

***PUBLICATION MANUSCRIPT***

**SECURE FILES USING VIGENERE CIPHER AND PLAYFAIR  
CIPHER METHODS**

Achmad Nur Zahir S, Muhammad Taufiq Sumadi, Faldi

**SUBMITTED BY:  
ACHMAD NUR ZAHIR S  
1911102441143**



**INFORMATICS ENGINEERING S1 STUDY PROGRAM  
FACULTY OF SCIENCE AND TECHNOLOGY  
UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR  
SAMARINDA  
2023**

*Publication Manuscript*

## **Secure Files using Vigenere Cipher and Playfair Cipher Methods**

Achmad Nur Zahir S, Muhammad Taufiq Sumadi, Faldi

**Submitted By:**

**Achmad Nur Zahir S**

**1911102441143**



**INFORMATICS ENGINEERING S1 STUDY PROGRAM  
FACULTY OF SCIENCE AND TECHNOLOGY  
UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR  
SAMARINDA**

**2023**

**Publication Manuscript**

**Approval Page**

**SECURE AUDIO FILES USING VIGENERE CIPHER AND PLAYFAIR  
CIPHER METHODS**

ARRANGED BY :

**Achmad Nur Zahir S**

**19111024411143**

It has been approved for publication,

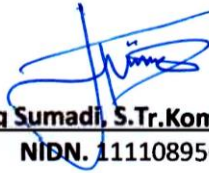
On July 19, 2023

Examiner



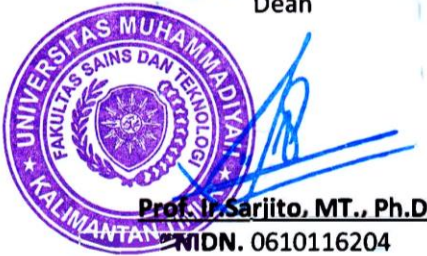
**Faldi, S.Kom., M.TI**  
NIDN. 1121079101

Advisor



**M. Taufiq Sumadi, S.Tr.Kom., M.Tr.Kom**  
NIDN. 1111089501

Dean



**Prof. Dr. Sarjito, MT., Ph.D**  
NIDN. 0610116204

Head of Study Program



**Asslia Johar Latipah, S.Kom., M.Cs**  
NIDN. 1124098902

# Secure Audio Files Using Vigenere Cipher and Playfair Cipher

Muhammad Taufiq Sumadi<sup>1</sup>, Achmad Nur Zahir S<sup>2</sup>, Faldi<sup>3\*</sup>)

<sup>1,2,3</sup>Program Study Informatics Engineering, Faculty of Science and Technology,  
Universitas Muhammadiyah Kalimantan Timur, Samarinda, Indonesia  
email: <sup>1</sup>mts653@umkt.ac.id, <sup>2</sup>anzahirs31@gmail.com, <sup>3</sup>fal146@umkt.ac.id

**Abstract** – This study aims to maintain the confidentiality of audio files sent using a combination of the Playfair cipher and Vigenere cipher methods. In this research, the object of research is an audio file with the extension wave or \*.wav. This research requires several stages, including Audio Data Analysis, Determination of System Architecture, Implementation, Testing, and Results Analysis. The results of this study indicate that in the Vigenere Cipher 256 Encryption in audio wave files, the audio messages conveyed sound unclear or have no meaning. From the 6 trial datasets based on analysis of MAE and PSNR, the average value of the encryption process at PSNR was 28.345, and MAE was 97.0625. The average value of the decryption process on PSNR and MAE is 0.0, indicating that the decryption process is successful. The speed of the encryption and decryption process is affected by the audio file's size, which means that the larger the file size, the longer the encryption and decryption time.

**Keywords** – Cryptography, Audio, Confidentiality, Vigenere Cipher, Playfair Cipher.

**Abstrak** – Penelitian ini bertujuan untuk menjaga kerahasiaan file audio yang dikirim menggunakan kombinasi metode Playfair cipher dan Vigenere cipher. Pada penelitian ini objek penelitian adalah file audio dengan ekstensi wave atau \*.wav. Penelitian ini memerlukan beberapa tahapan, antara lain Analisis Data Audio, Penentuan Arsitektur Sistem, Implementasi, Pengujian, dan Analisis Hasil. Hasil penelitian ini menunjukkan bahwa pada Enkripsi Vigenere Cipher 256 pada file gelombang audio, pesan audio yang disampaikan terdengar tidak jelas atau tidak bermakna. Dari 6 dataset percobaan berdasarkan analisis MAE dan PSNR diperoleh nilai rata-rata proses enkripsi di PSNR adalah 28.345, dan MAE adalah 97.0625. Nilai rata-rata proses dekripsi pada PSNR dan MAE adalah 0.0 yang menandakan bahwa proses dekripsi berhasil. Kecepatan proses enkripsi dan dekripsi dipengaruhi oleh ukuran file audio, yang artinya semakin besar ukuran file maka semakin lama waktu enkripsi dan dekripsinya.

**Kata Kunci** – Kriptografi, Audio, Kerahasiaan, Vigenere Cipher, Playfair Cipher.

## I. INTRODUCTION

The tremendous growth of digital technology, especially in Internet and communication technology, has facilitated multimedia data exchange. Multimedia data is used in real-time voice and video conferencing, air traffic and control,

broadcast monitoring, and voice-activated machine control and operation commands. However, sending confidential or sensitive information over networks and communication systems can be lethal as these networks are insecure and vulnerable to virus attacks.[1]

Audio, or sound, is a wave that contains many vital components (amplitude, wavelength, and frequency) that can make one sound into another. Computers can make sense of analog sound through analog-to-digital conversion (ADC). This process is performed by computer hardware called a sound card or sound card. Commonly used audio file formats are mp3, wav, and flax.[2] Wave format (\*.WAV) is a sound file format widely used in Windows operating systems for games and multimedia purposes. A wave is a raw form in which sound is recorded directly and digitally quantized. Ease of creation and processing This basic file format supports no compression by default and is known as PCM (Pulse Code Modulation).[3]

Encryption of audio data is a more difficult and complex process than the techniques used for text data. The US Department of Defense began work on voice encryption in the late 1940s. It was initially used in World War II for secure communications, so the enemy could not understand conversations between military personnel. The idea was first introduced by simply adding noise to the audio signal. The central concept is that the noise signal is added by engaging the sound signal and the recorded sound along the reception point, after which the noise signal is reduced to induce the original sound signal. However, since the same noise signal is required at both ends, the noise signal is generated in pairs for the transmitter and receiver. [4]

With the rapid growth of communication technology, protecting audio from hackers has become an essential issue for engineers. Therefore, to protect your valuable information, you need a way to encrypt information and data before transmission securely.[1] Cryptography is the science (or skill) of protecting sensitive information from intruders and hackers who want to use it for illegal purposes. It can be defined as ensuring the confidentiality of communications between both ends of a connection. It mainly includes encryption and decryption. Encryption deals with the encrypted content of a secure message so that it cannot be read or deciphered by unauthorized persons or programs.[5]

Vigenere Cipher is a classic cryptography that hides a message in plain text using a substitution technique that

\*) **corresponding author:** Faldi  
Email: fal146@umkt.ac.id

changes each character to another based on the key used. The Vigenere cipher is also an algorithm that uses a symmetric key and repeats the characters in the key until all the characters in the message have been processed.[6] Playfair Cipher is a classical cryptographic algorithm with a Polygram cipher that converts plaintext into Polygram form and performs Polygram encryption and decryption processes. The encryption key comprises 25 characters arranged in a 5x5 grid, excluding the letter J from the alphabet. So the key probability is 25.

In this study, the authors used the Vigenere and Playfair Cipher methods to design a system that can protect Wave format audio and maintain the confidentiality of audio files sent using 256 ASCII tables.

## II. RELATED RESEARCH

In research [1], Due to the extraordinary growth in digital technology, especially in the internet and communication science and engineering, sharing multimedia data becomes easier to transmit through networks and communication systems, which can be fatal because these networks are insecure and vulnerable to virus attacks. It is, therefore, mandatory to encrypt information or data before sending it to protect valuable information. With the results, the proposed algorithm is best suited for audio signal encryption in terms of complexity, security, and loss. It is strong against compression attacks and satisfies all encryption parameters, such as correlation coefficient, SCR, and UAC. It can be implemented in audio watermarking systems using the transform domain approach.

In research [2], The rapid development of information technology has made using MP3 format digital audio files standard and makes distribution easier. Therefore, you need an application that protects MP3 files by encrypting and decrypting them. As a result, applying encryption algorithms to audio files can save data from modification.

In research [3], Due to the rapidly increasing development of information technology, audio data transmission has also increased because it facilitates the exchange of information in various places. It is necessary to secure confidential data so that third parties do not know it. The result is that the software can encode audio data by applying the RSA algorithm and the WAV audio data structure, and the size of the WAV audio file becomes larger after being encrypted using the RSA algorithm based on the key size used.

In research [5], Secret information and its transmission method have occupied the most critical position since ancient times. The methods used to encrypt data varied, and the algorithms used in this domain were developed into what they are today. Due to the significant development and electronic revolution, the emergence of biometric information, and the introduction of this concept into remote data encryption systems, bio-encryption has crystallized. Based on the above problems, the author proposes to build an audio file encryption/decryption algorithm based on discrete wavelet transform (DWT) and biometric features and uses several hand geometry measurement properties to secure audio files. The result is a new proposed algorithm for audio file encryption/decryption based on discrete wavelet transform (DWT) and biometric features to provide a high level of confidentiality and reliability. Subjective and objective measures are applied based on the generated signal.

In research [9], Because audio files are often used as private messages, encryption techniques are needed to change the form of the data so that it is not easily read or seen by others. The result is that the time to carry out the encryption process depends on the file size and key digits used. Encryption cannot work in the audio format registered in the application. The larger the audio file size, the longer the estimated time for the encryption process. This is due to the length of the calculation process, and encryption of audio files with the RSA method results in a different file that can't be used at all.

In research [10], Because information security is essential, it is necessary to maintain its authenticity and integrity if it is to be sent because third parties can intercept and know the data during transmission. The result is that the paillier cryptosystem algorithm can encrypt audio, and audio encryption is very dependent on the size of the audio itself. The larger the audio size, the longer it takes for encryption.[10]

In research [11], Multimedia data security is a big challenge for open communication systems that various types of attacks can threaten. The primary tool for achieving this security is encryption. In the study, the method used consisted of three phases with three secret keys and two multi-chaotic pseudo-random generators, which were newly designed and evaluated using different measurements, including signal-to-noise ratio (SNR), peak signal-to-noise ratio (PSNR), correlation coefficient, histogram, key sensitivity, UACI, NSCR, RMS, and CF. The results prove that the proposed scheme is highly secure and more substantial than many similar recent audio encryption schemes against various types of attacks.

In research [12], Chaotic-based cryptosystems have many properties that make them suitable for encrypting multimedia such as images, videos, or audio data. Unlike text messages and pictures, audio information has a higher repetition and a stronger correlation between samples. Many scholars have made efforts to investigate audio encryption algorithms but have yet to find the algorithm to be called the best that is algorithm if it achieves a good balance between performance and security. The test results show that the proposed audio algorithm is secure because of its sizeable key space, uniform histogram, low PSNR, low correlation, high entropy, and high MSR, indicating that the algorithm is a good choice for encrypting audio and other multimedia data such as images and videos.

In research [13], When data is transferred from one person to another, a very high level of security is required. For this reason, the author wants to build cryptography using the SHA-256 hashing algorithm to help fight various hackers who will not be able to decipher confidential data and use Discrete Wavelet Transform (DWT) to improve performance efficiency. With experimental results and analyses such as PSNR, correlation, and histogram, we can conclude that the design is robust and very useful for audio encryption.

In research [14], Because one of the things that are important to ensure the confidentiality of data or information is encryption. So we need observation of details in the form of audio recordings in MP3 format using cryptography. The results show that the Skipjack Algorithm provides robust security to prevent the audio from being spread to unauthorized people.

In research [15], Because data and information are essential commodities for individuals and organizations, information can be presented in text, images, audio, video, or a mixture thereof. Some information is available and accessible to the public, whereas some are confidential. To secure information, cryptography can be applied. Cryptography is the science of keeping secrets. Therefore, the author wants to spread the vigenere cipher method with circular bit shifts to secure information data in images. Image encryption using the Vigenere cipher with circular bit shifts gives better results than using the Vigenere cipher alone. Vigenere cipher with circular shift bits produces better randomness images, resulting in images that are difficult to recognize.

### III. RESEARCH METHODOLOGY

#### A. Subjects and Research Objects

A research subject is a person, place, or thing considered in the context of a research object, and a research object is a place of a research study.

##### 1. Subject Research

This research aims to analyze and implement the security of audio files using the Vigenere cipher and the Playfair cipher. In this study, the objects of the investigation were audio files with wave extensions.

##### 2. Objects Research

This research was conducted at Universitas Muhammadiyah Kalimantan Timur, Jalan. Ir. H. Juanda No.15, Sidodadi, Kec. Samarinda Ulu, Kota Samarinda, Kalimantan Timur 75124.

#### B. Data Collection

A needs analysis phase is conducted to find devices that will be used to implement security for audio files using the Vigenere Cipher and Playfair Cipher methods.

##### 1. Hardware

Before building the research system, we need to prepare some hardware. The list of devices used in this study is written in Table 3.1

TABLE I  
HARDWARE

No	Name	Specification
1	Computer / Laptop	Processor: AMD Ryzen 7 4800H with Radeon Graphics (2.90 GHz), RAM: 16 GB, Storage: 500 GB, Graphic Card: NVIDIA GeForce RTX 2060, DirectX: 12

##### 2. Software

Before building the system, we will use some software. The list of software used in this study is written in Table 3.2

TABLE II  
SOFTWARE

No	Name	Specification	Function
1	Audio File	Extension *.wav	As the object of research that will be applied to the system to be designed

2	PyCharm Community Edition	Version 2022.2.3	As an editor, write down the Python programming language that will be applied to the system will be designed.
3	Python	Python 3	As a programming language that will be applied to the research system
4	HxD Hex Editor	2.5.0.0 (x86-64)	As an application to read audio hexadecimal values

#### C. Research Methods

This study requires several steps, including Analysis of voice data, determination of system architecture, implementation, system testing, and analysis of results. Fig 1 shows the flow chart for this study.

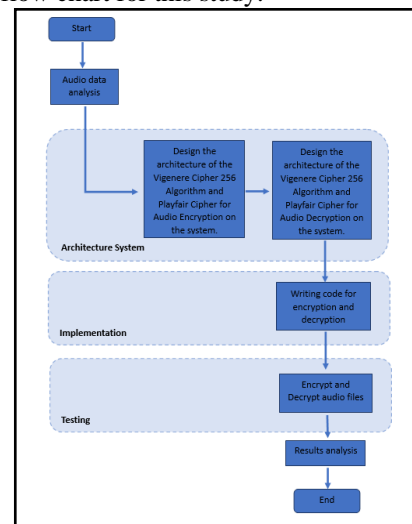


Fig. 1 Research Methods

##### 1. Audio Data Analysis

In audio data analysis, an understanding of the characteristics, architecture, and audio data is carried out at this stage to produce information that will facilitate system design and increase the success of securing audio files.

Because the Wave file is a substance or part of the RIFF file, it inherits the RIFF file structure. This data section uses the Vigenere Cipher and Playfair Cipher methods to perform encryption and decryption.

##### 2. System Architecture

Determining the architecture system is one way to facilitate building the system described in Figure 3.2. This image explains how the audio file security system works using the Vigenere cipher and the Playfair cipher created.

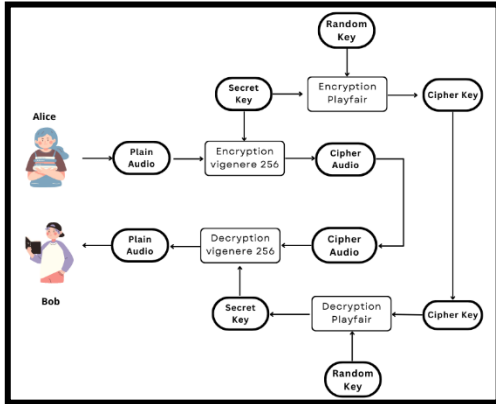


Fig. 2 System Architecture

In Fig 2, it is explained that when Alice tries to send audio files to Bob, the audio is encrypted using Vigenere Cipher with the key determined by Alice. After that, the Secret Key that Alice has determined will be encrypted using a Playfair Cipher with a random key and produce a cipher Key. Cipher Key and Cipher Audio will be sent to Bob simultaneously. After getting a cipher key and audio cipher, Bob will decrypt the Playfair Cipher method against the Cipher Key first using the Random Key that has been approved by Alice and Bob and will produce a Secret Key, after that Bob will decrypt the audio with the Vigenere Cipher method with the Secret Key and will produce plain audio. Then Bob can find out the information given by Alice to him.

### 3. Implementation

Writing programming code for encryption and decryption with the Vigenere Cipher and Playfair Cipher using Python is done during the implementation phase.

### 4. Testing

Audio files are encrypted and decrypted during the test phase using the previously created system with the Vigenere Cipher and Playfair Cipher methods assigned.

### 5. Results Analysis

In this phase, some encrypted and decrypted audio samples are analyzed, such as time, size, success rate, and errors of encrypted/decrypted audio files.

## IV. RESULT AND DISCUSSION

This chapter discusses the Vigenere Cipher 256 and Playfair Cipher algorithms in encrypting and decrypting audio wave files. MAE (Mean Absolute Error) is used as a measuring parameter to determine the decryption process's resulting performance and accuracy and how much randomization the data is when the encryption process is carried out. PSNR (Peak Signal Noise Ratio) is used to compare the maximum signal intensity over time with the noise base to determine how much it changes during the encryption and decryption process. MSE (Mean Squared Error) measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. Histogram analysis is an accurate metric for measuring the quality of an encrypted audio signal. A good encryption scheme should encrypt the original audio file into random sound. A spectrogram is a visual representation of the spectrum of frequencies found in a signal as it varies with time. Spectrograms are very helpful for vibration analysis in changing environments.

The Sample.wav audio file obtained in the dataset will be used as a research object or basic audio that will be used for

encryption and decryption. To read audio hexadecimal values, you will use the HxD application.

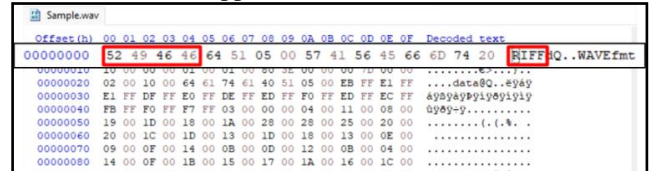


Fig 3.ChunkID

In Figure 3, we can see audio containing binary hexadecimal, which has a structure starting with the RIFF header consisting of ChunkID starting from byte 0 - 4, including the value 52 49 46 46 with decoded text "RIFF".

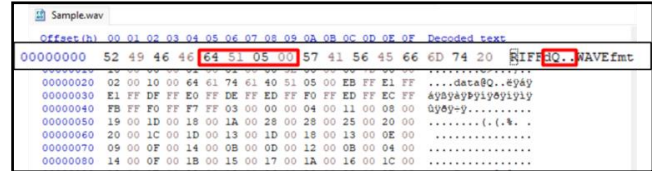


Fig 4. ChunkSize

In Figure 4, ChunkSize starts from bytes 4 - 8 containing 64 51 05 00, Usually 8 bytes (32-bit integer) of total file size - in bytes.

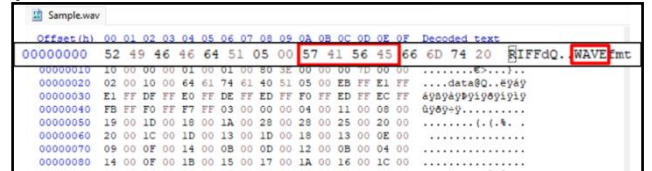


Fig 5.Format

And in Figure 5, the Format section starts from bytes 8-12, including the value 57 41 56 45. Hexadecimal representation of the ASCII form of the letters "WAVE". Therefore, it is 57 41 56 45 for every WAV file.

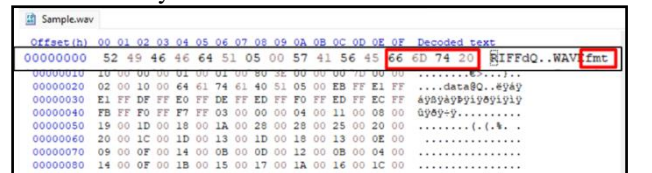


Fig 6. SubChunkID

In Figure 6, After RIFF is the "fmt " part consisting of SubchunkID starting from bytes 12 – 16 containing the value 66 6D 74 20. Hexadecimal representation of the ASCII form of the letters "fmt "(note the blank space here).

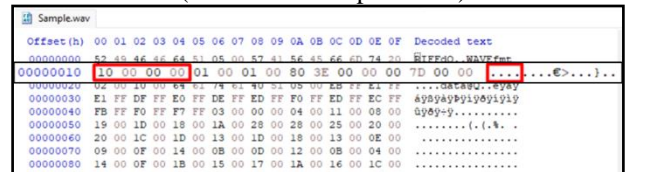


Fig 7. SubchunkSize

In Figure 7, SubchunkSize starts from bytes 16 – 20 containing value 10 00 00 00, subchunk size 16, that is the sum of the rest subchunk size from audio format to BitsPerSample (2+2+4+4+2+2=16).

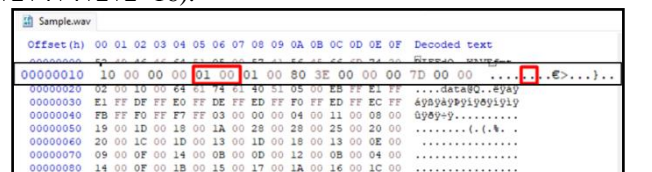


Fig 8. AudioFormat

In Figure 8, AudioFormat starts from bytes 20 -22 containing values 01 00. 1 for PCM, other values represent other forms of compression. Litter endian, so the hexadecimal 0001 is 1 in decimal.

```
Sample.wav
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 52 49 46 46 64 51 05 00 57 41 56 45 66 6D 74 20 RIFFDQ..WAVEfmt
00000010 10 00 00 00 01 00 01 00 80 3E 00 00 00 7D 00 00 .....<E>...}...
00000020 02 00 10 00 01 00 01 00 64 61 74 61 40 51 05 00 EB FF E1 FF ...data@Q..eYáY
00000030 E1 FF DF FF E0 FF DE FF ED FF EC FF EF FF ED FF EC FF 4949494949494949
00000040 FB FF F0 FF F7 FF F3 03 00 00 04 00 11 00 08 00 0949-y.....
00000050 19 00 1D 00 18 00 1A 00 28 00 28 00 25 00 20 00 .....(.(.%.
00000060 20 00 1C 00 1D 00 13 00 1D 00 18 00 13 00 0E 00 .....(.(.%.
00000070 09 00 0F 00 14 00 0B 00 0D 00 12 00 0B 00 04 00 .....
00000080 14 00 0F 00 1B 00 15 00 17 00 1A 00 16 00 1C 00 .....
```

Fig 9. NumChannel

In Figure 9, NumChannel starts from bytes 22 – 24 which contain values 01 00. Litter endian, so the hexadecimal 00 01 is 1 in decimal, 1 for mono, 2 for stereo.

```
Sample.wav
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 52 49 46 46 64 51 05 00 57 41 56 45 66 6D 74 20 RIFFDQ..WAVEfmt
00000010 10 00 00 00 01 00 01 00 80 3E 00 00 00 7D 00 00 .....<E>...}...
00000020 02 00 10 00 01 00 01 00 64 61 74 61 40 51 05 00 EB FF E1 FF ...data@Q..eYáY
00000030 E1 FF DF FF E0 FF DE FF ED FF EC FF EF FF ED FF EC FF 4949494949494949
00000040 FB FF F0 FF F7 FF F3 03 00 00 04 00 11 00 08 00 0949-y.....
00000050 19 00 1D 00 18 00 1A 00 28 00 28 00 25 00 20 00 .....(.(.%.
00000060 20 00 1C 00 1D 00 13 00 1D 00 18 00 13 00 0E 00 .....(.(.%.
00000070 09 00 0F 00 14 00 0B 00 0D 00 12 00 0B 00 04 00 .....
00000080 14 00 0F 00 1B 00 15 00 17 00 1A 00 16 00 1C 00 .....
```

Fig 10. SampleRate

In Figure 10, SampleRate starts from bytes 24 – 28 containing the value 80 3E 00 00. Sample Rate = Sample per second or Hertz. FYI, 80 3E 00 00 stands for 16000 Hz (little endian 00003E80). Typical values are 44100 (CD), 48000 (DAT).

```
Sample.wav
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 52 49 46 46 64 51 05 00 57 41 56 45 66 6D 74 20 RIFFDQ..WAVEfmt
00000010 10 00 00 00 01 00 01 00 80 3E 00 00 00 7D 00 00 .....<E>...}...
00000020 02 00 10 00 01 00 01 00 64 61 74 61 40 51 05 00 EB FF E1 FF ...data@Q..eYáY
00000030 E1 FF DF FF E0 FF DE FF ED FF EC FF EF FF ED FF EC FF 4949494949494949
00000040 FB FF F0 FF F7 FF F3 03 00 00 04 00 11 00 08 00 0949-y.....
00000050 19 00 1D 00 18 00 1A 00 28 00 28 00 25 00 20 00 .....(.(.%.
00000060 20 00 1C 00 1D 00 13 00 1D 00 18 00 13 00 0E 00 .....(.(.%.
00000070 09 00 0F 00 14 00 0B 00 0D 00 12 00 0B 00 04 00 .....
00000080 14 00 0F 00 1B 00 15 00 17 00 1A 00 16 00 1C 00 .....
```

Fig 11. ByteRate

In Figure 11, ByteRate starts from bytes 28 – 32 containing values 00 7D 00 00. ByteRates are obtained from (Sample Rate \* Bits Per Sample \* Channel Numbers) / 8. So ByteRate = 32000.

```
Sample.wav
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 52 49 46 46 64 51 05 00 57 41 56 45 66 6D 74 20 RIFFDQ..WAVEfmt
00000010 10 00 00 00 01 00 01 00 80 3E 00 00 00 7D 00 00 .....<E>...}...
00000020 02 00 10 00 01 00 01 00 64 61 74 61 40 51 05 00 EB FF E1 FF ...data@Q..eYáY
00000030 E1 FF DF FF E0 FF DE FF ED FF EC FF EF FF ED FF EC FF 4949494949494949
00000040 FB FF F0 FF F7 FF F3 03 00 00 04 00 11 00 08 00 0949-y.....
00000050 19 00 1D 00 18 00 1A 00 28 00 28 00 25 00 20 00 .....(.(.%.
00000060 20 00 1C 00 1D 00 13 00 1D 00 18 00 13 00 0E 00 .....(.(.%.
00000070 09 00 0F 00 14 00 0B 00 0D 00 12 00 0B 00 04 00 .....
00000080 14 00 0F 00 1B 00 15 00 17 00 1A 00 16 00 1C 00 .....
```

Fig 12. BlockAlign

In Figure 12, BlockAlign starts from bytes 32 – 34 containing values 02 00 (little Indian 0002). Usually contains from (NumChannels \* BitsPerSample) / 8. So the hexadecimal 00 02 is 2 in decimal

```
Sample.wav
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 52 49 46 46 64 51 05 00 57 41 56 45 66 6D 74 20 RIFFDQ..WAVEfmt
00000010 10 00 00 00 01 00 01 00 80 3E 00 00 00 7D 00 00 .....<E>...}...
00000020 02 00 10 00 01 00 01 00 64 61 74 61 40 51 05 00 EB FF E1 FF ...data@Q..eYáY
00000030 E1 FF DF FF E0 FF DE FF ED FF EC FF EF FF ED FF EC FF 4949494949494949
00000040 FB FF F0 FF F7 FF F3 03 00 00 04 00 11 00 08 00 0949-y.....
00000050 19 00 1D 00 18 00 1A 00 28 00 28 00 25 00 20 00 .....(.(.%.
00000060 20 00 1C 00 1D 00 13 00 1D 00 18 00 13 00 0E 00 .....(.(.%.
00000070 09 00 0F 00 14 00 0B 00 0D 00 12 00 0B 00 04 00 .....
00000080 14 00 0F 00 1B 00 15 00 17 00 1A 00 16 00 1C 00 .....
```

Fig 13. BitsPerSample

And in figure 13, BitsPerSample starting from bytes 34 – 36 containing 10 00. Usually contains 16 bits = 1000 (little endian 0010); 32 bits = 2000 (little endian 0020); etc. or 8 bits = 8, 16 bits = 16, etc.

```
Sample.wav
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 52 49 46 46 64 51 05 00 57 41 56 45 66 6D 74 20 RIFFDQ..WAVEfmt
00000010 10 00 00 00 01 00 01 00 80 3E 00 00 00 7D 00 00 .....<E>...}...
00000020 02 00 10 00 01 00 01 00 64 61 74 61 40 51 05 00 EB FF E1 FF ...data@Q..eYáY
00000030 E1 FF DF FF E0 FF DE FF ED FF EC FF EF FF ED FF EC FF 4949494949494949
00000040 FB FF F0 FF F7 FF F3 03 00 00 04 00 11 00 08 00 0949-y.....
00000050 19 00 1D 00 18 00 1A 00 28 00 28 00 25 00 20 00 .....(.(.%.
00000060 20 00 1C 00 1D 00 13 00 1D 00 18 00 13 00 0E 00 .....(.(.%.
00000070 09 00 0F 00 14 00 0B 00 0D 00 12 00 0B 00 04 00 .....
00000080 14 00 0F 00 1B 00 15 00 17 00 1A 00 16 00 1C 00 .....
```

Fig 14. SubchunkID2

In Figure 14, After "fmt" the last part of the wave structure is Data which consists of Subchunk2ID starting bytes 36 – 40 containing the values 64 62 74 61. Hexadecimal representation of the ASCII form of the letters "DATA".

```
Sample.wav
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 52 49 46 46 64 51 05 00 57 41 56 45 66 6D 74 20 RIFFDQ..WAVEfmt
00000010 10 00 00 00 01 00 01 00 80 3E 00 00 00 7D 00 00 .....<E>...}...
00000020 02 00 10 00 01 00 01 00 64 61 74 61 40 51 05 00 EB FF E1 FF ...data@Q..eYáY
00000030 E1 FF DF FF E0 FF DE FF ED FF EC FF EF FF ED FF EC FF 4949494949494949
00000040 FB FF F0 FF F7 FF F3 03 00 00 04 00 11 00 08 00 0949-y.....
00000050 19 00 1D 00 18 00 1A 00 28 00 28 00 25 00 20 00 .....(.(.%.
00000060 20 00 1C 00 1D 00 13 00 1D 00 18 00 13 00 0E 00 .....(.(.%.
00000070 09 00 0F 00 14 00 0B 00 0D 00 12 00 0B 00 04 00 .....
00000080 14 00 0F 00 1B 00 15 00 17 00 1A 00 16 00 1C 00 .....
```

Fig 15. Subchunk2Size

In Figure 15, Subchunk2Size starts from bytes 40 – 44 containing the value 40 51 05 00. Usually contains the Number of bytes in the data obtained from (NumSamples \* NumChannels \* BitsPerSample) / 8.

```
Sample.wav
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 52 49 46 46 64 51 05 00 57 41 56 45 66 6D 74 20 RIFFDQ..WAVEfmt
00000010 10 00 00 00 01 00 01 00 80 3E 00 00 00 7D 00 00 .....<E>...}...
00000020 02 00 10 00 01 00 01 00 64 61 74 61 40 51 05 00 EB FF E1 FF ...data@Q..eYáY
00000030 E1 FF DF FF E0 FF DE FF ED FF EC FF EF FF ED FF EC FF 4949494949494949
00000040 FB FF F0 FF F7 FF F3 03 00 00 04 00 11 00 08 00 0949-y.....
00000050 19 00 1D 00 18 00 1A 00 28 00 28 00 25 00 20 00 .....(.(.%.
00000060 20 00 1C 00 1D 00 13 00 1D 00 18 00 13 00 0E 00 .....(.(.%.
00000070 09 00 0F 00 14 00 0B 00 0D 00 12 00 0B 00 04 00 .....
00000080 14 00 0F 00 1B 00 15 00 17 00 1A 00 16 00 1C 00 .....
```

Fig 16. Beginning of Data Frame

```
Sample.wav
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 52 49 46 46 64 51 05 00 57 41 56 45 66 6D 74 20 RIFFDQ..WAVEfmt
00000010 10 00 00 00 01 00 01 00 80 3E 00 00 00 7D 00 00 .....<E>...}...
00000020 02 00 10 00 01 00 01 00 64 61 74 61 40 51 05 00 EB FF E1 FF ...data@Q..eYáY
00000030 E1 FF DF FF E0 FF DE FF ED FF EC FF EF FF ED FF EC FF 4949494949494949
00000040 FB FF F0 FF F7 FF F3 03 00 00 04 00 11 00 08 00 0949-y.....
00000050 19 00 1D 00 18 00 1A 00 28 00 28 00 25 00 20 00 .....(.(.%.
00000060 20 00 1C 00 1D 00 13 00 1D 00 18 00 13 00 0E 00 .....(.(.%.
00000070 09 00 0F 00 14 00 0B 00 0D 00 12 00 0B 00 04 00 .....
00000080 14 00 0F 00 1B 00 15 00 17 00 1A 00 16 00 1C 00 .....
```

Fig 17. End of Data Frame

The beginning of the data frame can be seen in Figure 4.14, and the end of the data frame can be seen in Figure 4.16.

A. *Vigenere Cipher Encryption and Decryption Process*

The process of encrypting and decrypting wave audio files using the Vigenere Cipher with a character key is discussed in this sub-chapter. The flow of the encryption process is as follows:

- 1) Insert Wave Audio File.
- 2) Read the length of the input raw audio data.
- 3) Enter the character key.
- 4) Encrypt vigenere cipher 256.



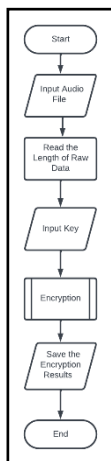


Fig 18. Vigenere Encryption Process

Decryption process flow:

- 1) Input Cipher Audio wave file.
- 2) Read the length of the input raw audio data.
- 3) Enter the key in the form of characters.
- 4) Decrypt Vigenere Cipher 256.

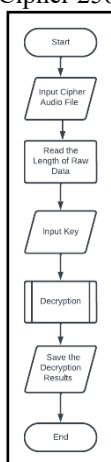


Fig 19. Vigenere Decryption Process

### B. Playfair Cipher Encryption and Decryption Process

This sub-chapter will discuss the process of encryption and decryption of the Modified Playfair Cipher 256 Random Seed. With encryption using the Playfair cipher 256 Random Seed, if there is input in the message in the form of the character "x," then during the encryption process, the character "x" is used to even out odd or unpaired letters and is used to separate the same character/alphabet pairs. Therefore, the author modifies the Playfair Cipher by replacing the "x" character with the ASCII extension  $x = 128$ . The encryption process flow uses the Modified Playfair Cipher 256 Random Seed as follows:

- 1) Enter the plain key.
- 2) Look for keys with random seeds using the frame rate parameter.
- 3) Enter the processed key from the random seed in the form of a number.
- 4) Perform Playfair Cipher 256 encryption. If some odd characters or characters are the same and in pairs, they will be paired and separated by 128 extended ASCII characters.

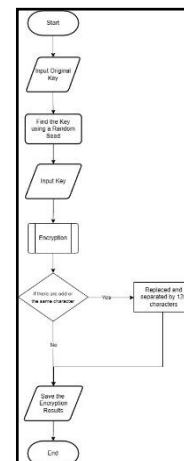


Fig 20. Playfair Encryption Process

Decryption process flow:

- 1) Enter the cipher key.
- 2) Search for keys with random seeds using the frame rate parameter.
- 3) Enter the processed key from the random seed in the form of a number.
- 4) Perform Playfair Cipher 256 decryption. If there is an extended ASCII 128 character, it will be removed.

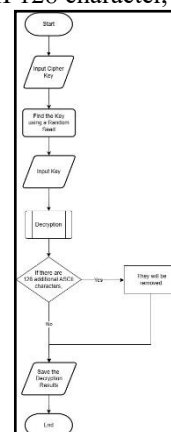


Fig 21. Playfair Decryption Process

### C. Testing and Analysis

In Testing and Analysis, MAE (Mean Absolute Error) is used as a measuring parameter to determine the accuracy of the performance in the decryption process. In addition, MAE determines how much data scrambling performance is when the encryption process is carried out. The MAE value represents the average absolute error between the forecasting results and the actual value. Mathematically MAE is defined as follows:

$$MAE = \frac{1}{n} \sum_{i=0}^{n-1} (|p_1 - y_1|) \quad (1)$$

Explanation:

$n$  = Number of data frames

$p_1$  = Audio Cipher Value

$y_1$  = Plain Audio Value

Peak Signal Noise Ratio compares the maximum intensity of the signal over time to the noise floor. This can be a valuable metric to approximate compression quality by comparing the source "signal" to the encoded target and reporting how much compression "noise" affected the signal in the output. PSNR is defined as follows:

$$PSNR = 20 \log_{10} \left( \frac{L-1}{RMSE} \right) \quad (2)$$

Explanation:

$L$  = Number of maximum possible intensity levels

MSE is the mean squared error & it is defined as:

$$MSE = \frac{1}{n} \sum_{i=0}^{n-1} (O - C)^2 \quad (3)$$

Explanation:

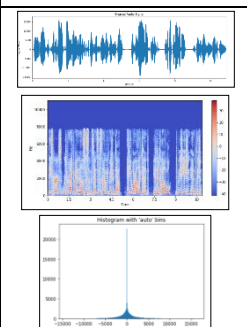
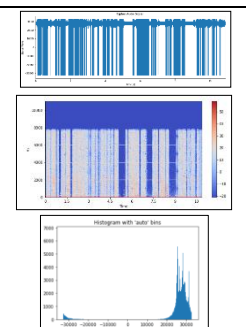
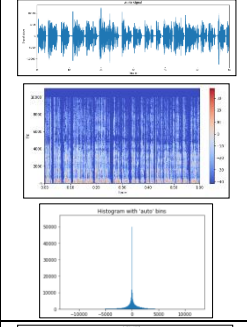
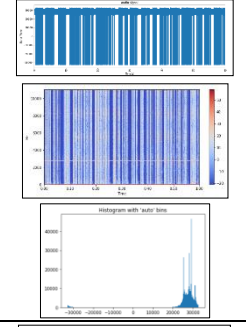
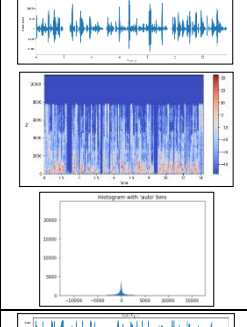
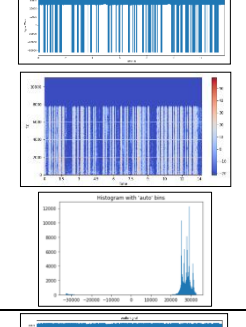
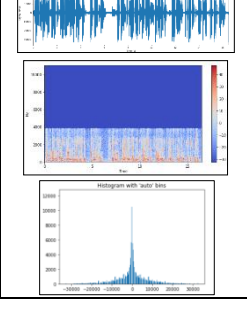
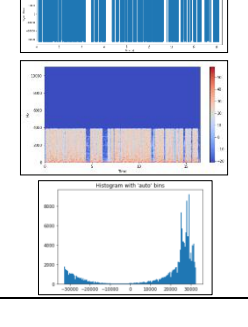
$n$  = Number of data frames

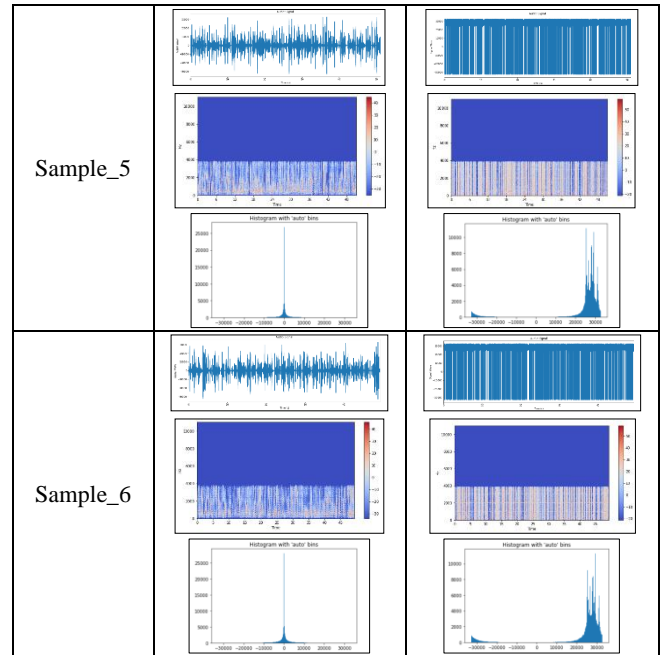
$C$  = Audio Cipher Value

$O$  = Plain Audio Value

Following are the results of the analysis of encryption and decryption testing of wav audio files using the Vignere cipher 256 using a spectrogram and histogram, which can be seen in Table 3.

TABLE 3  
VISUALIZATION TEST RESULTS IN GRAPHS

File Audio (*.wav)	Original Audio	Cipher Audio
Sample		
Sample_2		
Sample_3		
Sample_4		



A spectrogram is a visual representation of the spectrum of frequencies found in a signal as they vary with time. Spectrograms of audio frequencies are sometimes called voiceprints or voicegrams. A spectrogram is a visual representation of an audio file frequency spectrum varying with time. It is obtained with the Fourier transform. A spectrogram is most helpful for vibration analysis in a changing environment.

Histogram basically depicts an estimate of the probability distribution of some variable. To construct a histogram, the range of possible variable values gets divided into a series of intervals called bins. The bins must be adjacent to each other and are often (but necessarily) of equal width. Then a count of how many values fall into each interval determines the height of each bin such that the height is proportional to the number of cases in each bin. The histogram can be used to determine the search criteria. Histogram analysis is an accurate metric for measuring the quality of encrypted audio signals. A good encryption scheme has to encrypt the original audio file into a random-like noise.

In the dataset, there are 6 Audio that will be tested and analyzed for the encryption and decryption process can be seen in Table 3 and Table 4

TABLE 4  
DATASETS THAT ARE ANALYZED DURING THE ENCRYPTION PROCESS

File Audio (*.wav)	Size (KB)	Frame Length	Encrypt Time (s)	SPNR Analyst (dB)	MAE Analyst
Sample	340	174240	1.60	28.345	97.0625
Sample_2	2520	1323008	10.37	28.345	97.0625
Sample_3	429	219840	3.61	28.345	97.0625
Sample_4	259	132007	1.18	28.345	97.0625
Sample_5	798	408960	3.64	28.345	97.0625
Sample_6	770	394560	3.51	28.345	97.0625

Of the 6 datasets that have been tested 5 times, from Table 4.2, it can be seen that the results of the encryption process for Sample.wav with a duration of 10 seconds, a file size of 340 KB, a frame length of 174240, and from 5 trials, the average result of the encryption process takes 1.60 seconds. Sample\_2.wav, with a duration of 60 seconds, a file size of

2520 KB, a frame length of 1323008, and from 5 trials, the average result of the encryption process takes 10.37 seconds. Sample\_3.wav, with a duration of 13 seconds, a file size of 429 KB, a frame length of 219840, and from 5 trials, the average result of the encryption process takes 3.61 seconds. Sample\_4.wav, with a duration of 16 seconds, a file size of 259 KB, a frame length of 132007, and from 5 trials, the average result of the encryption process takes 1.18 seconds. Sample\_5.wav, with a duration of 51 seconds, a file size of 798 KB, a frame length of 408960, and from 5 trials, the average result of the encryption process takes 3.64 seconds. Sample\_6.wav, with a duration of 49 seconds, a file size of 770 KB, a frame length of 394560, and from 5 trials, the average result of the encryption process takes 3.51 seconds. Of the 6 datasets analyzed with PSNR and MAE, the level of change/randomness was obtained with an average of 28.345 and 97.0625.

TABLE 5  
DATASETS THAT ARE ANALYZED DURING THE DECRYPTION PROCESS

File Audio (*.wav)	Size (KB)	Frame Length	Decrypt Time (s)	SPNR Analyst (dB)	MAE Analyst
Sample	340	174240	1.54	0.0	0.0
Sample_2	2520	1323008	10.04	0.0	0.0
Sample_3	429	219840	1.97	0.0	0.0
Sample_4	259	132007	1.17	0.0	0.0
Sample_5	798	408960	3.60	0.0	0.0
Sample_6	770	394560	3.55	0.0	0.0

Of the 6 datasets that have been tested 5 times, from Table 4.3, it can be seen that the results of the decryption process for Sample.wav with a duration of 10 seconds, a file size of 340 KB, a frame length of 174240, and from 5 trials, the average result of the decryption process takes 1.54 seconds. Sample\_2.wav, with a duration of 60 seconds, a file size of 2520 KB, a frame length of 1323008, and from 5 trials, the average result of the decryption process takes 10.04 seconds. Sample\_3.wav, with a duration of 13 seconds, a file size of 429 KB, a frame length of 219840, and from 5 trials, the average result of the decryption process takes 1.97 seconds. Sample\_4.wav, with a duration of 16 seconds, a file size of 259 KB, a frame length of 132007, and from 5 trials, the average result of the decryption process takes 1.17 seconds. Sample\_5.wav, with a duration of 51 seconds, a file size of 798 KB, a frame length of 408960, and from 5 trials, the average result of the decryption process takes 3.60 seconds. Sample\_6.wav, with a duration of 49 seconds, a file size of 770 KB, a frame length of 394560, and from 5 trials, the average result of the decryption process takes 3.55 seconds. Of the 6 datasets analyzed with PSNR and MAE, the rate of return/change averaged 0.0 and 0.0, indicating successful decryption. So between the audio description and the original audio, there is no change or equal to 0.0.

## V. CONCLUSION

Based on the research that has been done, it can be concluded that:

- 1) Encryption of Vigenere Cipher 256 in audio wave files, the audio of the message being conveyed sounds unclear or meaningless
- 2) The audio files used with the wave or \*.wav extension can be audio with mono and stereo channels

- 3) The speed of the encryption and decryption process is affected by the audio file size, which means that the larger the file size, the longer the encryption and decryption time.
- 4) MAE and PSNR results show the performance of encryption and decryption results on audio waves. The average value of the encryption process on PSNR is 28.345, and MAE is 97.0625. The average value of the decryption process on PSNR and MAE is 0.0, indicating that the decryption process is successful.

## ACKNOWLEDGMENT

The author would like to thank the Faculty of Science and Technology, Informatics Engineering Study Program, Muhammadiyah University, East Kalimantan, for all the support, including facilities and funding.

## REFERENCES

- [1] Ramesh Shelke and Dr. Milind Nemade, *Audio Encryption Algorithm using Modified Elliptical Curve Cryptography and Arnold Transform for Audio Watermarking*. IEEE, 2018. Accessed: Mar. 13, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8284161>
- [2] A. Syahputra, "ANALISA IMPLEMENTASI PENGAMANAN FILEAUDIO MENGGUNAKAN ALGORITMA PLAYFAIR," 2018.
- [3] H. Santoso and M. Fakhriza, "PERANCANGAN APLIKASI KEAMANAN FILE AUDIO FORMAT WAV (WAVEFORM) MENGGUNAKAN ALGORITMA RSA," 2018.
- [4] V. B. Pawar, P. A. Tijare, S. N. Sawalkar, A. Professors, and A. Professor, "A Review Paper on Audio Encryption," *International Journal of Research in Advent Technology*, vol. 2, no. 12, 2014.
- [5] Z. N. Al-kateeb and S. J. Mohammed, "A novel approach for audio file encryption using hand geometry," *Multimed Tools Appl*, vol. 79, no. 27–28, pp. 19615–19628, Jul. 2020, doi: 10.1007/s11042-020-08869-8.
- [6] D. Rahmasari Kinasih Gusti, K. Agung Santoso, and A. Kamsyakawuni Jurusan Matematika, "VIGENERE CIPHER DENGAN MODIFIKASI PLAINTEXT (Vigenere Cipher Using Plaintext Modification)," 2020. [Online]. Available: <https://jurnal.unej.ac.id/index.php/MIMS/indexISSN1411-6669>
- [7] W. A. Lestari, R. Tulloh, A. Novianti, and S. St, "MEDIA PEMBELAJARAN INTERAKTIF ENKRIPSI CAESAR CIPHER, VIGENERE CIPHER, DAN ALGORITMA RSA Interactive Learning Media of Caesar Cipher, Vigenere Cipher, and RSA Algorithm Encryption," 2019.
- [8] Lindawati and Rita Siburian, *Steganography Implementation on Android Smartphone Using the LSB (Least Significant Bit) to MP3 and WAV Audio*. IEEE, 2017. Accessed: Mar. 13, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8284161>
- [9] A. Harsa, A. Yusika, and A. Ansharie, "ENKRIPSI DATA AUDIO MENGGUNAKAN METODE KRIPTOGRAFI RSA," 2017.
- [10] Juni Ade Nawer Purba, Debora Sinaga, and Saima Ronita Purba, *Implementasi Algoritma Paillier Cryptosystem Pengamanan Audio*. Seminar Nasional Teknologi Komputer & Sains (SAINTEKS), 2019. [Online]. Available: <https://seminar-id.com/seminas-sainteks2019.html>
- [11] R. I. Abdelfatah, "Audio Encryption Scheme Using Self-Adaptive Bit Scrambling and Two Multi Chaotic-Based Dynamic DNA Computations," *IEEE Access*, vol. 8, pp. 69894–69907, 2020, doi: 10.1109/ACCESS.2020.2987197.
- [12] Dr. Ekhlis Abbas Albahrani, *A New Audio Encryption Algorithm Based on Chaotic Block Cipher*. Iraq: IEEE Institute of Electrical and Electronics Engineers, 2017.
- [13] Chloé Albin, Dhruv Narayan, Ritika Varu, and Thanikaiselvan V, *DWT based Audio Encryption scheme*. 2018.
- [14] R. Harahap, "Implementasi Algoritma Skipjack Untuk Mengamankan Audio," vol. 2, no. 1, pp. 29–34, 2021, [Online]. Available: <https://ejournal.seminar-id.com/index.php/tin>
- [15] A. Susanto, "Image encryption using vigenere cipher with bit circular shift," 2021.
- [16] hdubey, "The Microsoft Scalable Noisy Speech Dataset (MS-SNSD)," [github.com](https://github.com/microsoft/MS-SNSD), Jan. 21, 2019. <https://github.com/microsoft/MS-SNSD> (accessed Jun. 16, 2023).



**SURAT KETERANGAN ARTIKEL PUBLIKASI**

*Assalamu'alaikum Warahmatullahi wabarakatuh*

Saya yang bertanda tangan dibawah ini:

Nama : M.Taufiq Sumadi, S.Tr.Kom., M.Tr.Kom  
NIDN : 1111089501  
Nama : Achmad Nur Zahir S  
NIM : 1911102441143  
Fakultas : Teknik Informatika  
Progam Studi : Sains dan Teknologi

Manyatakan bahwa artikel ilmiah yang berjudul "Secure Audio Files Using Vigenere Chipher And Playfair Cipher Methods" telah di submit pada Jurnal Informatika Jurnal Pengembangan IT pada tahun 2023.

<https://ejournal.poltektegal.ac.id/index.php/informatika/user>

Demikian surat keterangan ini dibuat untuk dapat dipergunakan sebagaimana mestinya.

*Wassalamu'alaikum Warahmatullahi wabarakatuh*

Samarinda, 24 Juli 2023

Mahasiswa

Dosen Pembimbing

**Achmad Nur Zahir S**  
NIM. 1911102441143

**M. Taufiq Sumadi, S.Tr.Kom., M.Tr.Kom**  
NIDN. 1111089501