VOICE SCRAMBLING

Hao Yen
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332
rick.yen@gatech.edu

Aoun Hussain
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332
aoun.hussain@gatech.edu

Linhao Zhao
Center for Music Technology
Georgia Institute of Technology
Atlanta, GA 30332

1 Introduction

In speech communication, a secure communication system is important to maintain privacy of conversation between two parties without the third party to listen in. We can divide the methods for secure communication into two main categories: scrambling and encryption. A scrambler is a device that transposes or inverts signals or otherwise encodes a message at the transmitter to make the message unintelligible at a receiver not equipped with an appropriately set descrambling device [1]. The idea of encryption is to employ a digital voice encoder, which turns the voice signal into a bit stream, and a digital encryption system, which encrypts the bit stream into another bit stream that cannot be decoded back into a useful bit stream without the encryption key. Whereas encryption usually refers to operations carried out in the digital domain, scrambling usually refers to operations carried out in the analogue domain.

While voice encryption often requires a sophisticated encryption system to do the job right, in this project, we mainly explore different methods for voice scrambling which is only designed to make the voice not readily intelligible. Specifically, we perform scrambling on both time domain and frequency domain to see the effectiveness of scrambling methods. It should be noted that the level of sophistication in voice scrambling is not very high and thus not very secure; however, for tactical or temporary privacy, it may suffice. In addition, we also implement a simple encryption system using generated key to have a better idea of how voice encryption works.

2 Related Techniques

2.1 Scrambling

As mentioned above, scrambling is a method that works on analogue domain. In general, we can characterize the method into two aspects: frequency domain and time domain scrambling. It is obvious that the two methods work on different domain. We will give a more detailed description regarding each in the following sections.

2.1.1 Frequency Domain Scrambling

In frequency domain scrambling, the whole process of scrambling is changing the original frequency bands into several sub frequency bands (channel), and then randomly permute the sub frequency bands based on the permissible rate, finally, reassemble the sub frequency bands. As for the descrambling, the descrambled channel is the same as scrambled channel, but the channel matrix must be inverted to redistribute the frequency band as in original audio. In this project, we followed the description of the theoretical frequency scrambling in this paper [2] and implemented from scratch.

2.1.2 Time Domain Scrambling

For time domain scrambling, the audio signal is divided into multiple segments, which are then permuted in a certain order. Some of the popular time domain techniques are Time-Inversion, Time Segment Permutation (TSP), Hopping-Window and Sliding Window TSP, Time Shifting of Speech Sub-bands, Reverberation [3] and time-domain based scrambler which does not need synchronization [4]. In this project, we will be implementing a simple TSP, which will be explained in detail in the system description section.

2.1.3 Voice Inversion

Voice inversion is a method of scrambling conversations to render speech nearly unintelligible for others [**?**, 5]. As implied by the name itself, this technique inverts the audio spectrum of a signal, making the lowest frequencies the highest and vice versa. That is, one can view this method as another frequency domain scrambling. In fact, voice inversion is the most used method of scrambling. It works by inverting the audio spectrum at a set maximum frequency called the inversion carrier. Frequencies near this carrier will thus become frequencies near zero Hz, and vice versa. The resulting audio is unintelligible, though familiar sentences can sometimes be recognized by human ears.

The most simple inversion method technique is called base-band inversion, which means choosing one single frequency carrier. The spectrum is then inverted at a single preset never changing frequency (split point). The descrambling is obviously very simple, if you take the scrambled input and invert it around the same frequency used to scramble you obtain the original signal again. Split-band inversion is an approach for more security as compared to base-band inversion. Instead of performing a single inversion of the complete band, the spectrum is divided into tow parts (bands), and base-band inversion is processed to both parts before they are recombined again. This way creates a more complex scheme and one can achieve more security during communication. However, even this technique does not increase the security level significantly since it still brings along the same vulnerabilities of insecurity. One of the problems of all the inversion techniques is that fundamental characteristics of the voice signal are not significantly altered, which makes them always vulnerable to malicious third parties. In this project, we implement a simple base-band inversion and split-band inversion to demonstrate the effectiveness of voice inversion.

2.2 Encryption

As mentioned before, encryption often refers to digital technologies. Digital encryption can be seen as a much stronger method of protecting speech communications than analogue scrambling. The big advantage of digital encryption is that it does not matter what kind of signal is encrypted. That makes digital encryption quite powerful because you can create one standard to handle e.g. text, audio, video and every other kind of data. Certainly, digital encryption takes always the same start point, the analogue to digital conversation. A digital secure voice usually includes two components, a digitizer to convert between speech and digital signals and an encryption system to provide confidentiality.

Encryption is a method in which data is rendered hard to read by an unauthorized party [6]. This process converts the original representation of the information, known as plaintext, into an alternative form known as ciphertext. Ideally, with a key, only the authorized parties can decipher a ciphertext back to plaintext and access the original information. Encryption does not itself prevent interference but denies the intelligible content to a malicious interceptor. For technical reasons, an encryption scheme usually uses a pseudo-random encryption key generated by an algorithm. It is possible to decrypt the message without possessing the key but, for a well-designed encryption scheme, considerable computational resources and skills are required. An authorized recipient can easily decrypt the message with the key provided by the originator to recipients but not to unauthorized users.

3 System Description

3.1 Frequency Domain Scrambling

The flowchart of scrambling and descrambling process is shown in Figure 1. The main idea of scrambler block is randomly permute the different frequency filters. And descrambler block is the inverse permutation to reassemble the spectrum back to the original order.

An example of frequency separation into four sub frequency bands (channels) is shown in Figure 2. In each stages, the audio stream will go through a high pass filter or a low pass filter following down sample by 2.

An example of randomly permutation of those sub frequency bands is shown in Figure 3, and the following frequency reassembling is shown in Figure 4. The frequency reassembling is a simple inverse of frequency separation with same filtering frequency and up sample rate in each stages of frequency separation.

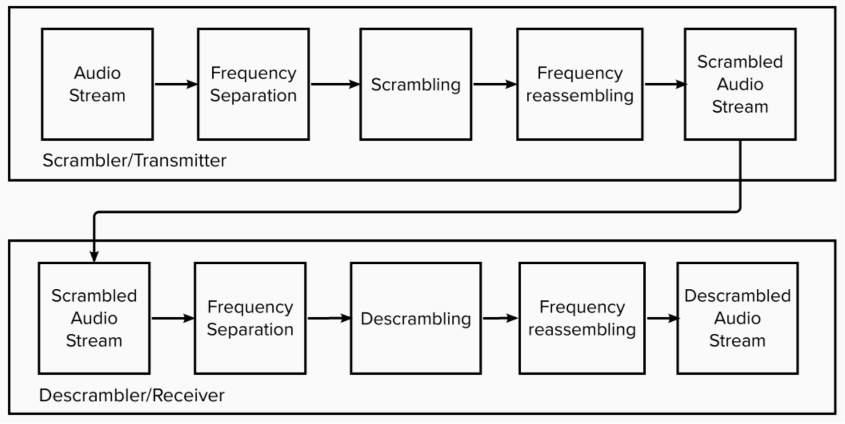


Figure 1: Block diagram of voice scrambler and descrambler in frequency domain method

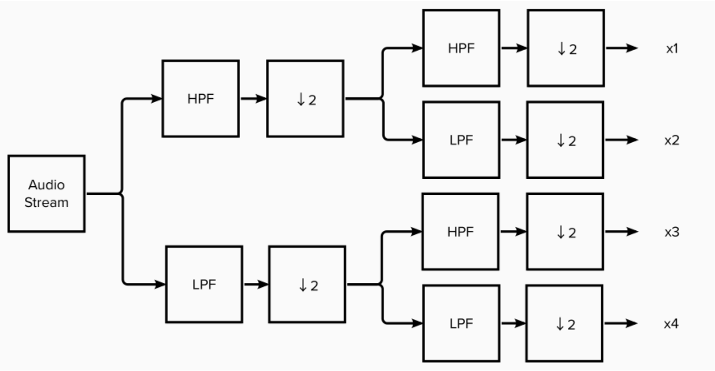


Figure 2: An example of frequency separation into four sub frequency bands (channels)

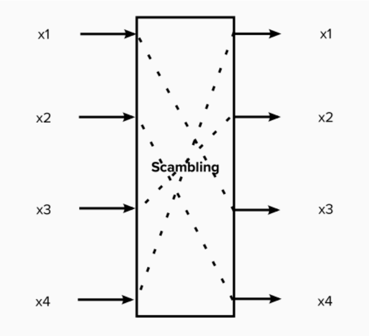


Figure 3: An example of randomly permutation of sub frequency bands

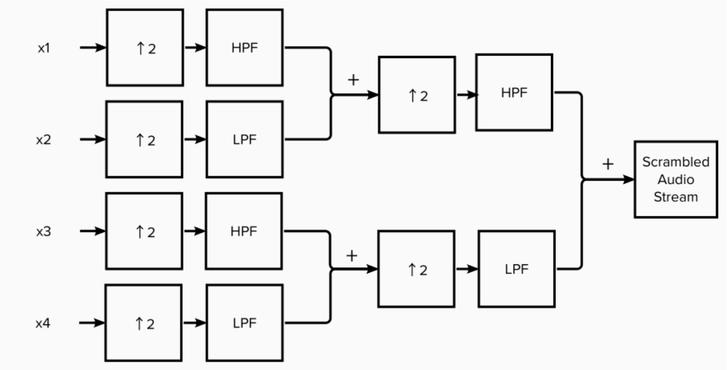


Figure 4: Frequency reassembling

3.2 Time Domain Scrambling (TSP)

We used time segment permutation technique to scramble an audio signal in time domain. The scrambling algorithm are as follow.

STEP 1: Read the audio file and detect its sampling frequency
STEP 2: Split the audio in equal intervals of time frames
STEP 3: Randomly order or permute the frames to make the audio unintelligible
STEP 4: Save the order of permutation as a key in the text file

To retrieve the original signal, we used the following algorithm:

STEP 1: Read the scrambled audio file and detect its sampling frequency
STEP 2: Read the key text file
STEP 3: Run an inverting algorithm to retrieve the original order of frames
STEP 4: Reorder the frames to retrieve unscrambled original audio

3.3 Deinvert

Deinver is an open source code for voice inversion [7]. The algorithm behind Deinvert [8] can be divided into three phases: 1) pre-filtering, 2) mixing, and 3) post-filtering. Mixing means multiplying the signal by an oscillation at the selected carrier frequency. This produces two sidebands, or mirrored copies of the signal, with the lower one frequency-inverted. Pre-filtering is necessary to prevent this lower sideband from aliasing when its highest components would go below zero Hertz. Post-filtering removes the upper sideband, leaving just the inverted audio. Both filters can be realized as low-pass FIR filters.

It should be noted that this operation is its own inverse. That is, by applying the same inversion again to the scrambled or inverted spectrogram, we can get intelligible speech back. For base-band inversion, we randomly set the preset frequency around 2600Hz. As for split point, we choose 1200Hz.

3.4 Encryption

Encryption is not the main method we want to discuss in this project. Therefore, we just want to provide some basic insights into how a digital encryption system works. In this project, we use public library from python to create an encryption key to encrypt and decrypt the speech signal. The code is provided and you are welcomed to try it. If you run the code correctly, you will find out the encrypted file is not playable and the decrypted file is the same as the original speech.

4 Results and Discussion

4.1 Frequency Domain Scrambling

Setting the $channel = 4$, $permissible\ rate = 0$, the original, scrambled, descrambled spectrogram of **threesentences.wav** is shown in Figure 5. In the scrambled spectrogram, we can see the the frequency distribution is permuted by comparing the original one. However, some power of the frequency lost during the the process of descrambling which leads to a lower audio quality of the descrambled one.

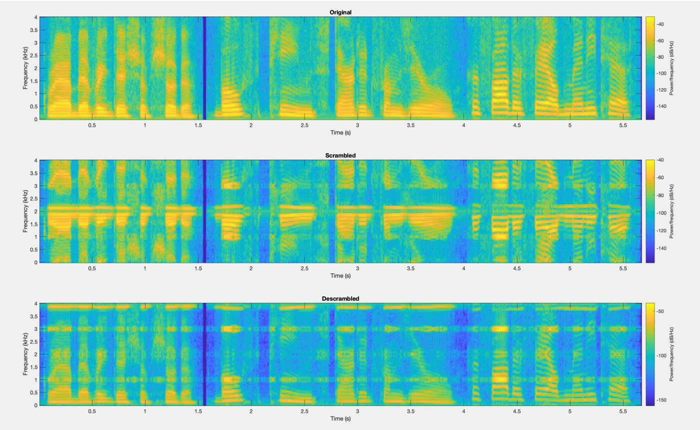


Figure 5: The original, scrambled, descrambled spectrogram of an example audio. The spectral channel is 4 and the permissible rate is 0

Setting the $channel = 4$, $permissible\ rate = \frac{1}{3}$, the original, scrambled, descrambled spectrogram of **threesentences.wav** is shown in Figure 6. In the scrambled spectrogram, we can see the the frequency distribution is one change in permutation every three seconds by comparing the original one. Also, some power of the frequency lost.

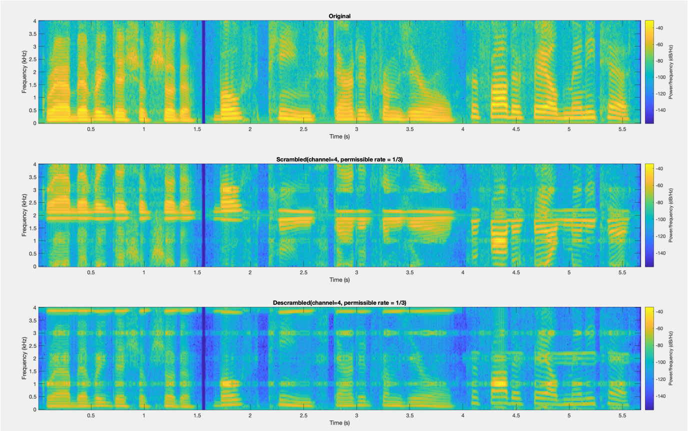


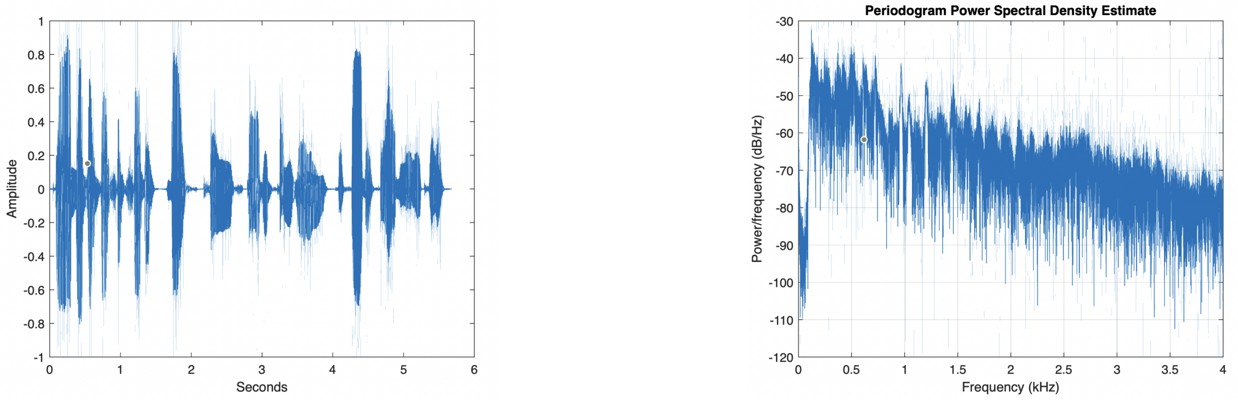
Figure 6: The original, scrambled, descrambled spectrogram of an example audio. The spectral channel is 4 and the permissible rate is 1/3

4.2 Time Domain Scrambling: TSP

After running the scrambling algorithm on **threesentences.wav**, we plotted the original audio (Figure 7a and 7b) and scrambled audio in both time and frequency domain to observe differences. After scrambling, and splitting and reordering the audio frames, the following plots can be observed in Figure 8a and 8b.

After reading the key and with the help of the descrambling algorithm, we finally recovered the audio on the receiver end. We observed that after scrambling, the amplitude of the audio signal decreased by half, due to the algorithm. However, the amplitude will only affect the sound intensity of the audio. The real trick is the reordering of the frames which can be clearly observed by the differences between Figure 7a and Figure 8a. After descrambling, the time domain domain graphs is retrieved, however with 1/4 amplitude of the original, as observed by the differences between Figure 7a and Figure 9a.

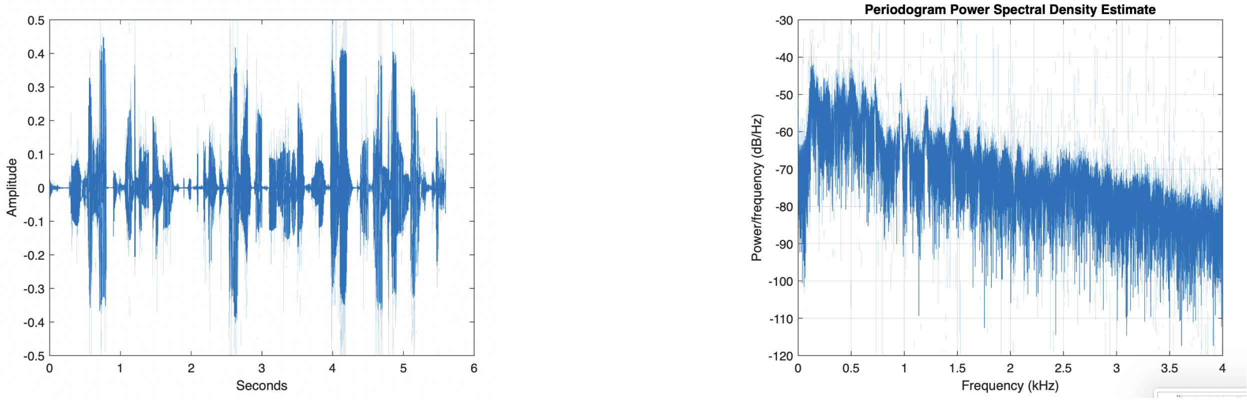
In the frequency domain, the results may not be that apparent, as this technique (Time Segment Permutation) primarily involved changing the time domain characteristics of the signal. No prominent trend is observed in all 3



(a)

(b)

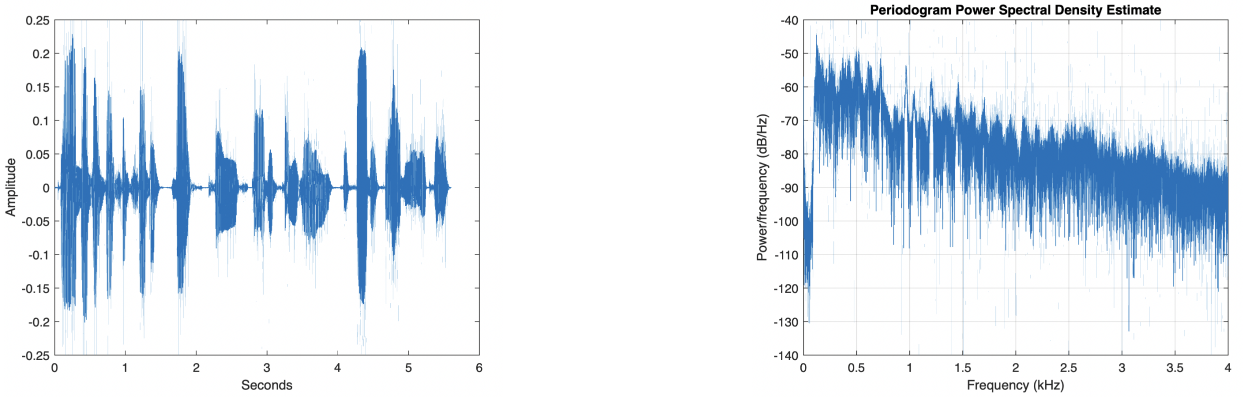
Figure 7: Original Audio Time Domain and Audio Power Spectral Density



(a)

(b)

Figure 8: Scrambled Audio Time Domain and Audio Power Spectral Density



(a)

(b)

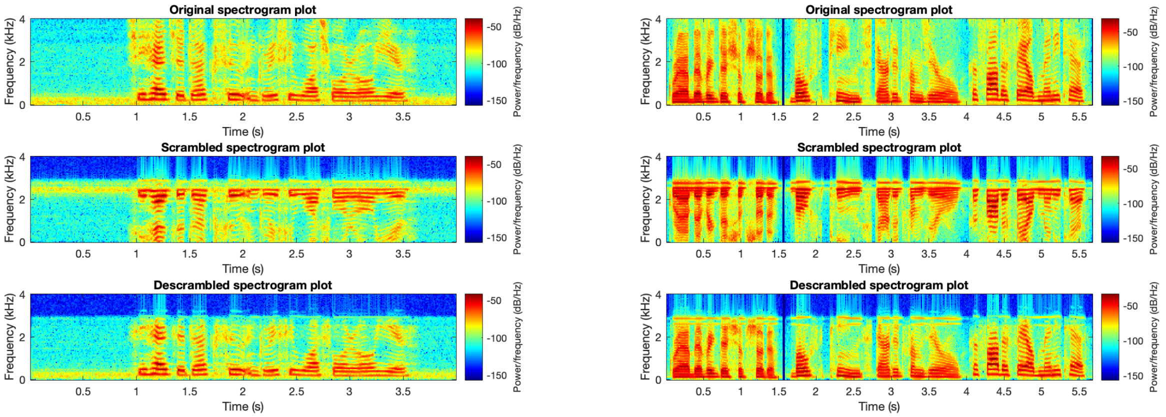
Figure 9: Descrambled Audio Time Domain and Audio Power Spectral Density

graphs of the power spectral density, however, constant decrease in power (db) is observed, but the min and max frequency of the signal remain same. The trend of PSD graph remain same.

4.3 Deinvert

From Figure 10, we can see the scrambled or inverted spectrogram using Deinvert toolkit for base-band inversion as described in 3.3. We can easily observe that the scrambled signal is just an inverted version of the descrambled spectrogram and also the spectrogram above the carrier frequency are set to zero so we might lose some information of the original signal. The descrambled or reconstructed spectrogram are basically the same as the original ones except the high frequency region.

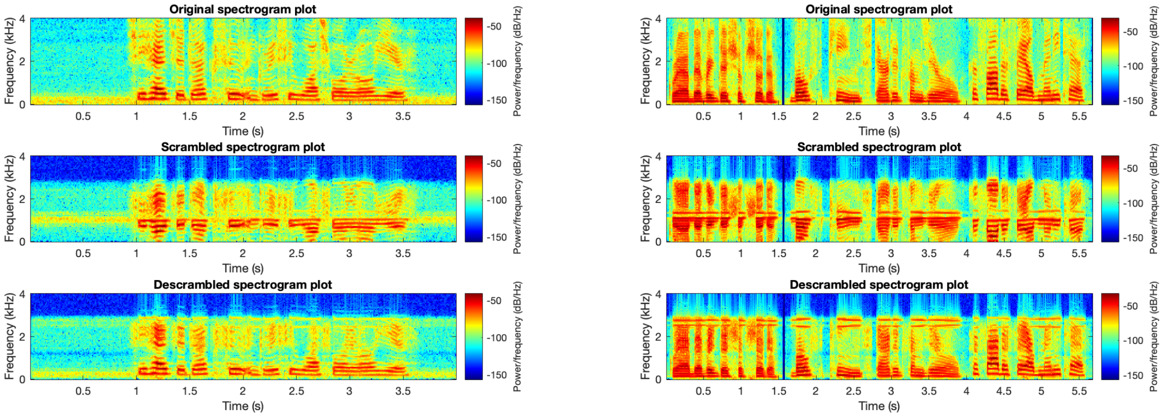
The split-band inversion in Figure 11 shows a similar results. The only difference is that from the scrambled spectrogram, we can clearly see the split point (around 1200Hz) where the spectrogram is split and both bands are inverted separately.



(a)

(b)

Figure 10: The original spectrogram of the speech, scrambled spectrogram, and descrambled spectrogram of the signal using the base-band inversion.



(a)

(b)

Figure 11: The original spectrogram of the speech, scrambled spectrogram, and descrambled spectrogram of the signal using the split-band inversion.

4.4 Encryption

Please refer to the code. The provided code will produce three outputs: a key file, an encrypted and a decrypted audio files.

5 Conclusion

In this project, we investigate voice security system using scrambling and encryption methods. We implement different voice scrambling methods on frequency and time domain respectively. The results show that for simple and temporary usage, our system can work as we rendered the speech unintelligible with the scrambler. The descrambler also works as expected as we can reconstruct the speech back to the original one with a proper key. Finally, we briefly introduce the common encryption method for a more secure communication system and demonstrate it using a simple code. The code and results will be provided in the submitted file.

References

[1] https://en.wikipedia.org/wiki/Scrambler.

[2] http://repository.sustech.edu/bitstream/handle/123456789/26343/DesigningofReal%20%20.....pdf?sequence=1.

[3] N. S. Jayant. Analog scramblers for speech privacy, computers, and security. *North-Holland Publishing Company*, pages 275–89, 1982.

[4] F. Huang and E. V. Stansfield. Time sample speech scrambler which does not require synchronization. *IEEE Transactions on Communications*, pages 1715–1722, 1993.

[5] Markus Albert Brandau. Implementation of a real-time voice encryption system. 2008.

[6] https://en.wikipedia.org/wiki/Secure_communication#Encryption.

[7] https://github.com/windytan/deinvert.git.

[8] https://www.windytan.com/2017/09/descrambling-split-band-voice-inversion.html.