Voice Scrambler

By

Captain Taimoor Mir

Captain Raja Muhammad Rashid Zaman

Captain Jaseem Ali

Supervised by:

Dr Abdul Wakeel

Submitted to the faculty of Department of Electrical Engineering,
Military College of Signals, National University of Sciences and Technology, Islamabad,
in partial fulfillment of the requirements of a B.E Degree in Electrical (Telecom) Engineering.

June 2023

In the name of ALLAH, the Most benevolent, the Most Courteous

CERTIFICATE OF CORRECTNESS AND APPROVAL

This is to officially state that the thesis work contained in this report

"Voice Scrambler"

is carried out by

Captain Taimoor Mir

Captain Raja Muhammad Rashid Zaman

Captain Jaseem Ali

under my supervision and that in my judgment, it is fully ample, in scope and excellence, for the degree of Bachelor of Electrical (Telecom.) Engineering in Military College of Signals, National University of Sciences and Technology (NUST), Islamabad.

Approved by

Supervisor

Dr Abdul Wakeel

Department of EE, MCS

Date: _____

DECLARATION OF ORIGINALITY

We hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

ACKNOWLEDGEMENTS

Allah Subhan'Wa'Tala is the sole guidance in all domains.

Our parents, colleagues, and most of all supervisors, Dr Abdul Wakeel without your guidance.

The group members, through all adversities, worked steadfastly.

Plagiarism Certificate (Turnitin Report)

This thesis has ____ similarity index. The Turnitin report endorsed by Supervisor is attached.

_____

Student 1 Name

NUST Serial no

_____

Student 2 Name

NUST Serial no

_____

Student 3 Name

NUST Serial no

_____

Student 4 Name

NUST Serial no

_____

Signature of Supervisor

ABSTRACT

In today's world, the security of sensitive information being communicated over voice channels has become a major concern. Many voice terminals, including mobile phones and telephone exchanges, lack encryption, making them vulnerable to hacking and eavesdropping. The objective of this project is to design and develop a voice scrambler that can modify the sound of a person's voice in real time using bit-shifting techniques, making it difficult for unauthorized listeners to understand or intercept the conversation. The proposed system has been tested and validated through simulations, tests, and trials, demonstrating its efficiency in scrambling the input voice signal. The system's effectiveness in ensuring the confidentiality of transmitted information has also been established, making it a promising solution for secure voice communication. It is believed that the proposed voice scrambler can play an essential role in protecting the privacy and security of sensitive or classified communications and could be adopted by organizations and individuals who require secure communication channels. Further research and development in this area could lead to the creation of even more advanced voice-scrambling technologies.

Table of Contents

List of Figures

Chapter 1: Introduction

Presently securing voice communication has become extremely important for both civil and military applications, especially for real-time communication. A scrambler is a device that alters, or inverts signals or otherwise encodes a message at the sender's side making it unintelligible at the receiver side without an appropriately configured descrambling device [1]. Security could be attained using the analogue technique known as scrambling. To create a scrambled channel for transmission, the original signal's parameters are changed in the time and frequency domain in accordance with a predetermined code. To create the original channel, the scrambled channel must be received and descrambled.

With the use of various cyphering techniques on the signal's digital value, a digital approach such as encryption could be used to establish security. Although very high degrees of security could be attained, doing so requires a lot of bandwidth and puts numerous limitations on real-time implementation, including stream cypher.

1.1 Overview

In the modern world, the era of information and technology voice communication is a need of everyone, and the security of the voice communication has become crucial for both civil and military purposes, particularly for real-time communication. Military forces require communication security to protect their sensitive voice communications from being intercepted by enemy forces. Intelligence agencies and private security firms require communication security to protect their communications and prevent espionage or unauthorized access to classified information.

A voice scrambler is a device that is used to modify or encrypt speech to make it unintelligible to anyone who is not authorized to listen to the communication. The use of voice scramblers is widespread in both military and civilian applications, including law enforcement, intelligence agencies, and private security [2].

In general, speech scrambler is used to safeguard private communications and guarantee the security and privacy of voice interactions. It is widely used in various military and civil sectors and is essential for upholding secrecy and guarding against espionage.

1.2 Problem Statement

Most voice links, including mobile and telephone exchanges, do not use cyphers. As a result, they become more open to hacking and eavesdropping.

Speech in real-time communication systems like GSM, Voice Over Internet Protocol, Telephone, and analogue Radio must be guaranteed with end-to-end security.

1.3 Proposed Solution

The major goal of our proposed solution is to build a device which convert input voice signal into unintelligible form to prevent against any unauthorized listening and eavesdropping and then recover the original voice signal at receiver end using the descrambler programmed with same algorithm as the scrambler.

1.4 Working Principle

The project mainly works on the principles of digital signal processing using STM32F407G microcontroller. The project is divided into three parts. These parts are as under:

- Signal Input
- Signal Processing/Scrambling
- Signal transmission
- Signal Reception

1.4.1 Signal Input:

Input voice signal is transmitted via digital microphone embedded on STM32F407G microcontroller. As STM32F407G has embedded digital microphone, neither external analogue microphone nor amplifier is required for transmission of input voice signal. Analog voice is converted into digital form and is sent to micro-chip for signal processing [3].

1.4.2 Signal Processing/Scrambling:

The digitized input signal is initially received by the microcontroller chip, which partitions it into 8-bit segments and subsequently stores them within a 32-bit buffer. Once this buffer reaches its capacity, the microcontroller chip engages in requisite bit manipulations, as dictated by the program design, prior to transferring these bits to the transmission port for conveyance across a designated communication channel.

The microcontroller is equipped with a total of five 32-bit buffers, namely the Sin buffer, record buffer, transmission buffer, receive buffer, and play buffer. In scenarios where scrambling has been enabled according to the program specifications, the record buffer assumes the responsibility of recording the initial 32 bits of the input signal. Subsequently, an AND operation is performed between these recorded bits and a pre-defined 32-bit pattern stored within the record buffer, thereby generating a scrambled signal that is then directed to the transmission buffer.

Conversely, when the program specifications entail descrambling functionality, the scrambled 32-bit signal is transmitted via the transmission buffer. Upon reception, the play buffer undertakes the inverse of the previously mentioned AND operation, effectively recovering the original signal. In essence, the voice scrambler system offers the flexibility and adaptability to enable or disable the scrambling process based on the program specifications. This provision allows for the tailoring of the system's operation to suit diverse use cases. The decision to scramble or descramble the audio data is contingent upon the desired level of security and confidentiality required for the specific communication channel at hand.

1.4.3 Signal Transmission:

Upon completion of the necessary operations on the digital voice signal, the micro-controller chip transmits the resulting signal to the transmission port using UART protocol, which in this case is PD8. Protocol and the transmission port number are pre-defined in the program specifications. This serves as the output port for the voice scrambler and allows for the secure transmission of the scrambled message over a communication channel.

1.4.4 Signal Reception:

The transmission medium utilized by a voice scrambler can be either wired or wireless, depending upon the design specifications of the program. The choice of transmission medium depends on a variety of factors, such as the type of communication channel, the distance between the sender and the receiver, and the level of security required. The signal received from the communication channel (in this case is wired) is directed to the port that has been previously declared in the program specifications, which in this case is PA3. The receive buffer receives the signal via

UART protocol and sends it to DAC or play buffer according to the program specifications. After necessary digital to audio conversion the signal is sent to amplifier/speakers where it can be heard.

The amplifier's functionality allows the scrambled or descrambled voice to be heard, depending on the specifications set forth in the program. This enables the listener to receive the transmitted audio signal with clarity and accuracy, while maintaining the desired level of security and confidentiality.

1.5 Objectives

1.5.1 General Objectives:

The main objective of this work is to design, test and implement a real time microcontroller-based voice scrambling / descrambling system, keeping the voice band without any change.

1.5.2 Academic Objectives:

- Develop a real-time voice scrambler that utilizes encryption algorithms to secure voice communication.
- Implement the developed encryption algorithm on a microcontroller and simulate the resulting encrypted voice communication to assess its effectiveness.
- Investigate the impact of teamwork on project productivity and explore methods to improve team collaboration for successful project completion.

- Develop a project that addresses the need for enhanced communication security in the Pakistan armed forces and the wider community through the implementation of the voice scrambler.

These objectives aim to contribute to the field of communication security through the development and evaluation of a real-time voice scrambler. Additionally, the project seeks to enhance productivity through effective teamwork and to address a specific national security need in Pakistan.

1.6 Scope

The project scope is to develop a voice scrambling system that can securely scramble the input voice signals in real-time. The system will have functionalities such as real-time audio scrambling, configurable scrambling strength, support for multiple audio formats, integration with various communication channels, user authentication, and logging and auditing of all voice communication activities.

1.7 Deliverables

1.7.1 Security and Privacy

It provides a secure and private voice communication system that assures confidentiality and ensures that only authorized parties can access the voice communication and preventing interception or eavesdropping by malicious actors.

1.7.2 Integration with Communication Channels:

It can integrate with various communication channels, such as phone lines, VoIP networks, and radio communication, to provide secure communication channels. It can work with different types of voice communication networks, providing scrambling and descrambling capabilities to protect sensitive information transmitted over these networks.

1.7.3 Compatibility with Audio Devices:

It has compatibility with a range of audio hardware devices, such as microphones and speakers, to allow secure voice communication. The compatibility with a range of audio hardware devices is an essential feature, as it allows users to use their preferred audio devices for secure voice communication, enhancing the convenience and flexibility of the system.

1.8 Relevant Sustainable Development Goals

The project voice scrambler can address several locally relevant socio-economic issues, such as protecting the privacy and confidentiality of communication in sensitive industries or activities, such as military, law enforcement, health care, and finance. By providing secure voice communication channels, the project can prevent unauthorized access to sensitive information, reducing the risk of data breaches, identity theft, fraud, or espionage.

1.8.1 Industry, Innovation and Infrastructure

The project can promote innovation in the communication industry by providing a new technology that enhances the security and privacy of voice communication. It can also contribute

to the development of a resilient and sustainable infrastructure that can support secure communication networks.

1.8.2 Peace, Justice, and Strong Institutions

The project can support the promotion of peaceful and inclusive societies by providing a tool that can prevent conflicts, reduce corruption, and enhance the rule of law. It can also contribute to the development of strong institutions that can protect the rights and privacy of individuals and prevent abuses of power.

1.8.3 Partnerships for the Goals

The project can foster partnerships between different stakeholders, such as governments, private companies, and civil society organizations, to promote the development and deployment of secure communication technologies. It can also contribute to the transfer of technology and knowledge to developing countries, promoting capacity building and sustainable development.

1.9 Structure of Thesis

- Chapter 2 contains the literature review and the background and analysis study this thesis is based upon.
- Chapter 3 contains the design and implementation of the project.
- Chapter 4 introduces detailed evaluation and analysis of the code.
- Chapter 5 contains the conclusion of the project.
- Chapter 6 highlights the future work needed to be done

Chapter 2: Literature Review

Differentiating digital encryption from analogue scrambling can be challenging and unclear to determine whether a particular device employs digital encryption or analogue scrambling. The method employed to secure the voice is what makes the two distinct from one another. The device employs analogue scrambling if the method employed to secure the voice merely entails some altering of the time or frequency characteristics of the speech signal. The device employs digital encryption if the method used to secure the voice entails encrypting digital voice with a traditional cypher algorithm to produce digital cypher text.

2.1 Techniques Used in Voice Scrambling

2.1.1 Frequency Scrambling

The most popular method for frequency scrambling relies on changing the signal frequencies in a certain way. Review of several speech scrambling design approaches in the audio band is provided in this part, with a focus on real-time implementation [4].

2.1.1.1 Fast Fourier Transform

The frequency components of the signal are calculated using the Fast Fourier transform [5], while the analog transform is described as

$$X(w) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$$

Where X (w) is frequency domain of the signal

The digital Fourier transform (DFT) is defined by

$$X(k) = N_{ck} = \sum_{n=0}^{N-1} \mathrm{x(n)e}^{-j\frac{2\pi \mathrm{kn}}{N}} \quad , \mathrm{k} = 0,1, \dots , \mathrm{N} - 1$$

Where X (k) is digital frequency component of the signal The frequency resolution (the space between X (K) and X (K+1) is given by

$$F_{resolution} = \frac{F_s}{N}, F_s = sampling\ frequency$$

For voice signal with $F_s = 8912\ Hz,\ N = 1024$ the frequency resolution = 8Hz

Noting that DFT require N*N complex multiplication (for N=1024, number of multiplication = 1,048,576) which is so difficult to be used in real time implementation.

To reduce the number of multiplication fast Fourier transform algorithm (FFT) is used. Which reduce the number of multiplications to Complex multiplication of FFT $= \frac{N}{2} log_2 N$

For N=1024, the number of complex multiplications = 3,465. The general block diagram for voice scrambler and descramble as shown

in figure (2.1).

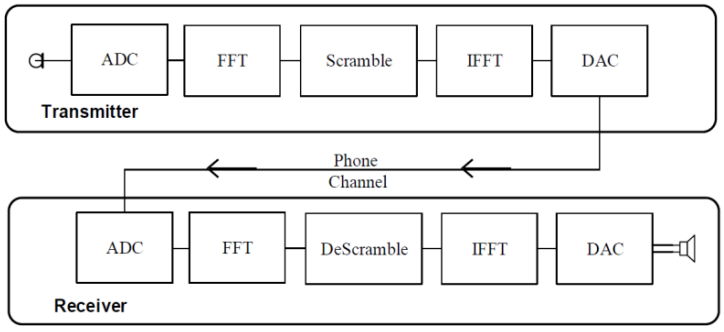


Figure (2.1) Block diagram of FFT voice scrambler and descrambler

The fundamental concept behind a scramble block is the permutation of FFT filters, either at random or in a predetermined order as specified in a lookup table. For FFT filters, the descramble block is the inverse permutation to restore the original sequence.

The primary considerations for real-time implementation include.

1. Floating point multiplication is required, necessitating the employment of a specialised DSP chip.
2. The number of FFT points to be employed, which adds a time delay to the frequency resolution and channel scrambling and descrambling (sampling frequency/number of points).
3. For sampling frequency 8000Hz the processing time for scrambling and descrambling have to less than 125 microseconds.

2.1.1.2 Frequency Hopping Inversion

In a frequency hopping spread spectrum (FHSS) system,the transmitted signal is spread across multiple channels as shown in Figure (2.2 a) below . in figure (2.2 b) the full bandwidth is divided into 8 channels , centered at f1 through f8 the signal “hops” between them in the following sequence f5 f8 f3 f6 f1 f7 f4 f2 [6] .

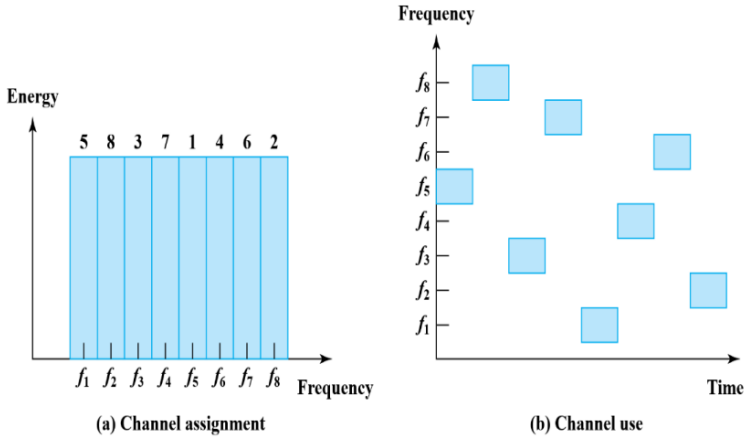


Figure (2.2) Frequency hopping example

Figure (2.2) shows the block diagram of a typical FHSS transmitter. First digital data is modulated using some digital-to-analog scheme. This base band signal is then modulated onto a carrier.

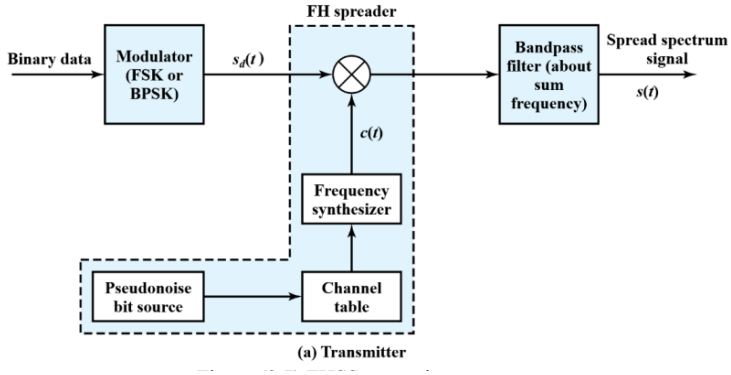


Figure (2.3) FHSS transmitter system

2.1.2 Time Domain Scrambling

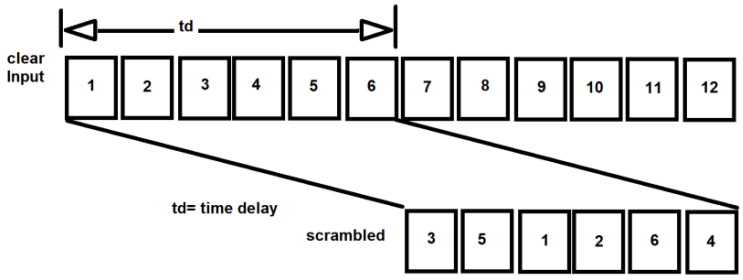
Cutting up long recordings of audio data into little segments is a popular approach. Depending on a secret code that produces a distinct, incomprehensible signal, these frames can then be broadcast

in a different time order. Because the spoken word fragments are no longer in the proper sequence [7].

This method of scrambling divides the voice signal and is segmented into time-domain. The descrambler attempts to reorganize the segments in the original speech signal, whereas the scrambler uses a predefined sequence (code) to change the order of the segments in time domain.

2.1.2.1 Time Segment Permutation (TSP)

Clear voice is first put into a memory device, after which it is read out as a series of time-permuted segments of the original "clear voice" signal after the storage phase is complete [8]. A preselected pseudorandom sequence, which may also be changing over time as shown in Figure (2.4), determines the order of the permutation and its rate of change.



Figure(2.4) Time segment permutation (TSP)

2.1.2.2 Reversed Time Segmentation (RTS)

Reversing the delivery of speech segments in time is one of the most effective ways to obliterate their understandability. Using this approach of reversed time segment encoding, a pattern was obtained [9]. A piece of a voice signal moving normally is depicted in part (a) of figure (2.5), with its segments formed in the order of 1, 2, 3, 4, and so on. The direction of each segment (time

interval) has been changed, but the order in which they are delivered and broadcast to the channel has not been altered, as seen in part (b) of figure (2.5)

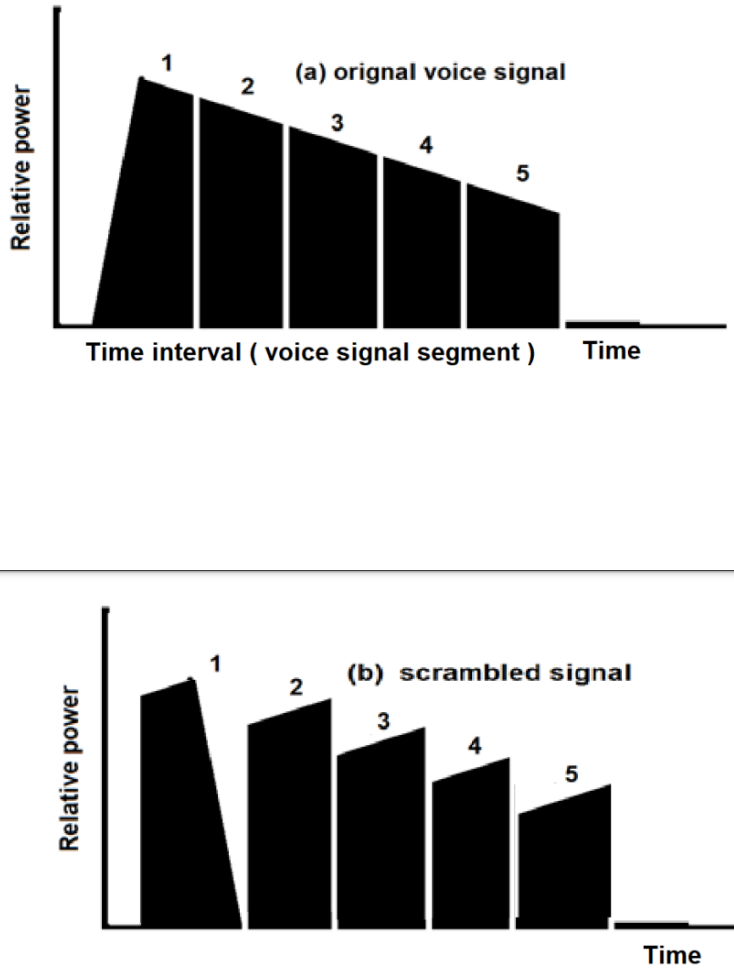


Figure (2.5) Reversed time segmentation

Depending on the system's memory capacity and other implementation factors, the usual duration of these reversed segments can range from 50 to 400 ms. While the direction of their delivery, as output to the channel, has not been reversed.

2.2 Existing solutions and their drawbacks

For communication (voice) security, various solutions have already been offered, however each product has some advantages and disadvantages. The following are some solutions which are already being prepared and being implemented.

- Digital scramblers
- Software-based scramblers
- Multi-band scramblers

2.2.1 Digital scramblers

Digital scramblers use more advanced encryption algorithms, and they require specialized hardware and software, which can be expensive and difficult to maintain. Digital scramblers may also have compatibility issues with different communication channels and devices. Also, a high bandwidth is needed and imposes many restrictions in real time implementation such as stream cipher [10].

2.2.2 Software-based scramblers:

Software-based scramblers are easy to install and use, as they run on standard computers and smartphones. However, they rely on the security of the underlying operating system and hardware, which can be vulnerable to malware, hacking, or other attacks. Software-based scramblers may also have performance issues, such as latency or buffering, which can affect the quality of the voice signal [11].

2.2.3 Multi-band scramblers:

Multi-band scramblers use several frequencies and modulation techniques to encrypt the voice signal, making it more resistant to interception and jamming. However, they require specialized hardware and software, which can be expensive and complex to operate. Multi-band scramblers may also have compatibility issues with different communication channels and devices [12].

Voice scramblers are encryption devices that are used to secure voice communication from eavesdropping and hacking. The literature review reveals that various encryption techniques have been employed to develop voice scramblers, including analog scrambling, digital speech encryption, and frequency inversion. While analog scrambling techniques were widely used in the past, they are no longer considered secure due to their susceptibility to being decoded. The literature suggests that voice scramblers can be effective in securing voice communication, but their effectiveness depends on the encryption technique used and the implementation of the algorithm on microcontrollers. The review also highlights the need for continued research to enhance the security of voice communication considering evolving eavesdropping techniques.

Chapter 3: SYSTEM DESIGN AND DEVELOPMENT

3.1 Introduction

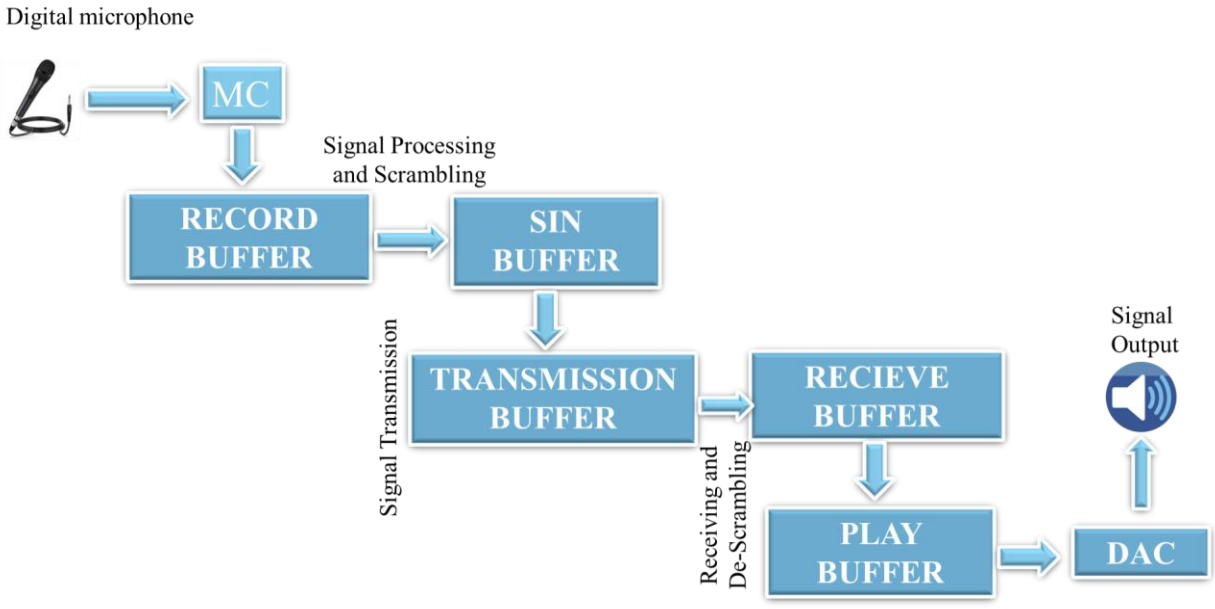
The system design and simulation procedures utilizing research-related software tools are introduced in this chapter.

3.2 Research Design

Review of various voice-scrambling methods, focusing on real-time applications.

Examine the various microcontrollers that are compatible with free development tools and selecting STM32F407G due to its specs suited for real-time applications. Testing the system's functionality in real time using MIC voice input and speaker output voice

3.3 Proposed System



The proposed system is aimed at real-time scrambling and descrambling of voice signals using the STM32F407G microcontroller. The input voice signal is transmitted through the digital microphone embedded on the STM32F407G microcontroller. This eliminates the need for an external analog microphone or amplifier for the transmission of the input voice signal. The analog voice is converted into digital form and sent to the microchip for signal processing.

The microcontroller is equipped with five buffers, each consisting of 32 bits in total. These buffers serve different purposes within the system. Firstly, there is a record buffer that is responsible for storing the incoming signal for recording. Additionally, there is a sin buffer that contains a pre-defined 32-bit pattern specified in the program. This pattern is used to perform an AND operation with the 32 bits in the record buffer, effectively scrambling the recorded signal. The transmission buffer is used to store the scrambled bits for transmission to the designated port. At the receiver end, the received bits are stored in the receive buffer. Finally, there is a play buffer that also contains the pre-defined 32-bit pattern used by the sin buffer. The play buffer performs the inverse operation of the sin buffer, allowing for the recovery of the original audio signal.

When scrambling is enabled in the program design, the 32-bit signal sent from the transmission buffer is received by the receive buffer and directly sent to the digital-to-analog converter (DAC) for conversion. The converted signal is then amplified, enabling the playback of the scrambled voice. Conversely, when descrambling is enabled, the signal received by the record buffer is sent to the play buffer. The play buffer uses the pre-defined pattern to reverse the scrambling operation, recovering the original audio signal. The signal is then sent to the DAC for the necessary digital-to-analog conversion and subsequently amplified for playback.

The entire system operates in a loop, continuously scrambling and descrambling the audio signal based on the specified requirements.

Chapter 4: Evaluation and Analysis of the code

The code is a firmware for the STM32F4-Discovery board to play back recorded audio data from the microphone using an external I2S DAC chip. The code includes the initialization of LEDs and the User_Button, setting the volume of the output device, and the implementation of the sound playback using interrupts. The sound data is sent via UART for further processing.

The code is for playing audio data through an external I2S DAC chip using a microphone input. The program includes the necessary header files and defines the required functions to configure the UART communication, to send and receive data, to initialize LEDs and buttons, and to handle the transfer of audio data. The main function initializes the system and starts recording and playing audio data using interruptions.

The program starts by defining the necessary header files for the program, such as main.h and stm32f4_discovery_audio_codec.h, and the waverecorder.h file, which includes the function to record audio data. The program defines the necessary functions to configure the UART communication, which is used to send data from the microcontroller to a serial monitor, and to receive data from the monitor.

The program also initializes the LEDs and user button on the STM32F4-Discovery board, which are used to indicate the status of the program and to trigger the recording and playback of audio data. The program then initializes the audio codec using the EVAL_AUDIO_Init() function, which configures the output device, sets the volume level, and selects the audio frequency.

The program then starts recording audio data using the simple_rec_start() function, which initializes the microphone and starts the recording process. The recorded data is stored in a buffer

called RecBuf, which is an array of 16-bit integers with a size of PCM_OUT_SIZE, defined in the waverecorder.h file.

The program then starts playing back the recorded audio data using the EVAL_AUDIO_Play() function, which starts the playback process by sending the first 32 samples of the audio data to the I2S DAC chip. The program then enters an infinite loop that continuously sends data to a serial monitor using the UART communication.

Inside the infinite loop, the program checks if there is enough data in the buffer to be sent to the serial monitor. If there is, the program converts the 16-bit audio data to 8-bit data and sends it to the monitor using the USART_SendData_s() function. The program also applies a simple algorithm to make the audio signal pseudo stereo by splitting the 16-bit audio data into two 8-bit data, with one data containing the most significant bits and the other data containing the least significant bits.

After sending the audio data to the serial monitor, the program increments the counter to check the next set of audio data in the buffer. If there is no more data to be sent, the program resets the counter and starts again from the beginning of the buffer.

In summary, the program is designed to record audio data using a microphone input and play it back through an external I2S DAC chip. The program uses interrupts to handle the recording and playback of audio data, and the UART communication is used to send the audio data to a serial monitor for debugging purposes. The program also includes functions to initialize LEDs and buttons, which can be used to trigger the recording and playback of audio data.

The use of interrupts to handle the recording and playback of audio data is a good approach, as it allows the program to operate asynchronously and reduces the load on the CPU. The program

also uses UART communication to send audio data to a serial monitor for debugging purposes, which is a useful feature for troubleshooting and testing.

Chapter 5: Conclusion

In conclusion, this thesis explores the development and implementation of a voice scrambler based on microcontroller STMF407G. The design of the voice scrambler involved the use of digital signal processing techniques, such as AND operation using 32-bit buffers, to enhance the privacy and security of voice communication. The performance of the voice scrambler was evaluated through various tests, which demonstrated its ability to effectively scramble voice signals while maintaining high-quality sound transmission.

The voice scrambler developed in this thesis has significant implications for various fields, including defense, law enforcement, and commercial industries, where secure communication channels are critical for safeguarding sensitive information. This project contributes to the development of cost-effective and efficient voice scramblers that can be integrated with various communication channels, such as phone lines, VoIP networks, and radio communication.

In summary, this thesis demonstrates the feasibility and effectiveness of using microcontroller-based voice scramblers for secure voice communication. Future research could focus on improving the real-time processing capabilities of the voice scrambler and enhancing its compatibility with a wider range of audio hardware devices.

Chapter 6: Future Work

In terms of future work, several areas have been identified for further exploration. These include the implementation of additional scrambling algorithms such as XOR or frequency shifting, integration with additional communication channels such as satellite communication or mobile communication networks, optimization of real-time processing capabilities, and compatibility with additional audio hardware devices. Additionally, the implementation of additional security features such as encryption algorithms or user authentication protocols could further enhance the security of voice communication. Overall, the voice scrambler project has shown promise in providing a secure and private voice communication system, and future work could further enhance its capabilities and accessibility.

References and Work Cite

[1] K. R. Chung, "Audio scrambler," 2020.

[2] S. B. Sadkhab, A. M. Raheema, and S. M. A. Sattar, "Design and implementation voice scrambling model based on hybrid chaotic signals," in *2019 First International Conference of Computer and Applied Sciences (CAS)*, 2019: IEEE, pp. 193-198.

[3] R. Sokullu, "People/Animal Counting–Integrated Sensor Based and Wifi/Machine Learning Based System," in *2022 8th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE)*, 2022: IEEE, pp. 1-4.

[4] G. Etter, S. van der Veldt, J. Choi, and S. Williams, "Optogenetic frequency scrambling of hippocampal theta oscillations dissociates working memory retrieval from hippocampal spatiotemporal codes," *Nature Communications*, vol. 14, no. 1, p. 410, 2023.

[5] T. Lu, T. Marin, Y. Zhuo, Y.-F. Chen, and C. Ma, "Nonuniform fast Fourier transform on TPUs," in *2021 ieee 18th international symposium on biomedical imaging (isbi)*, 2021: IEEE, pp. 783-787.

[6] D. Ban *et al.*, "A novel optical frequency-hopping system based on DFB laser integrated with an EA modulator," *IEEE Photonics Journal*, vol. 11, no. 6, pp. 1-8, 2019.

[7] Z. A. Hasan, S. M. Hadi, and W. A. Mahmoud, "Time domain speech scrambler based on particle swarm optimization," *Pollack Periodica*, vol. 18, no. 1, pp. 161-166, 2023.

[8] G. Lancia and M. Dalpasso, "Algorithmic strategies for a fast exploration of the tsp 4-opt neighborhood," *Advances in Optimization and Decision Science for Society, Services and Enterprises: ODS, Genoa, Italy, September 4-7, 2019*, pp. 457-470, 2019.

[9] M. S. D. Suliman, "DesigningofReal Time Voice Scrambler/DescramblerBased onMicrocontroller," Sudan University of Science and Technology, 2021.

[10] J. D. Paul, "Re-creating the sigsaly quantizer: This 1943 analog-to-digital converter gave the allies an unbreakable scrambler-[Resources]," *IEEE Spectrum*, vol. 56, no. 2, pp. 16-17, 2019.

[11] A. K. Chaurasiya and P. Yadav, "Live Broadcast Data Processing & Security by Software Based Encryption," in *2019 Fifteenth International Conference on Information Processing (ICINPRO)*, 2019: IEEE, pp. 1-6.

[12] H. Minami *et al.*, "Experimental Demonstration of Cascadable PPLN-Based Inter-Band Wavelength Converters for Band-Switchable Multi-Band Optical Cross-Connect," in *Optical Fiber Communication Conference*, 2023: Optica Publishing Group, p. M4G. 1.

ORIGINALITY REPORT

15%	15%	0%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

14%

★ repository.sustech.edu

Internet Source

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		