doi:10.1016/B978-0-408-00109-0.50001-8`.

[51] Google. WebRTC. URL: `https://webrtc.org/`.

[52] Google. Set up Voice Match on Google Home - Google Home Help, 2018. [Accessed 2019-07-08]. URL: `https://support.google.com/googlehome/answer/7323910`.

[53] Project Gutenberg. Frequently asked questions about project gutenberg. https://www.gutenberg.org/help/faq.html.

[54] Lei Han, Eddy Maddalena, Alessandro Checco, Cristina Sarasua, Ujwal Gadiraju, Kevin Roitero, and Gianluca Demartini. Crowd worker strategies in relevance judgment tasks. *WSDM 2020 - Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 241–249, 2020. `doi:10.1145/3336191.3371857`.

[55] Lei Han, Kevin Roitero, Ujwal Gadiraju, Cristina Sarasua, Alessandro Checco, Eddy Maddalena, and Gianluca Demartini. The Impact of Task Abandonment in Crowdsourcing. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2019. `doi:10.1109/tkde.2019.2948168`.

[56] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. Deep Speech: Scaling up end-to-end speech recognition. pages 1–12, 2014. URL: `http://arxiv.org/abs/1412.5567`, `arXiv:1412.5567`, `doi:arXiv:1412.5567v2`.

[57] Kotaro Hara, Abigail Adams, Kristy Milland, Saiph Savage, Chris Callison-Burch, and Jeffrey P. Bigham. A data-driven analysis of workers' earnings on amazon mechanical turk. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–14, New York, NY, USA, 2018. Association for Computing Machinery. `doi:10.1145/3173574.3174023`.

[58] Reed Hastings. How netflix changed entertainment – and where it's headed. TED2018, 2018. URL: `https://www.ted.com/talks/reed_hastings_how_netflix_changed_entertainment_and_where_it_s_headed`.

[59] Kenji Hata, Ranjay Krishna, Li Fei-Fei, and Michael S. Bernstein. A glimpse far into the future: Understanding long-term crowd worker quality. *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, pages 889–901, 2017. `arXiv:1609.04855`, `doi:10.1145/2998181.2998248`.

[60] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer. End-to-end text-dependent speaker verification. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2016-May(Section 3):5115–5119, 2016. `arXiv:1509.08062`, `doi:10.1109/ICASSP.2016.7472652`.

[61] Turid Helland and Randi Kaasa. Dyslexia in english as a second language. *Dyslexia*, 11(1):41–60, 2005.

[62] Paul Hitlin. Research in the Crowdsourcing Age, a Case Study. *Pew Research Center*, (July):1–7, 2016. URL: `https://www.pewresearch.org/internet/2016/07/11/research-in-the-crowdsourcing-age-a-case-study/%0Ahttp://www.pewinternet.org/2016/07/11/research-in-the-crowdsourcing-age-a-case-study/`.

[63] HMRC. Voice identification privacy notice. URL: `https://www.gov.uk/government/publications/voice-identification-privacy-notice/voice-identification-privacy-notice`.

[64] HSBC. Voice ID — HSBC UK, 2018. [Accessed 2019-07-08]. URL: `https://www.hsbc.co.uk/1/2/voice-id`.

[65] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang. Voice conversion from non-parallel corpora using variational auto-encoder. In *Proceedings of the Signal and Information Processing Association Annual Summit and Conference*, pages 1–6. IEEE, 2016.

[66] Tsuen-Ho Hsu and Jia-Wei Tang. Development of hierarchical structure and analytical model of key factors for mobile app stickiness. *Journal of Innovation & Knowledge*, 5(1):68–79, 2020. URL: `https://www.sciencedirect.com/science/article/pii/S2444569X19300204`, `doi:https://doi.org/10.1016/j.jik.2019.01.006`.

[67] ITU-T. Recommendation P.808: Subjective evaluation of speech quality with a crowdsourcing approach, 2018. URL: `https://www.itu.int/rec/T-REC-P.808/en`.

[68] Ye Jia, Yu Zhang, Ron J. Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez-Moreno, and Yonghui Wu. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 4485–4495, 2018. URL: `https://proceedings.neurips.cc/paper/2018/hash/6832a7b24bc06775d02b7406880b93fc-Abstract.html`.

[69] Andrew Teoh Beng Jin, David Ngo Chek Ling, and Alwyn Goh. Biohashing: Two factor authentication featuring fingerprint data and tokenised random number. *Pattern Recognition*, 37(11):2245–2255, 2004. `doi:10.1016/j.patcog.2004.04.011`.

[70] Qin Jin, Arthur R. Toth, Tanja Schultz, and Alan W. Black. Speaker de-identification via voice transformation. *Proceedings of the 2009 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2009*, pages 529–533, 2009. `doi:10.1109/ASRU.2009.5373356`.

[71] Nicolas Kaufmann, Thimo Schulze, and Daniel Veit. More than fun and money. Worker Motivation in Crowdsourcing – A Study on Mechanical Turk. *Proceedings of the Seventeenth Americas Conference on Information Systems*, 4(2009):1–11, 2011. URL: `http://schader.bwl.uni-mannheim.de/fileadmin/files/publikationen/Kaufmann_Schulze_Veit_2011_-_More_than_fun_and_money_Worker_motivation_in_Crowdsourcing_-_A_Study_on_Mechanical_Turk_AMCIS_2011.pdf`.

[72] Elie Khoury, Laurent El Shafey, and Sebastien Marcel. Spear: An open source toolbox for speaker recognition based on Bob. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 1655–1659. IEEE, 2014.

[73] Tomi Kinnunen and Haizhou Li. An overview of text-independent speaker recognition: From features to supervectors. *Speech Communication*, 52(1):12–40, 2010.

[74] Tomi Kinnunen, Massimiliano Todisco, Nicholas Evans, Junichi Yamagishi, and Kong Aik Lee. The ASVspoof 2017 Challenge : Assessing the Limits of Replay Spoofing Attack Detection National Institute of Informatics , Japan. (i):2–6, 2017.

[75] Tomi Kinnunen, Zhi-Zheng Wu, Kong Aik Lee, Filip Sedlak, Eng Siong Chng, and Haizhou Li. Vulnerability of Speaker Verification Systems Against Voice Conversion Spoofing Attacks: the Case of Telephone Speech. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 4401–4404. IEEE, 2012.

[76] Kazuhiro Kobayashi, Tomoki Toda, Graham Neubig, Sakriani Sakti, and Satoshi Nakamura. Statistical singing voice conversion with direct waveform modification based on the spectrum differential. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, (September):2514–2518, 2014.

[77] Artemy Kolchinsky and Brendan D Tracey. Estimating mixture entropy with pairwise distances. *Entropy*, 19(7):361, 2017.

[78] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *CoRR*, abs/1602.07332, 2016. URL: `http://arxiv.org/abs/1602.07332`, `arXiv:1602.07332`.

[79] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[80] Paul Lamere, Philip Kwok, Evandro Gouvea, Bhiksha Raj, Rita Singh, William Walker, Manfred Warmuth, and Peter Wolf. The cmu sphinx-4 speech recognition system. In *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong*, volume 1, pages 2–5, 2003.

[81] Yee Wah Lau, Dat Tran, and Michael Wagner. Testing Voice Mimicry with the YOHO Speaker Verification Corpus. In *Proceedings of the 9th International Conference on Knowledge-Based Intelligent Information And Engineering Systems*, volume 3584, pages 15–21, 2005.

[82] Galina Lavrentyeva, Sergey Novoselov, Egor Malykh, Alexander Kozlov, Oleg Kudashev, and Vadim Shchemelinin. Audio replay attack detection with deep learning frameworks. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, volume 2017-Augus, pages 82–86, 2017. `doi:10.21437/Interspeech.2017-360`.

[83] Abner Li. Google begins replacing full 'Voice Match' phone unlock w/ Assistant-only lock screen access, feb 2019. URL: `https://9to5google.com/2019/02/28/google-replacing-voice-match-unlock/`.

[84] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL: `http://arxiv.org/abs/1405.0312`, `arXiv:1405.0312`.

[85] J Lindberg and Mats Blomberg. Vulnerability In Speaker Verification - A Study Of Technical Impostor Techniques. *Proc. Eurospeech*, 3(MARCH 2001):1211–1214, 2001. URL: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.12.9603`.

[86] Leib Litman, Jonathan Robinson, and Tzvi Abberbock. TurkPrime.com: A versatile crowdsourcing data acquisition platform for the behavioral sciences. *Behavior Research Methods*, 49(2):433–442, 2017. URL: `http://dx.doi.org/10.3758/s13428-016-0727-z`, `doi:10.3758/s13428-016-0727-z`.

[87] Li Juan Liu, Zhen Hua Ling, Yuan-Jiang, Ming-Zhou, and Li Rong Dai. Wavenet vocoder with limited training data for voice conversion. *Proceedings of the Annual Conference of the International Speech Communication*

Association, INTERSPEECH, 2018-Septe(September):1983–1987, 2018. `doi:10.21437/Interspeech.2018-1190`.

[88] Lloyds Bank. Voice ID — Lloyds Bank, 2019. [Accessed 2019-07-08]. URL: `https://www.lloydsbank.com/contact-us/voice-id.asp`.

[89] Giulio Lovisotto, Henry Turner, Simon Eberz, and Ivan Martinovic. Seeing red: Ppg biometrics using smartphone cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 818–819, 2020.

[90] Carmen Magarinos, Paula Lopez-Otero, Laura Docio-Fernandez, Eduardo Rodriguez-Banga, Daniel Erro, and Carmen Garcia-Mateo. Reversible speaker de-identification using pre-trained transformation functions. *Computer Speech & Language*, 46:36–52, 2017.

[91] Mohamed Maouche, Brij Mohan Lal Srivastava, Nathalie Vauquier, Aurélien Bellet, Marc Tommasi, and Emmanuel Vincent. A Comparative Study of Speech Anonymization Metrics. In *Proc. Interspeech 2020*, pages 1708–1712, 2020. URL: `http://dx.doi.org/10.21437/Interspeech.2020-2248`, `doi:10.21437/Interspeech.2020-2248`.

[92] Winter Mason and Siddharth Suri. Conducting behavioral research on Amazon's Mechanical Turk. *Behavior Research Methods*, 44(1):1–23, 2012. `doi:10.3758/s13428-011-0124-6`.

[93] Driss Matrouf, J-F Bonastre, and Corinne Fredouille. Effect of speech transformation on impostor acceptance. In *Proceedings of the 31st International Conference on Acoustics Speech and Signal Processing*, volume 1. IEEE, 2006.

[94] Tomoko Matsui, Fernando Villavicencio, Sayaka Shiota, Isao Echizen, Junichi Yamagishi, and Nobutaka Ono. Voice Liveness Detection for Speaker Verification based on a Tandem Single/Double-channel Pop Noise Detector. *Odyssey 2016*, 2016:259–263, 2016. `doi:10.21437/odyssey.2016-37`.

[95] Brian McInnis, Dan Cosley, Chaebong Nam, and Gilly Leshed. Taking a hit: Designing around rejection, mistrust, risk, and workers' experiences in Amazon Mechanical Turk. *Conference on Human Factors in Computing Systems - Proceedings*, pages 2271–2282, 2016. `doi:10.1145/2858036.2858539`.

[96] Paul Mermelstein. Distance measures for speech recognition, psychological and instrumental. *Pattern recognition and artificial intelligence*, 116:374–388, 1976.

[97] Microsoft ML Blog Team. Now available: Speaker & video apis from microsoft project oxford. URL: `https://blogs.technet.microsoft.com/machinelearning/2015/12/14/now-available-speaker-video-apis-from-microsoft-project-oxford/`.

[98] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa. WORLD : A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications. (7):1877–1884, 2016.

[99] Andrew Morris, Viktoria Maier, and Phil Green. From wer and ril to mer and wil: improved evaluation measures for connected speech recognition. 01 2004.

[100] Mozilla. Deepspeech release 0.9.3 on Github. 2021. URL: `https://github.com/mozilla/DeepSpeech/releases/tag/v0.9.3`.

[101] Mozilla. Mozilla common voice dataset. 2021. URL: `https://commonvoice.mozilla.org/en`.

[102] Aymen Mtibaa, Dijana Petrovska-Delacrétaz, and Ahmed Ben Hamida. Cancelable speaker verification system based on binary Gaussian mixtures. *2018 4th International Conference on Advanced Technologies for Signal and Image Processing, ATSIP 2018*, pages 1–6, 2018. `doi:10.1109/ATSIP.2018.8364513`.

[103] Dibya Mukhopadhyay, Maliheh Shirvanian, and Nitesh Saxena. All your voices are belong to us: Stealing voices to fool humans and machines. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9327:599–621, 2015. `doi:10.1007/978-3-319-24177-7_30`.

[104] Rahul Murmuria, Angelos Stavrou, Daniel Barbará, and Dan Fleck. Continuous authentication on mobile devices using power consumption, touch gestures and physical movement of users. In *International Symposium on Recent Advances in Intrusion Detection*, pages 405–424. Springer, 2015.

[105] Elon Musk. The future we're building – and boring. TED 2017, 2017. URL: `https://www.ted.com/talks/elon_musk_the_future_we_re_building_and_boring`.

[106] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: A large-scale speaker identification dataset. In *Proc. Interspeech 2017*, pages 2616–2620, 2017. URL: `http://dx.doi.org/10.21437/Interspeech.2017-950`, `doi:10.21437/Interspeech.2017-950`.

[107] Karthik Nandakumar and Anil K. Jain. Biometric template protection: Bridging the performance gap between theory and practice. *IEEE Signal Processing Magazine*, 32(5):88–100, 2015. `doi:10.1109/MSP.2015.2427849`.

[108] Andreas Nautsch, Jose Patino, Natalia Tomashenko, Junichi Yamagishi, Paul-Gauthier Noé, Jean-François Bonastre, Massimiliano Todisco, and Nicholas Evans. The privacy zebra: Zero evidence biometric recognition assessment. 05 2020.

[109] Paul-Gauthier Noé, Jean-François Bonastre, Driss Matrouf, Natalia Tomashenko, Andreas Nautsch, and Nicholas Evans. Speech pseudonymisation assessment using voice similarity matrices. *arXiv preprint arXiv:2008.13144*, 2020.

[110] Beryl Noë, Liam D. Turner, David E.J. Linden, Stuart M. Allen, Bjorn Winkens, and Roger M. Whitaker. Identifying indicators of smartphone addiction through user-app interaction. *Computers in Human Behavior*, 99:56–65, 2019. URL: `https://www.sciencedirect.com/science/article/pii/S0747563219301712`, `doi:https://doi.org/10.1016/j.chb.2019.04.023`.

[111] V. Panayotov, G. Chen, D. Povey, and S. 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, 2015. `doi:10.1109/ICASSP.2015.7178964`.

[112] 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.

[113] Gabriele Paolacci, Jesse Chandler, and Panagiotis G Ipeirotis. Running experiments on amazon mechanical turk. *Judgment and Decision making*, 5(5):411–419, 2010.

[114] Tanvina B. Patel and Hemant A. Patil. Combining evidences from mel cepstral, cochlear filter cepstral and instantaneous frequency features for detection of natural vs. spoofed speech. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2015-January:2062–2066, 2015.

[115] M. Paulini, C. Rathgeb, A. Nautsch, H. Reichau, H. Reininger, and C. Busch. Multi-bit allocation: Preparing voice biometrics for template protection. *Odyssey 2016: Speaker and Language Recognition Workshop*, pages 291–296, 2016. `doi:10.21437/Odyssey.2016-42`.

[116] M. Pobar and I. Ipšić. Online speaker de-identification using voice transformation. In *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1264–1267, 2014. `doi:10.1109/MIPRO.2014.6859761`.

[117] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *Proceedings of the 2011 workshop on automatic speech recognition and understanding*. IEEE, 2011.

[118] Jianwei Qian, Haohua Du, Jiahui Hou, Linlin Chen, Taeho Jung, and Xiang Yang Li. Hidebehind: Enjoy voice input with voiceprint unclonability and anonymity. *SenSys 2018 - Proceedings of the 16th Conference on Embedded Networked Sensor Systems*, pages 82–94, 2018. `doi:10.1145/3274783.3274855`.

[119] Yao Qin, Nicholas Carlini, Ian Goodfellow, Garrison Cottrell, and Colin Raffel. Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition. 2019. URL: `http://arxiv.org/abs/1903.10346`, `arXiv:1903.10346`.

[120] Nalini K. Ratha, Sharat Chikkerur, Jonathan H. Connell, and Ruud M. Bolle. Generating cancelable fingerprint templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):561–572, 2007. `doi:10.1109/TPAMI.2007.1004`.

[121] Christian Rathgeb and Andreas Uhl. A Survey on Biometric Cryptosystems. pages 1–25, 2011.

[122] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing: A Review Journal*, 10(1):19–41, 2000. `doi:10.1006/dspr.1999.0361`.

[123] Sir Ken Robinson. Do schools kill creativity? TED, 2006. URL: `www.ted.com/talks/ken_robinson_says_schools_kill_creativity`.

[124] Rodrigo Rocha, Davide Carneiro, and Paulo Novais. Continuous authentication with a focus on explainability. *Neurocomputing*, 423:697–702, 2020.

[125] Jakob Rogstadius, Vassilis Kostakos, Aniket Kittur, Boris Smus, Jim Laredo, and Maja Vukovic. An Assessment of Intrinsic and Extrinsic Motivation on Task Performance in Crowdsourcing Markets. *Fifth International AAAI Conference on Weblogs and Social Media*, 2011. `doi:10.1016/S0020-7683(00)00068-8`.

[126] Andrew Rosenberg and Bhuvana Ramabhadran. Bias and statistical significance in evaluating speech synthesis with mean opinion scores. In *Proc. Interspeech 2017*, pages 3976–3980, 2017. URL: `http://dx.doi.org/10.21437/Interspeech.2017-479`, `doi:10.21437/Interspeech.2017-a479`.

[127] Lea Schönherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding. 54(1), 2018. URL: `http://arxiv.org/abs/1808.05665`, `arXiv:1808.05665`, `doi:arXiv:1808.05665v2`.

[128] K. Shelley and S. Shelley. *Pulse Oximeter Waveform: Photoelectric Plethysmography.* Clinical Monitoring. Saunders Company, 2001.

[129] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[130] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, Rif A. Saurous, Yannis Agiomvrgiannakis, and Yonghui Wu. Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions. *ICASSP,*

IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 2018-April:4779–4783, 2018. `doi:10.1109/ICASSP.2018.8461368`.

[131] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, R. J. Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions. *CoRR*, abs/1712.05884, 2017. URL: `http://arxiv.org/abs/1712.05884`, `arXiv:1712.05884`.

[132] Sayaka Shiota, Fernando Villavicencio, Junichi Yamagishi, Nobutaka Ono, Isao Echizen, and Tomoko Matsui. Voice liveness detection algorithms based on pop noise caused by human breath for automatic speaker verification. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2015-Janua:239–243, 2015. `doi:10.1109/EMBC.2013.6610736`.

[133] Gwynne Shotwell. Spacex's plan to fly you across the globe in 30 minutes. TED2018, 2018. URL: `https://www.ted.com/talks/gwynne_shotwell_spacex_s_plan_to_fly_you_across_the_globe_in_30_minutes`.

[134] Simon Sinek. How great leaders inspire action. TEDxPuget Sound, 2009. URL: `https://www.ted.com/talks/simon_sinek_how_great_leaders_inspire_action`.

[135] Anthony Smith, Kristy de Salas, Ian Lewis, and Benjamin Schüz. Developing smartphone apps for behavioural studies: The alcorisk app case study. *Journal of Biomedical Informatics*, 72:108–119, 2017.

[136] J. O. Smith. Physical audio signal processing. [Accessed 2019-07-08]. URL: `https://ccrma.stanford.edu/~jos/pasp/Freeverb.html`.

[137] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust DNN embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 5329–5333. IEEE, 2018. `doi:10.1109/ICASSP.2018.8461375`.

[138] Brij Mohan Lal Srivastava, Natalia A. Tomashenko, Xin Wang, Emmanuel Vincent, Junichi Yamagishi, Mohamed Maouche, Aurélien Bellet, and Marc Tommasi. Design choices for x-vector based speaker anonymization. In Helen Meng, Bo Xu, and Thomas Fang Zheng, editors, *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 1713–1717. ISCA, 2020. `doi:10.21437/Interspeech.2020-2692`.

[139] Brij Mohan Lal Srivastava, Nathalie Vauquier, Md. Sahidullah, Aurélien Bellet, Marc Tommasi, and Emmanuel Vincent. Evaluating voice conversion-based privacy protection against informed attackers. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 2802–2806. IEEE, 2020. `doi:10.1109/ICASSP40776.2020.9053868`.

[140] Lifa Sun, Kun Li, Hao Wang, Shiyin Kang, and Helen Meng. Phonetic posteriorgrams for many-to-one voice conversion without parallel data training. In *Proceedings of the 2016 International Conference on Multimedia and Expo*, pages 1–6. IEEE, 2016.

[141] D. Sundermann and H. Ney. VTLN-based voice conversion. In *Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology (IEEE Cat. No.03EX795)*, pages 556–559, 2003. `doi:10.1109/ISSPIT.2003.1341181`.

[142] ML Blog Team. Now available: Speaker and video apis from microsoft project oxford. URL: `https://blogs.technet.microsoft.com/machinelearning/2015/12/14/now-available-speaker-video-apis-from-microsoft-project-oxford/`.

[143] Tomoki Toda, Ling Hui Chen, Daisuke Saito, Fernando Villavicencio, Mirjam Wester, Zhizheng Wu, and Junichi Yamagishi. The voice conversion challenge 2016. In *Proceedings of the Annual Conference of the International Speech Communication Association*, 2016.

[144] Massimiliano Todisco, Xin Wang, Ville Vestman, Md. Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas W. D. Evans, Tomi H. Kinnunen, and Kong Aik Lee. Asvspoof 2019: Future horizons in spoofed and fake audio detection. In Gernot Kubin and Zdravko Kacic, editors, *Interspeech*

2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019, pages 1008–1012. ISCA, 2019. `doi:10.21437/Interspeech.2019-2249`.

[145] Francis Tom, Mohit Jain, and Prasenjit Dey. End-to-end audio replay attack detection using deep convolutional networks with attention. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2018-September(September):681–685, 2018. `doi:10.21437/Interspeech.2018-2279`.

[146] Natalia Tomashenko. The voiceprivacy 2020 challenge - challenge setup and results. Odyssey 2020, 2020. URL: `https://www.voiceprivacychallenge.org/docs/1___VoicePrivacy_challenge_setup_and_results_N_Tomashenko.pdf`.

[147] Natalia Tomashenko, Brij Mohan Lal Srivastava, Xin Wang, Emmanuel Vincent, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Jose Patino, Jean-François Bonastre, Paul-Gauthier Noé, and Massimiliano Todisco. Introducing the VoicePrivacy initiative. 2020.

[148] Natalia Tomashenko, Brij Mohan Lal Srivastava, Xin Wang, Emmanuel Vincent, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Jose Patino, Jean-François Bonastre, Paul-Gauthier Noé, and Massimiliano Todisco. The VoicePrivacy 2020 Challenge evaluation plan. 2020. URL: `https://www.voiceprivacychallenge.org/docs/VoicePrivacy_2020_Eval_Plan_v1_3.pdf`.

[149] Henry Turner, Simon Eberz, and Ivan Martinovic. Recurring turking: Conducting daily task studies on mechanical turk. *arXiv preprint arXiv:2104.12675*, 2021.

[150] Henry Turner, Giulio Lovisotto, Simon Eberz, and Ivan Martinovic. I'm hearing (different) voices: Anonymous voices to protect user privacy, 2022. `arXiv:2202.06278`.

[151] Henry Turner, Giulio Lovisotto, and Ivan Martinovic. Attacking speaker recognition systems with phoneme morphing. In *European Symposium on Research in Computer Security*, pages 471–492. Springer, 2019.

[152] Henry Turner, Giulio Lovisotto, and Ivan Martinovic. Speaker anonymization with distribution-preserving x-vector generation for the voiceprivacy challenge 2020. *arXiv preprint arXiv:2010.13457*, 2020.

[153] Henry Turner, Giulio Lovisotto, and Ivan Martinovic. Generating identities with mixture models for speaker anonymization. *Computer Speech & Language*, 72:101318, 2022. URL: `https://www.sciencedirect.com/science/article/pii/S0885230821001133`, `doi:https://doi.org/10.1016/j.csl.2021.101318`.

[154] Vodafone UK. Vodafone voice id. URL: `https://www.vodafone.co.uk/explore/voiceid/`.

[155] Tavish Vaidya, Yuankai Zhang, Micah Sherr, and Clay Shields. Cocaine noodles: exploiting the gap between human and machine speech recognition. In *Proceedings of the 9th USENIX Workshop on Offensive Technologies*, 2015.

[156] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. pages 1–15, 2016. URL: `http://arxiv.org/abs/1609.03499`, `arXiv:1609.03499`.

[157] C. Veaux, J. Yamagishi, and Kirsten Macdonald. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. 2017.

[158] Voxforge Dataset. Free speech... recognition. [Accessed 2019-07-08]. URL: `http://www.voxforge.org/`.

[159] Robert Waldinger. What makes a good life? lessons from the longest study on happiness. TEDxBeaconStreet, 2015. URL: `https://www.ted.com/talks/robert_waldinger_what_makes_a_good_life_lessons_from_the_longest_study_on_happiness`.

[160] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez-Moreno. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 4879–4883. IEEE, 2018. `doi:10.1109/ICASSP.2018.8462665`.

[161] Qian Wang, Xiu Lin, Man Zhou, Yanjiao Chen, Cong Wang, Qi Li, and Xiangyang Luo. Voicepop: A pop noise based anti-spoofing system for voice authentication on smartphones. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2062–2070. IEEE, 2019.

[162] Ben Weinshel, Miranda Wei, Mainack Mondal, Euirim Choi, Shawn Shan, Claire Dolin, Michelle L. Mazurek, and Blase Ur. Oh, the places you've been! User reactions to longitudinal transparency about third-party web tracking and inferencing. *Proceedings of the ACM Conference on Computer and Communications Security*, pages 149–166, 2019. `doi:10.1145/3319535.3363200`.

[163] Meng-Han Wu and Alexander Quinn. Confusing the crowd: Task instruction quality on amazon mechanical turk. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 5, 2017.

[164] Zhizheng Wu, Tomi Kinnunen, Nicholas Evans, Junichi Yamagishi, Mirjam Wester, Ali Khodabakhsh, Cenk Demiroglu, Daisuke Saito, Tomoki Toda, Zhen-Hua Ling, and Others. Automatic speaker verification spoofing and countermeasures challenge (asvspoof 2015) database. *University of Edinburgh. The Centre for Speech Technology Research (CSTR), Tech. Rep*, 2015.

[165] Zhizheng Wu, Junichi Yamagishi, Tomi Kinnunen, Cemal Hanilçi, Mohammed Sahidullah, Aleksandr Sizov, Nicholas Evans, Massimiliano Todisco, and Héctor Delgado. ASVspoof: The automatic speaker verification spoofing and countermeasures challenge. *IEEE Journal on Selected Topics in Signal Processing*, 11(4):588–604, 2017. `doi:10.1109/JSTSP.2017.2671435`.

[166] W. Xiong, L. Wu, Fil Alleva, Jasha Droppo, Xuedong Huang, and Andreas Stolcke. The microsoft 2017 conversational speech recognition system. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 5934–5938. IEEE, 2018. `doi:10.1109/ICASSP.2018.8461870`.

[167] Wenhua Xu, Qianhua He, Yanxiong Li, and Tao Li. Cancelable voiceprint templates based on knowledge signatures. *Proceedings of the International Symposium on Electronic Commerce and Security, ISECS 2008*, (1):412–415, 2008. `doi:10.1109/ISECS.2008.100`.

[168] Junichi Yamagishi, Christophe Veaux, Kirsten MacDonald, et al. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92). 2019.

[169] Geng Yang, Shan Yang, Kai Liu, Peng Fang, Wei Chen, and Lei Xie. Multi-band melgan: Faster waveform generation for high-quality text-to-speech. In *IEEE Spoken Language Technology Workshop, SLT 2021, Shenzhen, China, January 19-22, 2021*, pages 492–498. IEEE, 2021. `doi:10.1109/SLT48900.2021.9383551`.

[170] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al. The htk book. *Cambridge university engineering department*, 3:175, 2002.

[171] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, XiaoFeng Wang, and Carl A Gunter. Commandersong: A systematic approach for practical adversarial voice recognition. In *Proceedings of the 27th USENIX Security Symposium*, pages 49–64, 2018.

[172] Heiga Zen, Rob Clark, Ron J. Weiss, Viet Dang, Ye Jia, Yonghui Wu, Yu Zhang, and Zhifeng Chen. Libritts: A corpus derived from librispeech for text-to-speech. In *Interspeech*, 2019. URL: `https://arxiv.org/abs/1904.02882`.

[173] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech. In *Proc. Interspeech 2019*, pages 1526–1530, 2019. URL: `http://dx.doi.org/10.21437/Interspeech.2019-2441`, `doi:10.21437/Interspeech.2019-2441`.

[174] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. Dolphinattack: Inaudible voice commands. In *Proceedings of the 24th SIGSAC Conference on Computer and Communications Security*, pages 103–117. ACM, 2017.

[175] Linghan Zhang, Sheng Tan, and Jie Yang. Hearing Your Voice is Not Enough: An Articulatory Gesture Based Liveness Detection for Voice Authentication. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security - CCS '17*, pages 57–71, 2017.

[176] Linghan Zhang, Sheng Tan, Jie Yang, and Yingying Chen. Voicelive: A phoneme localization based liveness detection for voice authentication on smartphones. In *Proceedings of the 23rd SIGSAC Conference on Computer and Communications Security*, pages 1080–1091. ACM, 2016.

[177] Yangyong Zhang, Lei Xu, Abner Mendoza, Guangliang Yang, Phakpoom Chinprutthiwong, and Guofei Gu. Life after Speech Recognition: Fuzzing Semantic Misinterpretation for Voice Assistant Applications. (February), 2019. URL: `https://dx.doi.org/10.14722/ndss.2019.23525`, `doi:10.14722/ndss.2019.23525`.

[178] Jinyu Zuo, Nalini K. Ratha, and Jonathan H. Connell. Cancelable iris biometric. In *19th International Conference on Pattern Recognition (ICPR 2008), December 8-11, 2008, Tampa, Florida, USA*, pages 1–4. IEEE Computer Society, 2008. `doi:10.1109/ICPR.2008.4761886`.

Appendix A

Voice Morphing Collected Audio

A.1 Commands

Command data was sourced as both utterances that could be presented to systems in existence, as well as commands used specifically by the Azure Speaker recognition system for verification. The utterances recorded were as follows:

1. Hey Siri (Repeated 4 times)
2. Ok Google (Repeated 4 times)
3. What is the weather like?
4. What time is it?
5. Who am I?
6. How tall is the shard?
7. My voice is stronger than passwords (Repeated 4 times)
8. My password is not your business (Repeated 4 times)
9. Apple juice tastes funny after toothpaste (Repeated 4 times)
10. Houston we have had a problem (Repeated 4 times)
11. You can activate security system now (Repeated 4 times)
12. My voice is my password (Repeated 4 times)

A.2 Conference

Conference talk transcripts were obtained from popular TED talks. The transcripts were shortened, so that they contained approximately the first 6 minutes of a given talk. The transcripts were then split into individual utterances, with each utterance being recorded as a separate audio file by the participant. Five different conference talk transcripts were used, which are the following:

1. Do schools kill creativity? by Sir Ken Robinson [123]
2. Your body language may shape who you are by Amy Cuddy [33]
3. What makes a good life? by Robert Waldinger [159]
4. How great leaders inspire action by Simon Sinek [134]
5. The power of vulnerability by Brené Brown [20]

A.3 Cafe

Our conversation audio is derived from TED talks where two people are having a conversation. A single speakers audio was extracted from each transcript, and the transcript was shortened until it was approximately 6 minutes in length. Five different conversation transcripts were used, which were derived from the following talks:

1. SpaceX's plan to fly you across the globe in 20 minutes - Gwynne Shotwell [133]
2. How Netflix changed entertainment - Reed Hastings [58]
3. Mammoths resurrected, geoengineering and other thoughts from a futurist - Stewart Brand [19]
4. The future we're building and boring - Elon Musk [105]
5. What everyday citizens can do to claim power on the internet - Fadi Cehadé [25]

A.4 Enrolment

Enrolment audio was used to enrol individual speakers with the Azure Speaker Recognition API for identification. Participants were asked to read the first 6 paragraphs of the speech given by UK Prime Minister David Cameron at the start of the London 2012 Olympics. The speech can be found on the UK government speeches website [21].

Appendix B

GMM entropy estimators

As shown in Section 4.3.4.2, estimating the GMM entropy is useful in order to best choose the parameters in a system development phase. While in Chapter 4 we directly estimate entropy using the log-likelihood of generated samples (Equation 4.3), in presence of large datasets or in online applications it might be useful to have faster estimators of entropy. Here we report the GMM entropy estimators introduced in [77], which we show generally provide very tight entropy bounds:

$$\hat{H}(X) = H(X|C) - \sum_i \pi_i \log \sum_j \pi_j \exp(-D(p_i||p_h)),$$

$$H(X|C) = \frac{1}{2} \sum_i \pi_i [\log |\Sigma_i| + d \log 2\pi + d],$$

for the lower bound we replace D with the Chernoff α-divergence distance function $C_\alpha(p_1||p_2)$:

$$\begin{aligned} C_\alpha(p_1||p_2) = \frac{(1-\alpha)\alpha}{2} (\mu_1 - \mu_2)^T ((1-\alpha)\Sigma_1 + \alpha\Sigma_2)^{-1} (\mu_1 - \mu_2) \\ + \frac{1}{2} \ln \Big(\frac{|(1-\alpha)\Sigma_1 + \alpha\Sigma_2)|}{|\Sigma_1|^{1-\alpha}|\Sigma_2|^\alpha} \Big), \end{aligned}$$

for the upper bound, we replace D with the Kullback-Leibler divergence $\mathrm{KL}(p_1||p_2)$:

$$\begin{aligned} \mathrm{KL}(p_1||p_2) = \Big[\ln \frac{1}{2} |\Sigma_2| - \ln |\Sigma_1| + (\mu_1 - \mu_2)^T \Sigma_1^{-1} (\mu_1 - \mu_2) \\ + \mathrm{tr}\Big(\Sigma_2^{-1} \Sigma_1 \Big) - d \Big]. \end{aligned}$$

In all equations, d indicates the Gaussian's dimensionality, α is set to 0.5, Σ_i is the co-variance matrix of the i-th component, μ_i are the means of the i-th component.

Appendix C

MOS Statistical Significance Tests

System One	System Two	U Statistic	p-value
Original	Baseline	75224.5	< 0.0001
Original	Proposed System	50757.5	< 0.0001
Original	Proposed System w/ $\theta_{FD} = 0.9$	46916.5	< 0.0001
Baseline	Proposed System	315373.0	< 0.0001
Baseline	Proposed System w/ $\theta_{FD} = 0.9$	320107.5	< 0.0001
Proposed System	Proposed System w/ $\theta_{FD} = 0.9$	431770.0	0.432

Table C.1: Mann-Whitney U Test Statistics and p-values for MOS scores between each pair of systems evaluated.

We perform statistical significant testing following the method outline in [126] on the mean opinion scores for the produced anonymized audio in Chapter 4. We use normalised-rank normalisation to first normalise the scores by participant and then by utterance, to correct for participant bias and utterance bias respectively.

We then use these scores to conduct a Mann-Whitney U test, between each pair of audio sets. In all cases the null hypothesis is that the distributions are equal to one another. Table C.1 shows the U statistic and p-value for each of the pairwise sets. We see that the differences are statistically significant for all pairs, except the Proposed System with and without Forced Distancing.