OBSCURE ATTACKS ON SPEECH AND SPEAKER RECOGNITION SYSTEMS

Jaipal Reddy Attuluri
Clemson University

Kota Jagathi
Clemson University

Thota Sahaja
Clemson University

1 ABSTRACT

The technology of voice recognition is a piece of hardware or software that can decode human speech. This technology, often known as voice-activated or speech-recognition software, is starting to get used to regular users. People frequently utilize these gadgets to conveniently fulfill requests, obtain data, or record sounds in various spots all over their homes. It would help if you learned more about voice-activated technology before allowing it into your life, home, or business because it is becoming more common and readily available. Recently, the security community found various flaws in specialized machine learning models that support numerous VPSes. These flaws enable an attacker to insert commands that are undecipherable by humans yet are accurately transcribed by VPSes.

This paper focuses on using Fast Fourier Transform feature extraction algorithms on multiple audio samples that have similar features. The paper uses three perturbation techniques that take an audio sample and a parameter to generate an attack. Techniques for perturbation include Random Phase Generation, Time Domain Inversion,and Time Scaling. This paper demonstrates the effectiveness of different signal models and real systems by testing the modified generated audio samples on multiple hardware configurations. These techniques can be used to generate hidden voice attacks successfully.

2 INTRODUCTION

Artificial intelligence and machine learning are used in speech and speaker recognition systems to identify and comprehend human voices. While these systems offer many potential uses, such as voice-controlled gadgets and contact center automation, they also present various challenges. Speaker identification systems may have difficulty recognizing speech [15] in noisy environments or identifying dialects and accents accurately. Additionally, voice and speaker identification technologies are vulnerable to security breaches and adversarial examples.

The expansion of voice-enabled products and services has brought a technological breakthrough in our daily lives. However, the rise of voice technology has also created new security concerns. For example, attackers can use synthesized speech or voice technology to simulate someone else's voice and circumvent security precautions that rely on speaker recognition. Furthermore, attackers can exploit automatic speech recognition [13] and speaker identification systems [9] vulnerabilities to compromise their accuracy and dependability.

Voice recognition technology is used increasingly in various industries, including healthcare, finance, security, and personal assistants. However, attackers can also exploit flaws in motion sensors [3] in smartphones to eavesdrop on private conversations and obtain confidential information. Moreover, hidden audio channel attacks [4] that transmit signals to a voice assistant via inaudible sound waves use to run malicious commands or access personal information.

Audio-visual speech recognition [2] combines audio and visual data to improve speech recognition systems' accuracy. However, these systems are susceptible to adversarial attacks [14], where attackers manipulate audio or visual input to deceive the system. It's essential to be proactive about security and regularly conduct vulnerability assessments, monitor systems for unusual activity, and implement multi-layered security solutions that address known and unknown threats. Businesses should be proactive and well-informed while defending against security risks. Keeping up with security trends and research can make it easier to spot emerging threats and develop strong responses.

Speech-to-text conversion having an exceptionally high degree of accuracy is what ASR aims to accomplish. The

software converts audio input of a specific language into text output. ASR is a critical component of Internet of Things (IoT) systems, where voice assistants like Amazon Echo, Google Alexa, Apple Siri, and others are at the center of controlling smart devices. ASR [5] is frequently used for conferences, online meetings, and the captioning of live news broadcasts. Microsoft Azure Speech-to-Text, Google Cloud Speech-to-Text, Amazon Transcribe, and Dragon Naturally Speaking are well-known productized ASR services. For consumers to appropriately interact with smart gadgets, most Automatic Speech Recognition (ASR) services require some use of VPS.

Four key building components make up a modern voice processing system (VPS): audio sample, audio pre-processing, signal processing, and model inference. Each block is essential for transforming audio input into the intended output, which may be speaker identification or speech-to-text. Sound can enter the system directly or through a microphone in the first block, called audio sample or input as you seen in Fig 1. Adversarial assaults can manipulate this input. The second step, audio pre-processing, separates unnecessary partitions and extracts the necessary audio data. By understanding what is processed or filtered out, adversaries can avoid detection. The signal prepares for machine learning algorithm's analysis and inference in the third stage, signal processing. Defense depends on this phase to introduce detection measures that attackers aim to avoid. Model inference, the fourth block, relies on how the VPS is used. Each block in a VPS is essential to converting audio input into the output you want [**?**]. Each block has weaknesses and problems that must be fixed to avoid adversarial attacks and guarantee accurate and dependable voice processing.

We have used perturbation techniques like Time Domain Inversion, Random Phase Generation, and Time Scaling. A signal's initial waveform can be recreated using the time-domain inversion signal processing technique using the frequency domain representation. In audio signal processing, it is employed frequently to reverse the effects of frequency-domain processing like spectral filtering and compression. Time domain inversion aims to produce a waveform for the output that closely resembles the initially generated input waveform. Implementing this strategy can be challenging since it requires intricate mathematical procedures. In audio signal processing, random phase generation produces a randomized phase spectrum that may be utilized to recreate a signal with a specific spectral form. It entails applying an inverse Fourier transformation to a frequency domain signal to derive the time domain representation after randomizing the phase spectrum. Although the phase information in the resulting signal will be random, it will still have the desired spectral shape, which can be helpful in several audio applications like audio watermarking or audio scrambling. There are several ways to generate random phases, such as utilizing a pseudorandom number generator or modifying the phase spectrum with a random phase mask. A technique that can be used to avoid detection with a speech recognition system is a time scaling attack. A time scaling attack alters the spoken command's duration by accelerating it upwards or slowing it down. This may result in the control being interpreted as a different word or knowledge, or it can result in the system completely failing to recognize the command. If the attacker is familiar with the particular speech recognition system being utilized, they can modify their time-scaling attack to capitalize on any vulnerabilities or weaknesses in the system.

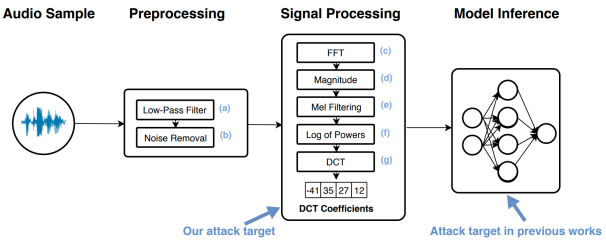

Figure 1: The various processing phases carried out on the audio before it is given to a Machine Learning Model in transcription are demonstrated through this generic VPS workflow in audio transcription [1]. To reduce noise, the audio is first preprocessed. The audio is then subjected to a signal processing algorithm, which modifies the input's format while keeping the crucial components and discarding the remainder. Finally, a model based on machine learning is given these features for inference.

3 PROBLEM STATEMENT

In recent years, breakthroughs in deep learning, neural networks, and machine learning have led to the development of numerous audio processing systems. Voice processing systems are among the latest devices produced by recent technological advancements and modifications that are employed all over the world. Mobile devices can control them by issuing commands that they can quickly carry out.

Voice processing systems, among the most often used equipment, are one of the causes of the insertion of covert commands with noise disruption. Automated

speech recognition (ASR) software is necessary for recognizing and differentiating voices.

Some ASR technologies ask users to train the machine to identify their voice to provide a speech-to-text conversion that is more accurate. This problem statement seeks to create audio attacks with disturbances humans cannot perceive. Still, it detects by a voice processing system these transcriptions of audio tests with machine learning models. The primary cause behind the injection of hidden commands with noise disturbance is voice processing systems.

Therefore, humans cannot understand such speeches or declarations. These systems are susceptible to various attacks, including Time Domain Inversion (TDI) and Random Phase Generation (RPG), Time-Scaling which can have profound ethical and security repercussions for people and organizations who rely on them. It is so important because they draw attention to the weaknesses of these technologies, which deploy in more and more applications. Understanding these threats will help researchers and developers work on enhancing these systems' security and trustworthiness, which is essential for protecting users' privacy and safety. We can accomplish the following three objectives if we achieve this problem statement.

4 RELATED WORK

Speech recognition and personal assistant systems frequently use machine learning models to find patterns and finish jobs quickly and effectively. But ML models are naturally open to a wide range of threats [13]. Early studies concentrated on improving train-time resilience for circumstances when the attacker may taint training data, leading to spam detection. In addition, the article emphasizes the possible effects of these assaults on real world applications that depend on ASR and SI systems, such speech identification, speaking transcribing, and voice-activated assistants.

The research article "Hidden Voice Commands" [4] examines the potential security dangers of speech recognition technologies and the ability of attackers to give these systems secret orders. The authors describe a technique for producing voice recognition that is undetectable to humans but can hear by speech recognition software, such as that used by well-known voice assistants such as Siri and Google Now. They show how personal assistants and other devices can operate using these covert commands without the user's knowledge or agreement. The authors discussed signal processing methods and algorithms for machine learning as possible countermeasures to these attacks. Ultimately, the article emphasizes the need for more investigation into the security of voice recognition systems.

As in the article Speech and Speaker Recognition System, a system for voice and speaker recognition is described that combines hidden Markov models and artificial neural networks (ANNs) (HMMs). The authors begin by briefly introducing speech and speaker identification [6] technology and its uses, such as voice-based access management and authentication. Next, they go over the system's architecture, which is divided into three phases: feature extraction, extraction of features, and classification. The effectiveness of the authors' approach is then assessed using a collection of voice samples from various speakers. The strengths and weaknesses of their suggested system are discussed in the paper's conclusion and potential future lines of inquiry for voice and speaker recognition research.

The paper "A review on speaker recognition: Technology and problems" gives an overview of speaker recognition technology and its present level of development [10]. The authors define speaker identification and discuss its uses, such as speech-based authentication, access control, and forensic analysis. The writers also go into the different difficulties with speaker identification, such as the impacts of noise as well as channel distortion and changes in speech associated with age, gender, and emotion. Additionally, they emphasize the significance of data gathering, preprocessing, and modeling methods for precise speaker detection. The paper concludes the paper by discussing speaker identification technology's current state and potential for advancement, including the application of deep learning methods.

By utilizing both auditory and visual data, the field of research known as audio-visual speech recognition (AVSR) [11] seeks to increase speech recognition accuracy. Deep learning has been essential in improving this discipline, with a range of models based on deep understanding being built and tested on AVSR tasks. Convolutional neural network, recursive neural networks, and a hybrid model that blend CNNs and RNNs are a few of the most often utilized models. According to recent studies, these models can perform at the cutting edge on AVSR tasks, demonstrating the possibilities of deep learning in this field.

The paper "Adversarial Attacks and Defenses in Speaker Recognition Systems: A Survey" describes the potential drawbacks of adversarial assaults on speaker identification systems and possible countermeasures.

The authors begin by outlining the operation of speaker recognition[8] systems before looking at the possible peril posed by adversarial assaults, which modify input data to trick a speaker recognition system and yield false results. The remainder of the study explores several adversarial attacks, such as waveform modification, noise or other alterations to the audio stream, impersonation, and synthesized speech. The authors also review various methodologies, including gradient-based and optimization-based strategies, for producing adversarial cases. The authors also look at some countermeasures that have been put out to lessen the effects of hostile attacks. These consist of techniques for spotting adversarial examples, building more robust models, and preprocessing input data to filter out adversarial perturbations. The study continues by discussing unresolved issues, such as how to develop defense mechanisms that are more effective and efficient and how to increase the resistance of speaker identification systems against a wider variety of adversary approaches. The paper generally provides a thorough overview of adversarial attacks defenses in speech identification systems.

An overview of adversarial example attacks against automated speech recognition (ASR) systems is given in the paper "Adversarial Example Attacks Against ASR Systems: An Overview." The authors describe the concept of adversarial[16] examples and how they trick machine learning models, particularly ASR systems, in the opening paragraphs. They discuss adversarial example attacks, such as those that alter audio transcriptions and add noise or distortion to audio signals, that can be employed against ASR systems. The remainder of the study examines several techniques, such as gradient-based, optimization-based, and evolutionary algorithms, for producing adversarial cases against ASR systems. In addition, the authors discuss several defense strategies that can be employed to shield ASR systems from attacks using adversarial examples, including adversarial training and input preprocessing. Following that, the authors present a review of current studies in adversarial attacks against ASR systems, highlighting the most important conclusions and future directions for research. They explore unresolved research issues and potential approaches for further study in this area before ending. The paper thoroughly summarizes adversarial example assaults against ASR systems and the state of the art of related research. It emphasizes how critical it is to provide robust defense mechanisms to safeguard ASR systems against such assaults to guarantee their dependability and security in practical applications.

The "High-Frequency Adversarial Defense for Speech and Audio" paper suggests a fresh strategy for protecting speech and audio systems from hostile assaults. The authors start by defining adversarial attacks and how they work to trick speech and audio systems. Then they present their suggested defense strategy focusing on high-frequency audio signal components [12]. To make it more difficult for adversarial attacks to modify the audio signal, the defense mechanism operates by filtering out high-frequency audio signal components that are unimportant for human perception. The authors demonstrate how this strategy can successfully lower the rate of various adversarial assaults on speech and audio systems. The defense mechanism is explained in more detail in the study, and the authors present experimental findings that show how successful it is in fending off aggressive attacks. Additionally, they demonstrate how their approach outperforms other cutting-edge defense mechanisms in terms of computational efficiency and defense effectiveness. The author's discussion of the approach's potential drawbacks and potential future research avenues in this field concludes. Overall, the study offers an innovative and successful method for protecting speech and audio systems from adversarial assaults, which has real-world applications for boosting the security and dependability of these systems.

5 BACKGROUND

5.1 Voice processing system

Any machine learning-based voice processing tool, such as Automatic Speech Recognition (ASR) and Speaker Identification models, is referred to as a VPS.

5.1.1 Automatic speech recognition The three basic processes in the Automatic Speech Recognition (ASR) process are pre-processing, signal processing, and model inference. Pre-processing entails filtering the audio to eliminate ambient noise and frequencies outside the audible range for humans. The most crucial audio properties are captured by signal processing algorithms, which build a feature vector using methods like Mel-Frequency Cepstrum Coefficients (MFCC). The model is then given this feature vector, either training or inferencing. ASR models are frequently employed to translate spoken language into text.

5.1.2 speaker Identification model Speaker identification models compare voice samples to identify the speaker in a recording. They employ a voting system for each audio subsample to determine which speaker is the most plausible. The speaker with the most votes is chosen because of the source of the input audio

sample after the complete audio file has been processed and processed. Speaker identification models' internal operations resemble ASRs in many ways. They use a feature vector as the model's input for inference or training and are trained on speaker speech samples.

5.1.3 signal processing Signal processing is essential for any Voice Processing System (VPSes) only to record the relevant aspects of the audio data. The success of training a machine learning model for VPSes depends on the signal processing algorithm's quality.

Mel-Frequency Cepstrum Coefficient (MFCC): The audio sample is initially divided into 20 ms windows before being processed to produce an MFCC vector. Figure 1 illustrates the four main processes that each window goes through.

Fast Fourier Transform (FFT) and Magnitude: Every window containing the audio is initially subjected to an FFT to obtain a frequency domain representation of the sound before the MFCC vector of the audio sample is obtained. A magnitude spectrum, which comprises details about the frequency components and their related intensities that make up the signal, is produced after computing the FFT's magnitude.

Mel Filtering: The Mel scale converts actual frequency differences into differences in frequencies that the human ear can hear. Mel filter banks convert frequency data onto the Mel scale employing overlapping triangular windows.

Logarithm of Powers: The energies for every Mel filter bank are then placed on a logarithmic scale to imitate how human hearing perceives loudness. **Discrete Cosine Transform (DCT):** Applying the discrete cosine transform of a list of Mel filter bank energies is the last step in acquiring the MFCCs. The end result is a vector of MFCCs, which stands for the coefficients that characterize the spectral envelope for the audio signal. Then, these coefficients can be employed as features by algorithms that use machine learning to carry out operations like speaker identification or speech recognition.

5.1.4 Other methods Different signal processing methods, including MFSC, Linear Predictive Coding, and PLP, are used by contemporary VPSes. These deterministic methods capture the most crucial features of the data. One model is trained to extract features, and the other is utilized for inference, using probabilistic approaches like transfer learning. In the most recent development of VPSes [7], intermediate modules among the raw input and model were eliminated. The "end-to-end" method seeks to streamline system implementation and speed up data processing.

5.1.5 Model Inference VPSes transmit signal processing features to machine learning algorithms via inference. Machine learning systems must use a training set to map inputs into outputs and reduce error. Modern systems use data-driven methodologies for feature extraction, whereas early systems need intensive feature engineering or domain expertise. These methods employ a predefined cost function and automatically pick up pertinent features to reduce mistakes. Modern systems offer improved extrapolation performance, are more adaptable, and can be divided into separate modules. Transfer learning makes learning to reuse existing modules across several applications or domains possible.

5.1.6 Psychoacoustics

The study of how people interpret and experience sound is known as psychoacoustics. It is a broad area of study that covers a variety of subjects, including the cognitive processes involved in speech recognition, the physiological mechanisms of hearing, and the psychological impacts of sound on human behavior. As a result, it's essential to comprehend psychoacoustics to create voice recognition systems that consider the subtleties and complexity of human hearing. Human hearing has both strong and weak points, according to psychoacoustics research. The Cocktail Party effect, for instance, describes how easily individuals can concentrate on a single source of sound when numerous sources are present. Human hearing can be limited, especially at higher frequencies, since we tend to hear louder sounds at higher frequencies. Humans also have difficulty understanding random or discontinuous sounds, which most people find startling and unsettling. Understanding psychoacoustics enables the modulation of voice commands to maximize comprehension and perturbation while creating speech recognition systems. The advantages and disadvantages of human hearing can be considered while creating voice recognition systems, which will help them function better. Speech recognition systems can boost accuracy and usability, for instance, by optimizing the frequency of voice instructions and lowering the effect of noise on its intelligibility.

6 METHODOLOGY

In the research conducted, the focus was on perturbation attacks targeting Voice Processing Systems (VPS). The goal was to create attack audio that could successfully trick the VPS into carrying out instructions that the user couldn't understand. In order to generate test audio samples, the study required analyzing several attack scenarios and perturbation approaches.

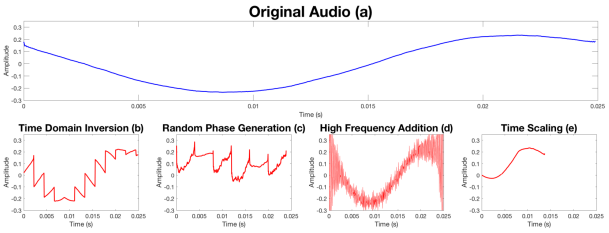

Figure 2: The aforementioned figure displays the various perturbation methods used on the initial signal (a). Signals (b) through (e) demonstrate the outcome of using the perturbation techniques on signal (a).

The process involves feeding the algorithm with an audio sample and a set of parameters. Next, attack audio samples were made using perturbation techniques, particularly the Random Phase Generation and Time Domain Inversion Attack methods. The efficiency of the voice assistant system was then evaluated using these samples, which were subsequently given to the VPS as.wav files and played through speakers.

6.1 Attack Scenario

The attack scenario involved an attacker attempting to use a Home Assistant or VPS to carry out a malicious command. The attacker would play masked audio near the target device from a specified distance in order to do this. With automated speech recognition (ASR) capabilities, the VPS or Home Assistant would interpret the command and carry it out as necessary. Before performing the instruction, the VPS may also use a speaker identification framework to confirm the speaker's identity.

The attack's garbled audio may have included numerous instructions, like how to access banking data or open doors, that were intended to activate VPS features. Without the speaker's owner or personal assistant being aware of the attack, the audio clips might be played through any speaker or device close to the targeted VPS. Three different forms of perturbations were used in this study. Each instance that results has one or more perturbations intended to harm the victim's VPS.

6.2 Perturbation Techniques

Time Domain Inversion (TDI): In order to obscure the content of a particular audio signal, the perturbation technique known as Time Domain Inversion (TDI) is employed in audio signal processing. It operates by splitting an audio signal into identically sized windows or frames and flipping the positions of each window. The final signal is created by concatenating the windows once again.

To encode a concealed message or to make an audio signal more resilient to specific types of attacks, such as audio fingerprinting, TDI can be used as an audio watermarking technique. In order to conceal a confidential message within an audio transmission, it can also be utilized as a type of audio steganography.

Speech recognition, audio signal manipulation, and audio processing can all be done using the provided code, which combines functions and scripts. Below is a list of all the code sections:

1. Bringing in Libraries The required libraries for signal manipulation and audio processing are imported in the first portion of the code. The following libraries:
speech recognition: for speech recognition
os, os. path, wave: for file management
numpy, scipy, librosa: for numerical processing and Audio signal manipulation
sklearn.decomposition: for Principal Component Analysis (PCA) **pandas:** for data analysis
time: for time measurement
itertools: for iteration tools

2. Audio transcription A function called "transcribe()" is included in the second portion of the code and is used to convert the Audio file into text. The process accepts the transcription model and the Audio file path as inputs. The model argument determines the transcribing service to be utilized. Only the Google transcription service is supported in this instance.

```
result2:
{   'alternative': [   {   'confidence': 0.73109657,
                           'transcript': 'call Mom call Mom'},
                       {'transcript': 'call Mom call Mom call Mom'},
                       {'transcript': 'call to Mom call Mom call Mom'},
                       {'transcript': 'call Mom'},
                       {'transcript': 'call Mom call to Mom'}],
    'final': True}
Out[10]: 'call Mom call Mom'
```

Figure 3: Results Of Time Domain Inversion

```
data2 = np.asanyarray(1)

#Write back the file in proper form
scipy.io.wavfile.write("/Users/sahajathota/Downloads/SEN1/PRO/Code/Audio3.wav", fs, data2)
transcribe("/Users/sahajathota/Downloads/SEN1/PRO/Code/Audio3.wav", "google")
t[13]: 'text Mom Mom Mom'
```

Figure 4: Results Of Time Domain Inversion

3. Manipulation of Audio Signals The third component of the code uses the Time Domain Inversion (TDI)

technique to alter the Audio signal. This method generates a new Audio file by flipping the order of a group of Audio samples within each time window. The path to the source audio file, the number of samples to invert, and the sampling rate are all inputs to the code.

The scipy library is used to initially read the Audio file, after which the code calculates the number of samples for each time frame based on the window size. The Audio file is then divided into temporal windows, and each window is given the TDI treatment. Using the Scipy package, the inverted samples are concatenated and written back to a new Audio file.

4. Testing the Audio Signal Manipulation and Transcription The final piece of the code loads an Audio file and uses the TDI technique to test the transcription and Audio signal manipulation functionalities. When the TDI technique is applied to the file, the order of the 25 samples in each time window is reversed. The newly created audio file is then used to transcribe text using Google's transcription service. The expected output would be "call mom call mom," (Figure 3) as that is the content of the original Audio file. Yet, it looks like the TDIAttack function altered the Audio file by flipping part of the array's members, which would cause the Audio to sound distorted. Moreover, the output reads, "text mom mom mom." (Figure 4)

Random Phase Generation: A random phase generation attack (RPGA) is a type of security vulnerability that can affect certain cryptographic systems, particularly those based on the discrete Fourier transform (DFT). An attacker can take advantage of flaws in an RPGA's random phase value generation process to decrypt data.

Let's first think about the fundamentals of encryption using the DFT in order to comprehend RPGA. Before performing further encryption operations in some cryptographic systems, the DFT is used to convert the plaintext into the frequency domain. Random phase values are frequently added during this transformation to mask the original signal and increase the encryption's security.

The code we provided is a combination of two different sections. The Singular Spectrum Analysis (SSA) functions for time series decomposition and prediction are included in the first section. The second section consists of unrelated code that relates to audio analysis and signal processing. The first and second sections of the code combine several time series analysis and single spectrum analysis (SSA)-related functions and scripts. The code is broken down as follows:

1. The code begins with some import statements for essential libraries, including scipy, numpy, and matplotlib. Next, the availability of the matplotlib library is checked.

2. The code defines a number of useful functions, including 'isscalar', 'nans','ssa', 'inv ssa','ssa predict','ssa cutoff order', and'ssaview'. The next sections of the code make use of these functions.

3. The code sets the sampling rate and data to the variables 'fs' and 'data', respectively, then imports an audio file using'scipy.io.wavfile.read()'.

4. The code defines a number of signal processing-related methods, including 'getSignalSum', 'getMagnitude', and'signalBound'. These functions calculate signal magnitudes and signals the [Figure 5] shows the half-magnitude frequency of the audio generated by random phase generation attack.

5. The code generates all lists of positive integers that add up to the value 'n' by using a generator function called'sum to n'.

6. The code then duplicates some of the definitions of earlier created functions and variables from the previous code section.

7. The code then assigns the sample rate and data to the variables 'fs' and 'data' before reading an audio file once more using'scipy.io.wavfile.read()'. The data array's length is printed.

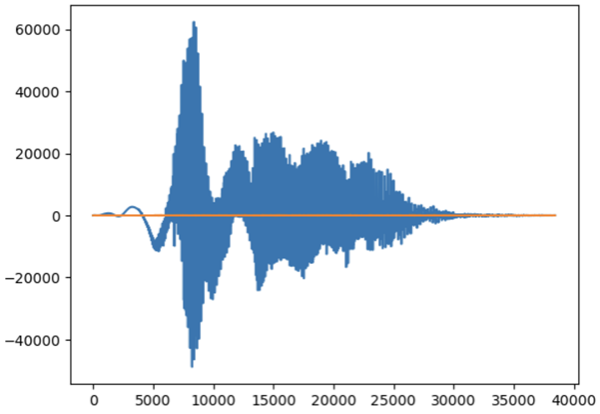

Figure 5: This figure shows the half-magnitude frequency of the audio produced by Random Phase Generation Attack.

Time Scaling: Time scaling, when used in relation to audio processing, describes the alteration of audio signals in order to change the audio's tempo or speed without altering its pitch. It entails shrinking or stretching the audio waveform's time axis, hence changing the audio's duration. This adjustment can be accomplished using time scaling techniques like time stretching and resampling. Time stretching allows for faster or slower playback without disturbing the pitch by adjusting the sample rate while preserving the frequency content. By alter-

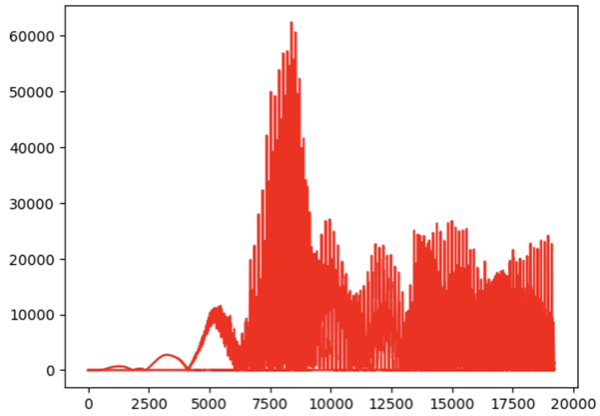

Figure 6: This figure shows the full amplitude frequency of the audio produced by the Random Phase Generation Attack

ing the audio's sample rate, resampling produces compressed or extended audio length. In order to change the speed of instructions or increase comprehensibility, time scaling can be utilized in a variety of applications, including voice recognition and audio editing. However, it's crucial to carefully weigh the trade-offs between preserving audio clarity and quality and obtaining the necessary speed modification. The provided code looks to be a synthesis of different audio processing, voice recognition, and data manipulation tools and modules. Here is a list of the code's functions:

1. Importing Libraries: The code imports a number of libraries, including speech recognition, os, wave, scipy, librosa, numpy, PIL, matplotlib, and others. These libraries offer features for speech detection, file manipulation, audio processing, and visualization of data.

2. Reading an audio file: The scipy.io.wavfile.read() function is used in the code to read an audio file. /Users/sahajathota/Downloads/SEN1/PRO/MID/Code/phva demo.wav is the location to the audio file. The variables fs and 'data' respectively store the sampling rate and audio data.

3. voice Transcription: Using several voice recognition models, the 'transcribe()' function in the code does speech transcription. For transcription, it currently supports the Google Speech-to-Text API. The function accepts as input parameters the desired recognition model and the path of the audio file. It converts the audio to text using the "speech recognition" package and then returns the text the [Figure 9,10] shows the input audio file message and the output that was faked with help of time scaling.

4. Changing Audio: The code lengthens the audio data array by appending zeros. The original audio data is kept in the 'new' variable, while the appended zeros are kept

```
This is magFreq:
 [[  2.   0.]
 [  2.   0.]
 [  2.   0.]
 ...
 [379.  13.]
 [143.   3.]
 [273.  11.]]

 [[   2.-0.j    0.-0.j]
 [   2.-0.j    0.-0.j]
 [   2.-0.j    0.-0.j]
 ...
 [-379.-0.j  -13.-0.j]
 [-143.-0.j   -3.-0.j]
 [ 273.-0.j   11.-0.j]]
5.0
[[ 2.-0.j  0.-0.j]
 [ 2.-0.j  0.-0.j]
 [ 2.-0.j  0.-0.j]
 [ 2.-0.j  0.-0.j]
 [ 4.-0.j  0.-0.j]
 [ 4.-0.j  0.-0.j]
 [ 4.-0.j  0.-0.j]
 [ 6.-0.j  0.-0.j]
 [ 6.-0.j  0.-0.j]
 [ 7.-0.j -1.-0.j]]
```

Figure 7: This figure shows the audio generated by a random phase generation attack's magnitude frequency.

in the 'zeros' variable. This operation makes use of the NumPy library's np.append()' and np.asarray() capabilities. Using scipy.io.wavfile.write(), the changed audio data is then written back to the original audio file path.

5. Speech Transcription (Again): To execute speech transcription on the updated audio, the 'transcribe()'

```
[array([3.60555128, 3.46410162]),
 array([3.74165739, 3.31662479]),
 array([3.87298335, 3.16227766]),
 array([4., 3.]),
 array([4.12310563, 2.82842712]),
 array([4.24264069, 2.64575131]),
 array([4.35889894, 2.44948974]),
 array([4.47213595, 2.23606798]),
 array([4.58257569, 2.        ]),
 array([4.69041576, 1.73205081]),
 array([4.79583152, 1.41421356]),
 array([4.89897949, 1.        ]),
 array([3.60555128, 3.46410162]),
 array([3.60555128, 3.46410162]),
 array([3.60555128, 3.46410162]),
 array([3.60555128, 3.46410162]),
 array([3.60555128, 3.46410162])]
```

Figure 8: This figure shows the sounds generated by a Random Phase Generation Attack's Signal Sum.

```
result2:
{   'alternative': [   {   'confidence': 0.73751611,
                           'transcript': 'call Mom Mom Mom'},
                       {'transcript': 'call Mom call Mom'},
                       {'transcript': 'mom mom mom'},
                       {'transcript': 'call mom-mom'},
                       {'transcript': 'call a mom mom mom'}],
    'final': True}

'call Mom Mom Mom'
```

Figure 9: This figure shows the input of Time scaling attack

```
result2:
call Cat Cat Cat
```

Figure 10: This figure shows the result of Time scaling attack.

function is called once more with the modified audio file path. The function returns the text that was transcription.

7 Experimental Setup and Evaluation

7.1 Experimental Setup

Our project relies heavily on audio generation, which we accomplished by using machine learning techniques. To create the perturbed audio, a mixture of signal processing steps and feature extraction was used. Our main attention was on the signal processing phase, when we added noise to the audio's background.

We deliberately chose various phrase kinds to create the perturbed audio in order to ensure the diversity of our attack samples. To maintain the originality and potency of our attacks, we purposefully refrained from utilizing terms like "ok Google."

The resulting perturbed audio was then stored as an input file in the.wav extension. The perturbation assaults were then given access to these audio files as input for additional processing. Various strategies were used during this attack phase, including reducing the window size, developing the magnitude spectrum, and including high frequencies outside of the typical range.

overall, our experimental setup included perturbing audio using machine learning, concentrating on signal processing, choosing a variety of phrases, executing perturbation assaults, and assessing the transcriptions using the Google Speech API.

7.2 Evaluation

In our analysis, we attacked both speaker identification models and Automated Speech Recognition (ASR) systems. Using the perturbation engine, we created attack audio samples for the ASR attacks and tested them by sending them as.wav files to the ASR system. The ASR system's ability to accurately translate the sentence from the attack audio, regardless of any transcription mistakes, was the criterion for success for these attacks. The ASR system correctly identified a sample audio that we provided, demonstrating the effectiveness of the assault. We also made notice of the fact that contemporary ASR systems may be made to mimic earlier attacks, enabling attackers to reuse the attack audio repeatedly across several ASR systems.

Our attacks on the speaker identification models were equally successful. It was intended to determine whether the identification model correctly identified the attack audio as coming from the original audio source. The fact that we got the outcomes we anticipated proves that the attacks were successful. It is crucial to remember that the speaker models did not provide any information regarding the success of the audio, which means that hackers cannot adapt their attacks in response to the model's input. Because of this, we advise adopting the offline perturbation method described in our study to frame the data for evaluation.

Overall, our research shows how well the perturbation engine can produce attack audio samples that can successfully fool both speaker identification models and ASR systems. These results highlight the need for strong defenses against such attacks in order to improve the security and dependability of speaker identification

and speech recognition systems.

7.3 Positive Outcomes

We made substantial discoveries regarding the perturbation engine's parameters during our research that can greatly shorten the time it takes to generate an attack. We specifically discovered three key characteristics connected to the window size parameter:
1. Smaller window sizes are correlated with larger auditory problems.
2. If an assault audio sample can be read correctly at a particular window size, then bigger window sizes will also produce accurate results.
3. The transcription of audio samples with window sizes less than 1.0 ms needs to be fixed.

By starting with a window size of 1.00 ms and increasing it until they acquire the first properly transcribed assault audio, attackers can now optimize the process of creating attack audio. Attackers can minimize the number of attack audio clips they need to create by combining these three elements.

We also learned that the amount an audio file can be changed before the VPS loses the ability to recognize it. Faster perturbations are made possible by expanding the window size, but there is a point beyond which the audio becomes inaudible. Attackers can use this information to modify additional perturbation factors, like speed, and reduce the likelihood that the VPS will detect audio. The efficiency of the attack is increased by tailoring the attack audio to the target and the particular attack circumstances.

We found that the success rates of the perturbation strategies varied. In our experiments, RPG (Random Phase Generator) outperformed Time Domain Inversion (TDI). RPG includes conducting an inverse FFT after applying the Fast Fourier Transform (FFT) to the audio. However, because the model is a "black box," we were unable to identify the precise FFT method or its operating parameters.

Furthermore, our tests demonstrated that the TDI perturbation is not dependent on the discontinuity matching the timescales of the feature extraction algorithm in the black-box models. Given that we don't know much about the time frame discontinuities in black-box models, this observation supports the assumption that perturbation tactics can be successful against them.

We were able to successfully reproduce the findings for methods like Time Domain Inversion, Random Phase Generator, and Time Scaling that were presented in the selected research paper. Due to our lack of resources in comparison to the researchers, we were unable to collect complete data for the Time Scaling technique.

8 SURVEY

This survey aims to gather information and insight about hidden voice attacks and their relevance to speech and speaker recognition systems. Hidden voice attack refers to the deliberate manipulation of audio signals to deceive or manipulate these systems. This survey emphasizes how crucial it is to have security precautions in place to shield these systems from flaws and unwanted activity. Security, finance, and voice assistants use systems for recognizing speakers and speech. These systems process human voices for interactions and identification using cutting-edge algorithms. However, hidden voice attacks take advantage of flaws in these systems to trick authentication, get around security precautions, and access private data without authorization.
By conducting this survey, we seek to gather insights from participants about their awareness of hidden voice attacks and their potential impact. By protecting user privacy, data security, and system integrity, these details will help to better understand the necessity for security measures to safeguard voice and speaker recognition systems.

8.1 Methodology

The survey was created using a multiple-choice questionnaire that could be completed online. It was disseminated via social media and email invitations to reach a wide spectrum of people. Target Audience and Inclusion Criteria: The survey is aimed at individuals who have experience or knowledge of speech and speaker recognition systems. To guarantee a diverse sampling of responders, there were no explicit inclusion requirements regarding age, career, or geography. Timeframe: The survey was carried out from [04-20-2023] to [05-01-2023] during a two-week period. Participants could complete the survey at their convenience time period.

8.2 Survey Questions and Responses

A. Demographic Information
The demographic features of the participants are intended to be gathered through these questions. In terms of age and gender, the survey gathered responses from a varied set of respondents. The majority of participants were between the ages of 21-24 comprising 32 individuals. This was closely followed by the 17-20 age group, with 9 participants. Additionally, there were 4 participants each from the 25-30, 4 participants above 30 age groups and 1 participant from 14-16 age group the

[Figure 11] clearly depicts the participant's age group ratio. The survey featured a higher percentage of male participants comprising of 33 male participants and 17 female participants when it came to gender distribution the [Figure 12] clearly depicts the participant's gender ratio.

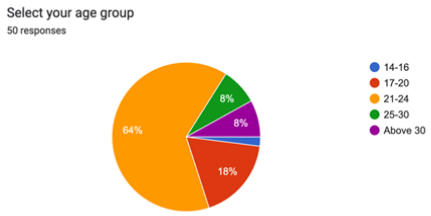

Figure 11: This figure illustrates the age group.

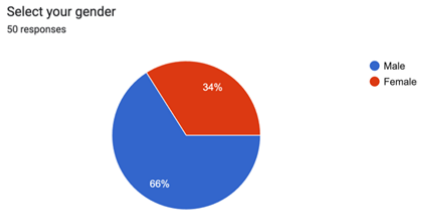

Figure 12: This figure illustrates the gender.

B. Experience with Speech and Speaker Recognition
These questions aim to gather information about their experience with speech and speaker recognition systems. The study found that a sizable percentage of respondents had used speaker or speech recognition systems either privately or professionally. 46 of the 50 respondents acknowledged using speaker or speech recognition devices the [Figure 13,14] clearly depicts the participant's Experience with Speech and Speaker Recognition. With 34 mentions, Alexa was the home assistant or voice processing system that participants were most likely to own.

These results show that speaker and speech recognition systems have been widely adopted by survey respondents, with Alexa being the most preferred option. The information gathered from this area offers important insights about how familiar and how often users use various tools.

C. Usage and Reliability
The survey revealed that among the 50 participants, the majority 31 use Home Assistant, Voice Processing, or Speaker Recognition devices very frequently to issue

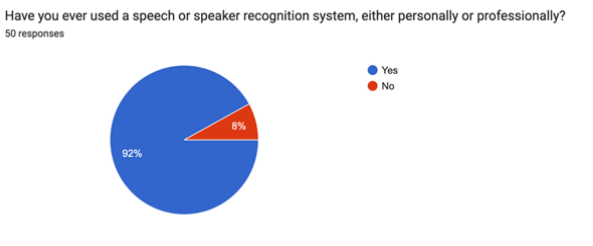

Figure 13: This figure illustrates Experience with Speech and Speaker Recognition.

Figure 14: This figure shows what kind of home assistant participants use.

commands, indicating a high level of engagement.While some respondents reported frequent or moderate usage, a few mentioned rare or no usage at all. In terms of reliability, participants generally expressed satisfaction, with "Fair Enough" being the most common response by 33 participants, suggesting reliable command execution. Some participants even mentioned higher reliability levels, such as "Accurate" or "More Accurate." However, a few participants reported low reliability or indicated their devices were not reliable at all. These findings highlight variations in usage frequency and perceived reliability among participants. the [Figure 15,16] clearly depicts the participant's home assistant usage, and reliability.

D. Hidden Voice Attacks
The survey revealed that a majority of the participants (40 out of 50) had encountered unidentified commands being carried out by their Home Assistant, Voice Processing, or Speaker Recognition systems, indicating potential vulnerabilities. Furthermore, 34 participants were aware of hidden voice attacks against speech and speaker recognition systems, demonstrating a general under-

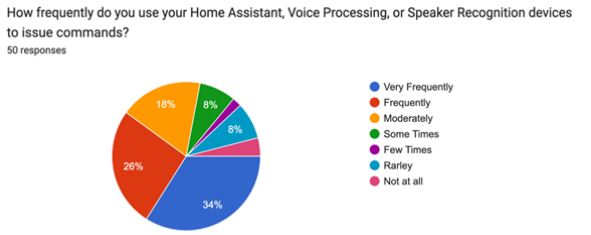

Figure 15: This figure illustrates participant's home assistant usage.

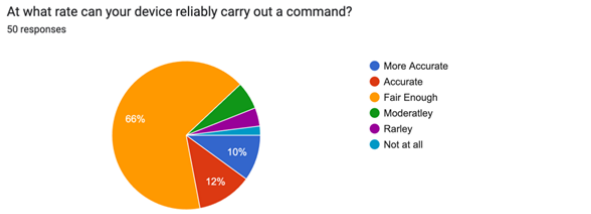

Figure 16: This figure illustrates participants home assistant Reliability.

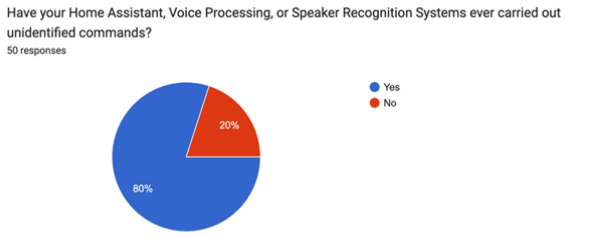

Figure 17: This figure shows any unidentified commands carried out

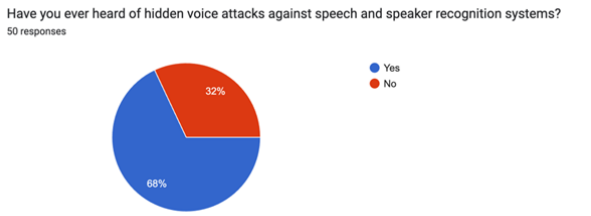

Figure 18: This figure illustrates participants understanding on hidden voice attacks.

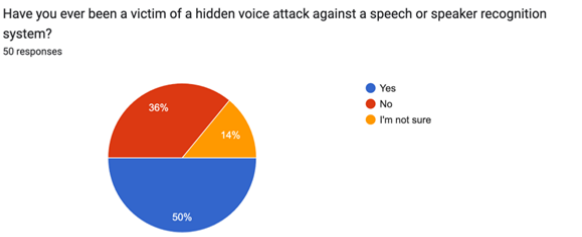

Figure 19: This figure illustrates any participant's are victim of hidden voice attacks.

standing of this security threat the [Figure 17,18,19] clearly depicts participants understanding over hidden voice atatcks, and any victims. Although only a small number of respondents (25) claimed to have personally experienced a hidden voice attack, these findings underscore the importance of addressing and mitigating the risks associated with hidden voice attacks to enhance the overall security of these systems.

E. Concerns and Security Measures

The survey results indicate that participants are highly concerned about the security of speech and speaker recognition systems in relation to hidden voice attacks. The majority expressed a belief that more should be done to protect these systems against such attacks.

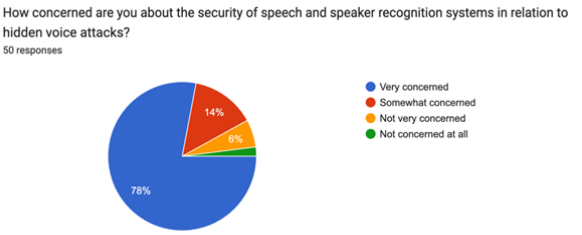

Figure 20: This figure shows how concern partipants are towards there voice assistants.

In terms of security measures, many participants have taken proactive steps, such as enabling multi-factor authentication, using strong passphrases, regularly updating software, being cautious with personal information, and monitoring system activity. These efforts demonstrate a proactive approach to safeguarding their speech and speaker recognition systems from hidden voice attacks.The [Figure 20,21,22] clearly depicts participants concern towards security measures.

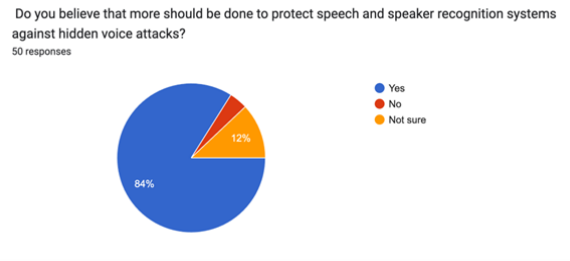

Figure 21: This figure shows participants are more concern with security measures.

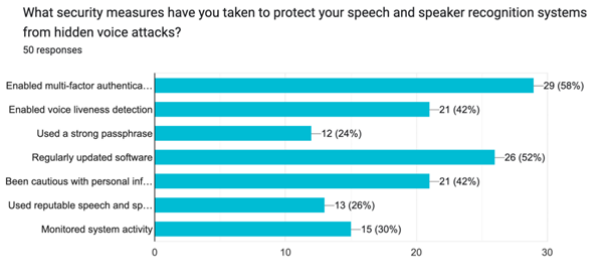

Figure 22: This figure shows what kind of measures are required for better security.

F. Importance of Security

The survey's findings show that participants place a high value on voice and speaker recognition system security in both their personal and professional life. The majority of respondents expressed that it is "very important" to them. Although some participants thought it was "somewhat important," security was emphasized significantly overall. This demonstrates how these systems' potential hazards are recognized, and how protection must be provided for them in order to secure sensitive and private data. The [figure 23] clearly depicts importance of security related to personal or professional lifes.

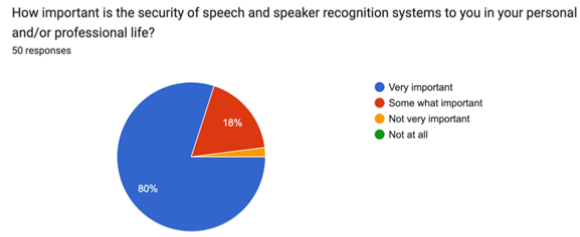

Figure 23: This figure shows how important is the security of speech and speaker recognition system related to personal and professional life.

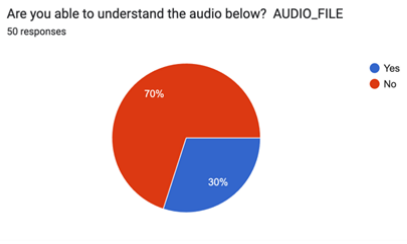

Figure 24: This figure shows the understanding of audio file that is used in this research study.

G. Audio Understanding

According to the survey's findings, the majority of participants (35 out of 50) said the audio was difficult for them to comprehend. Only 15 of the respondents said they could understand the audio The [Figure 24] show the understanding of audio file by participants. This shows that the participants may have had issues or troubles understanding the audio content.

8.3 Conclusion

The survey on hidden voice attacks and their impact on speech and speaker recognition systems revealed several important findings. First, it was discovered that a sizable portion of participants had experienced their Home Assistant, Voice Processing, or Speaker Recognition systems carrying out unknown commands, suggesting possible flaws in these systems. Furthermore, the majority of respondents expressed worries about the security of these systems in light of hidden voice attacks and said they were aware of them. The survey made clear how crucial it is to have security measures in place to safeguard speech and speaker identification systems against covert voice attacks. Participants agreed that more needs to be done to improve the security of these systems. The use of strong passphrases, multi-factor authentication, routine software updates, and being cautious with personal information are among the proactive security methods that many participants reported implementing. These actions prove that the organization is committed to protecting sensitive data and is aware of the dangers posed by covert voice attacks.

The survey results underscore the significance of mitigating concealed voice threats in speech and speaker identification systems, in light of their findings. We can improve the overall security and integrity of these systems, protecting user privacy and sensitive data, by putting in place strong security measures and increasing awareness of these assaults.

9 Conclusion and FutureWork

In conclusion, we have successfully implemented three different attacks on speech and speaker recognition systems: Time Domain Inversion, Random Phase Generation, and Time-Scaling. These attacks have demonstrated the vulnerabilities of these systems to hidden voice attacks and highlighted the need for stronger security measures to protect against such attacks. Further research and development are needed to improve the security of these systems and prevent them from being exploited in real-world situations.

Based on the results of the survey, you can identify specific areas of concern and potential solutions for improving the security of speech and speaker recognition systems against hidden voice attacks. Some potential future work could include Developing and implementing new security features, such as multi-factor authentication and voice liveness detection, that are specifically designed to protect against hidden voice attacks. Conducting further research into the effectiveness of existing security measures, such as password requirements and software updates, in preventing hidden voice attacks. Working with speech and speaker recognition system providers to encourage the adoption of stronger security measures and hold them accountable for ensuring the security of their systems.

10 Contributions

Jaipal	Kota	Thota
Introduction	**Related Work**	**Abstract and Background**
Time Scaling	**Time Domain Inversion**	**Random Phase Generation**
Experimental setup	**Evaluation**	**Positive Outcomes**
Survey implementation	**Survey implementation**	**Survey implementation**

References

[1] ABDULLAH, H., GARCIA, W., PEETERS, C., TRAYNOR, P., BUTLER, K., AND WILSON, J. Practical hidden voice attacks against speech and speaker recognition systems.

[2] AFOURAS, T., CHUNG, J. S., SENIOR, A., VINYALS, O., AND ZISSERMAN, A. Deep audio-visual speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 44*, 12 (2022), 8717–8727.

[3] ANAND, S. A., AND SAXENA, N. Speechless: Analyzing the threat to speech privacy from smartphone motion sensors. *2018 IEEE Symposium on Security and Privacy (SP)* (2018), 1000–1017.

[4] CARLINI, N., MISHRA, P., VAIDYA, T., ZHANG, Y., SHERR, M., SHIELDS, C., WAGNER, D., AND ZHOU, W. Hidden voice commands. In *25th USENIX Security Symposium (USENIX Security 16)* (Austin, TX, Aug. 2016), USENIX Association, pp. 513–530.

[5] CHANG, R., KUO, L., LIU, A., AND SEHATBAKHSH, N. Sok: A study of the security on voice processing systems. *ArXiv abs/2112.13144* (2021).

[6] DEY, N. S., MOHANTY, R., AND CHUGH, K. Speech and speaker recognition system using artificial neural networks and hidden markov model. In *2012 International Conference on Communication Systems and Network Technologies* (2012), pp. 311–315.

[7] GRAVES, A., AND JAITLY, N. Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the 31st International Conference on Machine Learning* (Bejing, China, 22–24 Jun 2014), E. P. Xing and T. Jebara, Eds., vol. 32 of *Proceedings of Machine Learning Research*, PMLR, pp. 1764–1772.

[8] LAN, J., ZHANG, R., YAN, Z., WANG, J., CHEN, Y., AND HOU, R. Adversarial attacks and defenses in speaker recognition systems: A survey. *Journal of Systems Architecture 127* (2022), 102526.

[9] LEU, F.-Y., AND LIN, G.-L. An mfcc-based speaker identification system. In *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)* (2017), pp. 1055–1062.

[10] MOHD HANIFA, R., ISA, K., AND MOHAMAD, S. A review on speaker recognition: Technology and challenges. *Computers Electrical Engineering 90* (2021), 107005.

[11] NODA, K., YAMAGUCHI, Y., NAKADAI, K., OKUNO, H., AND OGATA, T. Audio-visual speech recognition using deep learning. *Applied Intelligence 42*, 4 (June 2015), 722–737. Funding Information: This work has been supported by JST PRESTO "Information Environment and Humans" and MEXT Grant-in-Aid for Scientific Research on Innovative Areas "Constructive Developmental Science" (24119003), Scientific Research (S) (24220006), and JSPS Fellows (265114). Publisher Copyright: © 2014, The Author(s).

[12] OLIVIER, R., RAJ, B., AND SHAH, M. High-frequency adversarial defense for speech and audio. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), pp. 2995–2999.

[13] ABDULLAH, T., WARREN, K., BINDSCHAEDLER, V., PAPERNOT, N., AND TRAYNOR, P. SoK: The Faults in our ASRs: An Overview of Attacks against Automatic Speech Recognition and Speaker Identification Systems. In *IEEE Symposium on Security and Privacy (IEEE S&P)* (2021).

[14] YUAN, X., CHEN, Y., ZHAO, Y., LONG, Y., LIU, X., CHEN, K., ZHANG, S., HUANG, H., WANG, X., AND GUNTER, C. A. CommanderSong: A systematic approach for practical adversarial voice recognition. In *27th USENIX Security Symposium (USENIX Security 18)* (Baltimore, MD, Aug. 2018), USENIX Association, pp. 49–64.

[15] ZHANG, X., PENG, Y., AND XU, X. An overview of speech recognition technology. In *2019 4th International Conference on Control, Robotics and Cybernetics (CRC)* (2019), pp. 81–85.

[16] ZHANG, X., TAN, H., HUANG, X., ZHANG, D., TANG, K., AND GU, Z. Adversarial example attacks against asr systems: An overview. In *2022 7th IEEE International Conference on Data Science in Cyberspace (DSC)* (2022), pp. 470–477.