Jurnal Teknik Informatika (JUTIF)
Vol. 5, No. 6, December 2024, pp. 1497-1504
DOI: https://doi.org/10.52436/1.jutif.2024.5.6.1624
p-ISSN: 2723-3863
e-ISSN: 2723-3871

SECURE AUDIO FILES USING VIGENERE CIPHER AND PLAYFAIR CIPHER

Muhammad Taufiq Sumadi[*1], Achmad Nur Zahir S[2], Faldi[3]

[1,2,3]Informatics Engineering, Faculty of Science and Technology, Universitas Muhammadiyah Kalimantan Timur, Indonesia
Email: [1]mts653@umkt.ac.id

(Article received: December 21, 2023; Revision: February 02, 2024; published: December 29, 2024)

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: *Audio, Confidentiality, Cryptography, Playfair Cipher, Vigenere Cipher.*

MENGAMANKAN FILE AUDIO MENGGUNAKAN VIGENERE CIPHER DAN 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: *Audio, Kerahasiaan, Kriptografi, Playfair Cipher, Vigenere Cipher.*

1. PENDAHULUAN

Pesatnya pertumbuhan teknologi digital, khususnya Internet dan teknologi komunikasi, telah memfasilitasi pertukaran data multimedia. Data multimedia digunakan dalam konferensi suara dan video waktu nyata, kontrol dan lalu lintas udara, pemantauan siaran, serta kontrol mesin dan perintah pengoperasian yang diaktifkan dengan suara. Namun, mengirimkan informasi rahasia atau sensitif melalui jaringan dan sistem komunikasi dapat berakibat fatal karena jaringan tersebut tidak aman dan rentan terhadap serangan virus. [1]

Audio, atau suara, adalah gelombang yang mengandung banyak komponen penting (amplitudo, panjang gelombang, dan frekuensi) yang dapat mengubah satu suara menjadi suara lainnya. Komputer dapat memahami suara analog melalui konversi analog-ke-digital (ADC). Proses ini dilakukan oleh perangkat keras komputer yang disebut kartu suara atau sound card. Format file audio yang umum digunakan adalah mp3, wav, dan rami.[2] Format gelombang (*.WAV) adalah format file suara yang banyak digunakan di sistem operasi Windows untuk keperluan game dan multimedia. Gelombang adalah bentuk mentah di mana suara direkam secara langsung dan dikuantisasi secara digital. Kemudahan pembuatan dan pemrosesan Format file dasar ini tidak mendukung kompresi secara default dan dikenal sebagai PCM (Pulse Code Modulation).[3]

Dengan pesatnya pertumbuhan teknologi komunikasi, melindungi audio dari peretas telah

menjadi isu penting bagi para insinyur. Oleh karena itu, untuk melindungi informasi berharga Anda, Anda memerlukan cara untuk mengenkripsi informasi dan data sebelum dikirimkan dengan aman.[1] Kriptografi adalah ilmu (atau keterampilan) untuk melindungi informasi sensitif dari penyusup dan peretas yang ingin menggunakannya untuk tujuan ilegal. Hal ini dapat didefinisikan sebagai memastikan kerahasiaan komunikasi antara kedua ujung koneksi. Ini terutama mencakup enkripsi dan dekripsi. Enkripsi berkaitan dengan konten terenkripsi dari pesan aman sehingga tidak dapat dibaca atau diuraikan oleh orang atau program yang tidak berwenang.[6]

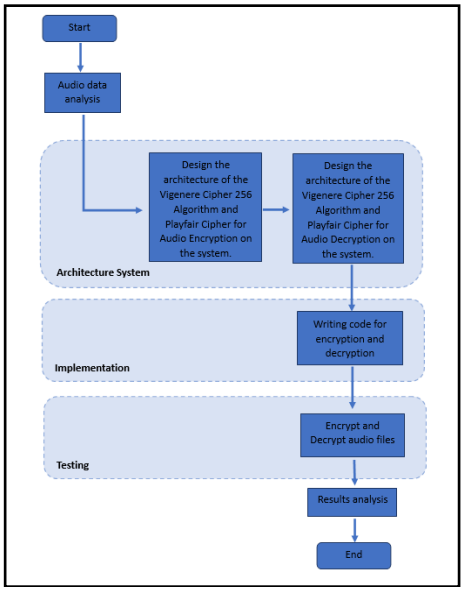
Enkripsi data audio adalah proses yang lebih sulit dan kompleks dibandingkan teknik yang digunakan untuk data teks. [4] Enkripsi audio adalah di mana sinyal audio atau suara ditransmisikan dengan aman dari satu titik ke titik lainnya. Sinyal audio dienkripsi agar benar-benar tidak terdeteksi oleh peretas atau kelompok pihak ketiga. Enkripsi audio adalah metode penggunaan kunci (yang bisa berupa noise) pada audio teks biasa dan sedangkan dekripsi adalah proses mengambil kembali teks biasa asli dengan menggunakan kunci yang sama. Berbagai teknik enkripsi digunakan untuk menyandikan sinyal audio untuk memastikan transmisi audio dengan aman hingga sampai ke tujuan. Namun tetap saja, ada trade-off antara keamanan dan kinerja berbagai algoritma. Ketika keamanan meningkat, kinerja menurun. Masalah ini akut pada file Audio.[5]

Vigenere Cipher merupakan kriptografi klasik yang menyembunyikan pesan dalam teks biasa dengan menggunakan teknik substitusi yang mengubah setiap karakter ke karakter lainnya berdasarkan kunci yang digunakan. Cipher Vigenere juga merupakan algoritma yang menggunakan kunci simetris dan mengulangi karakter dalam kunci tersebut hingga semua karakter dalam pesan telah diproses.[6] Playfair Cipher merupakan algoritma kriptografi klasik dengan cipher Polygram yang mengubah teks biasa menjadi bentuk Polygram dan melakukan proses enkripsi dan dekripsi Polygram. Kunci enkripsi terdiri dari 25 karakter yang disusun dalam kotak 5x5, tidak termasuk huruf J dari alfabet. Jadi probabilitas kuncinya adalah 25.

Pada penelitian ini penulis menggunakan metode Vigenere dan Playfair Cipher untuk merancang sistem yang dapat melindungi audio berformat Wave dan menjaga kerahasiaan file audio yang dikirim menggunakan 256 tabel ASCII.

2. METODE PENELITIAN

Pada penelitian ini memerlukan beberapa langkah antara lain Analisis data audio, penentuan arsitektur sistem, implementasi, pengujian sistem, dan analisis hasil. Gambar 1 menunjukkan diagram alir penelitian ini.



Gambar 1. diaram alir penelitian

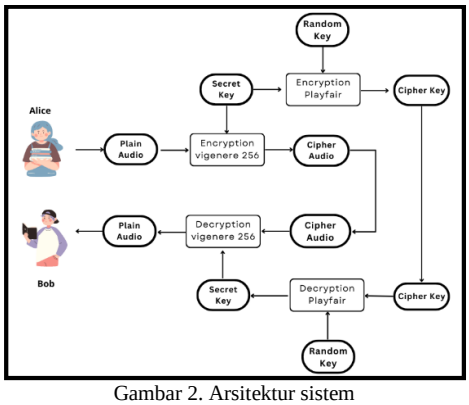
2.1. Analisis Data Audio

Dalam analisis data audio, pada tahap ini dilakukan pemahaman mengenai karakteristik, arsitektur, dan data audio untuk menghasilkan informasi yang akan memudahkan perancangan sistem dan meningkatkan keberhasilan pengamanan file audio.

Karena file Wave adalah substansi atau bagian dari file RIFF, maka ia mewarisi struktur file RIFF. Bagian data ini menggunakan metode Vigenere Cipher dan Playfair Cipher untuk melakukan enkripsi dan dekripsi.

2.2. Arsitektur Sistem

Penentuan arsitektur sistem merupakan salah satu cara untuk memudahkan pembangunan sistem yang dijelaskan pada Gambar 3.2. Gambar ini menjelaskan cara kerja sistem keamanan file audio menggunakan cipher Vigenere dan cipher Playfair yang dibuat.



Gambar 2. Arsitektur sistem

Pada Gambar 2 dijelaskan bahwa ketika Alice mencoba mengirimkan file audio ke Bob, audio tersebut dienkripsi menggunakan Vigenere Cipher dengan kunci yang ditentukan oleh Alice. Setelah itu,

Secret Key yang telah ditentukan Alice akan dienkripsi menggunakan Playfair Cipher dengan kunci acak dan menghasilkan cipher Key. Cipher Key dan Cipher Audio akan dikirimkan ke Bob secara bersamaan. Setelah mendapatkan cipher key dan audio cipher, Bob akan mendekripsi dengan metode Playfair Cipher terhadap Cipher Key terlebih dahulu menggunakan Random Key yang telah disetujui oleh Alice dan Bob dan akan menghasilkan Secret Key, setelah itu Bob akan mendekripsi audio dengan Vigenere Metode cipher dengan Secret Key dan akan menghasilkan audio yang polos. Kemudian Bob dapat mengetahui informasi yang diberikan oleh Alice kepadanya.

2.3. Implementasi

2.3.1. *Hardware*

Sebelum membangun sistem penelitian, kita perlu menyiapkan beberapa perangkat keras. Daftar perangkat yang digunakan dalam penelitian ini tertulis pada Tabel 1.

Tabel 1. *Hardware*

Nama	Spesifikasi
Komputer / 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*

2.3.2. *Software*

Sebelum membangun sistem, kami akan menggunakan beberapa perangkat lunak. Daftar perangkat lunak yang digunakan dalam penelitian ini tertulis pada Tabel 2.

Tabel 2. *Software*

Nama	Spesifikasi	Fungsi
Audio File	Ekstensi *.wav	Sebagai objek penelitian yang akan diterapkan pada sistem yang akan dirancang.
PyCharm Community Edition	*Version 2022.2.3*	Sebagai editor, menuliskan bahasa pemrograman Python yang akan diterapkan pada sistem yang akan dirancang.
Python	*Python 3*	Sebagai bahasa pemrograman yang akan diterapkan pada sistem penelitian
HxD Hex Editor	*2.5.0.0 (x86-64)*	Sebagai aplikasi untuk membaca nilai heksadesimal audio

2.4. Pengujian

File audio dienkripsi dan didekripsi selama tahap pengujian menggunakan sistem yang dibuat sebelumnya dengan metode Vigenere Cipher dan Playfair Cipher yang ditetapkan.

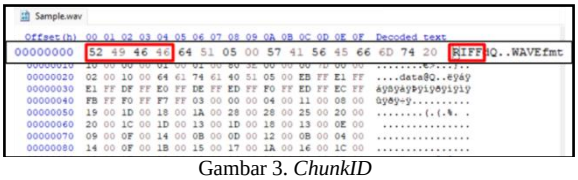
2.5. Hasil analisis

Pada fase ini, beberapa sampel audio terenkripsi dan terdekripsi dianalisis, seperti waktu, ukuran, tingkat keberhasilan, dan kesalahan file audio terenkripsi/dekripsi.

3. HASIL DAN PEMBAHASAN

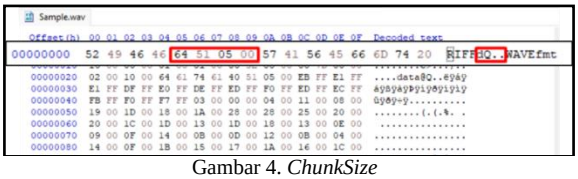
Pada bagian ini membahas tentang algoritma Vigenere Cipher 256 dan Playfair Cipher dalam mengenkripsi dan mendekripsi file gelombang audio. MAE (Mean Absolute Error) digunakan sebagai parameter pengukur untuk mengetahui performa dan akurasi yang dihasilkan dari proses dekripsi serta seberapa besar pengacakan data pada saat proses enkripsi dilakukan. PSNR (Peak Signal Noise Ratio) digunakan untuk membandingkan intensitas sinyal maksimum dari waktu ke waktu dengan basis noise untuk menentukan seberapa besar perubahannya selama proses enkripsi dan dekripsi. MSE (Mean Squared Error) mengukur rata-rata kuadrat kesalahan—yaitu, selisih kuadrat rata-rata antara nilai perkiraan dan nilai sebenarnya. Analisis histogram adalah metrik akurat untuk mengukur kualitas sinyal audio terenkripsi. Skema enkripsi yang baik harus mengenkripsi file audio asli menjadi suara acak. Spektogram adalah representasi visual dari spektrum frekuensi yang ditemukan dalam suatu sinyal yang bervariasi terhadap waktu. Spektogram sangat membantu untuk analisis getaran di lingkungan yang berubah.

File audio Sample.wav yang diperoleh pada dataset akan digunakan sebagai objek penelitian atau audio dasar yang akan digunakan untuk enkripsi dan dekripsi. Untuk membaca nilai heksadesimal audio, Anda akan menggunakan aplikasi HxD.



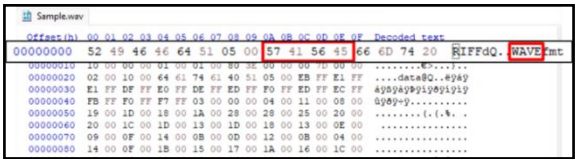
Gambar 3. *ChunkID*

Pada Gambar 3 terlihat audio berisi heksadesimal biner yang memiliki struktur dimulai dengan header RIFF yang terdiri dari *ChunkID* mulai dari *byte* 0 – 4, termasuk nilai 52 49 46 46 dengan teks yang diterjemahkan "RIFF".



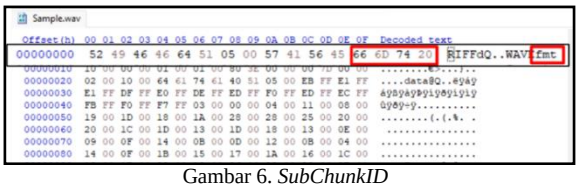
Gambar 4. *ChunkSize*

Pada Gambar 4, *ChunkSize* dimulai dari *byte* 4 - 8 berisi 64 51 05 00, Biasanya 8 *byte* (*integer 32-bit*) dari total ukuran file - dalam *byte*.



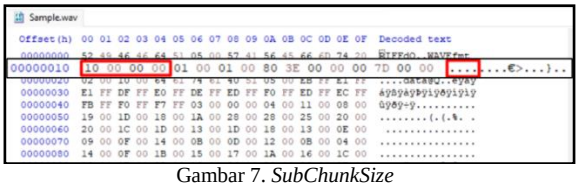
Gambar 5, *Format*

Dan pada Gambar 5, bagian *Format* dimulai dari *byte* 8-12, termasuk nilai 57 41 56 45. Representasi heksadesimal dari ASCII berupa huruf “WAVE”. Oleh karena itu, 57 41 56 45 untuk setiap file WAV.



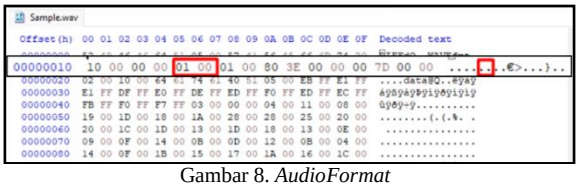
Gambar 6. *SubChunkID*

Pada Gambar 6 setelah RIFF merupakan bagian “fmt” yang terdiri dari *SubchunkID* yang dimulai dari byte 12 – 16 yang berisi nilai 66 6D 74 20. Representasi heksadesimal dari ASCII berupa huruf “fmt ” (perhatikan terdapat spasi kosong).



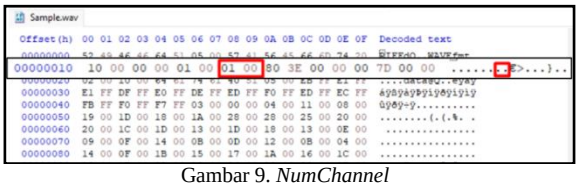
Gambar 7. *SubChunkSize*

Pada Gambar 7, *SubchunkSize* dimulai dari byte 16 – 20 yang berisi nilai 10 00 00 00, ukuran *subchunk* 16, yaitu penjumlahan sisa ukuran subchunk dari *format audio* ke *BitsPerSample* (2+2+4+4+2+2=16).



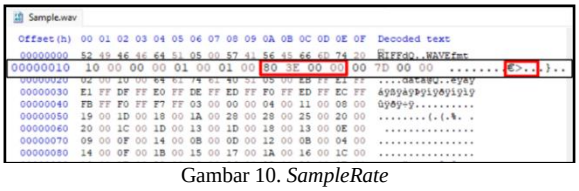
Gambar 8. *AudioFormat*

Pada Gambar 8, *AudioFormat* dimulai dari *byte* 20 -22 yang berisi nilai 01 00. 1 untuk PCM, nilai lainnya mewakili bentuk kompresi lainnya. Sampah endian, jadi heksadesimal 0001 adalah 1 dalam desimal.



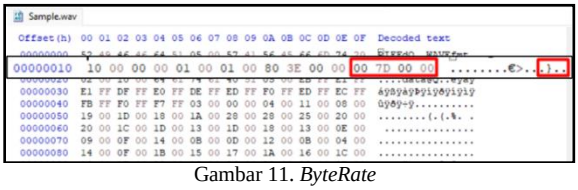
Gambar 9. *NumChannel*

Pada Gambar 9, *NumChannel* dimulai dari *byte* 22 – 24 yang berisi nilai 01 00. Litter endian, jadi heksadesimal 00 01 adalah 1 untuk desimal, 1 untuk mono, 2 untuk stereo.



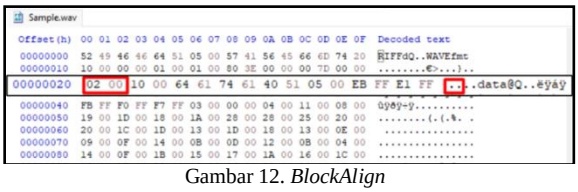
Gambar 10. *SampleRate*

Pada Gambar 10, *SampleRate* dimulai dari byte 24 – 28 yang berisi nilai 80 3E 00 00. *Sample Rate* = Sample per second atau Hertz. FYI, 80 3E 00 00 adalah singkatan dari 16000 Hz (little endian 00003E80). Nilai umumnya adalah 44100 (CD), 48000 (DAT).



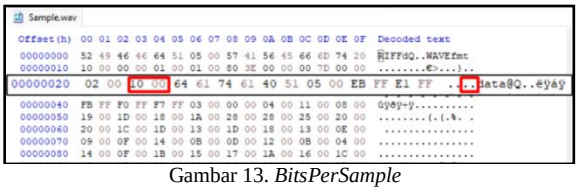
Gambar 11. *ByteRate*

Pada Gambar 11, *ByteRate* dimulai dari *byte* 28 – 32 yang berisi nilai 00 7D 00 00. *ByteRate* didapat dari (*Sample Rate * Bits Per Sample * Channel Numbers*) / 8. Jadi *ByteRate* = 32000.



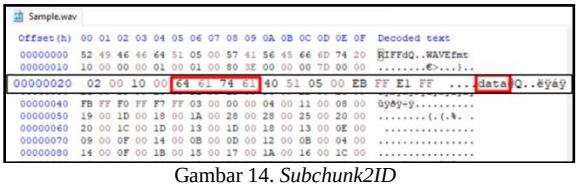
Gambar 12. *BlockAlign*

Pada Gambar 12, *BlockAlign* dimulai dari byte 32 – 34 yang berisi nilai 02 00 (*little Indian* 0002). Biasanya berisi dari (*NumChannels * BitsPerSample*) / 8. Jadi heksadesimal 00 02 adalah 2 dalam desimal.



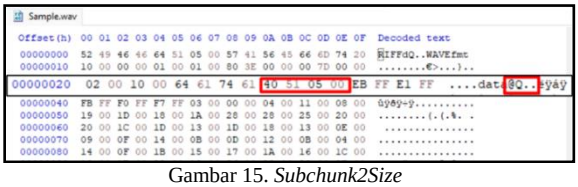
Gambar 13. *BitsPerSample*

Dan pada gambar 13, *BitsPerSample* dimulai dari *byte* 34 – 36 berisi 10 00. Biasanya berisi 16 bit = 1000 (*little endian* 0010); 32 bit = 2000 *(little endian* 0020); dst. atau 8 bit = 8, 16 bit = 16, dst.



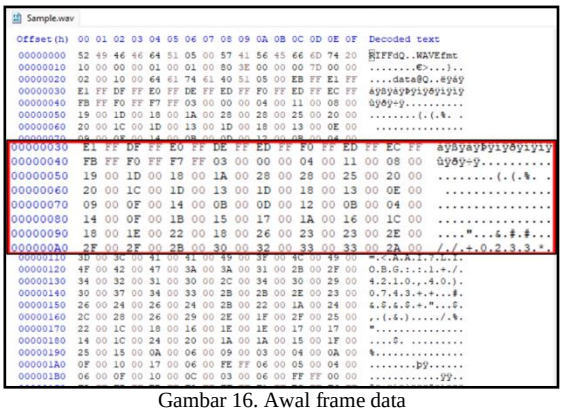
Gambar 14. *Subchunk2ID*

Pada Gambar 14, Setelah " fmt" bagian terakhir dari struktur *wave* adalah Data yang terdiri dari *Subchunk2ID* yang dimulai dengan *byte* 36 – 40 yang berisi nilai 64 62 74 61. Representasi heksadesimal dari ASCII berupa huruf "DATA".

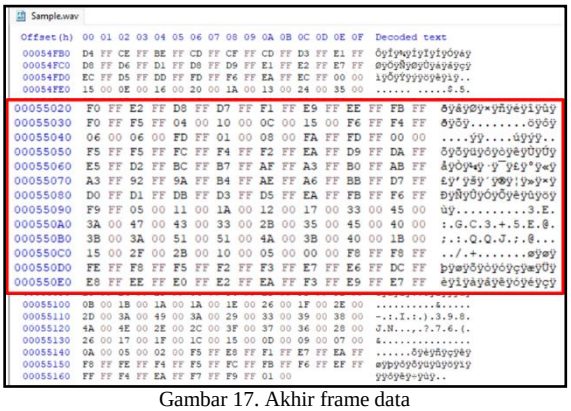


Gambar 15. *Subchunk2Size*

Pada Gambar 15, *Subchunk2Size* dimulai dari byte 40 – 44 yang berisi nilai 40 51 05 00. Biasanya berisi Jumlah *byte* pada data yang diperoleh dari (*NumSamples* * *NumChannels* * *BitsPerSample*) / 8.



Gambar 16. Awal frame data



Gambar 17. Akhir frame data

Bagian awal frame data dapat dilihat pada Gambar 4.14, dan bagian akhir frame data dapat dilihat pada Gambar 4.16.

3.1. Proses enkripsi dan dekripsi vigenere cipher

Proses enkripsi dan dekripsi file audio gelombang menggunakan Vigenere Cipher dengan kunci karakter dibahas pada sub bab ini. Alur proses enkripsi adalah sebagai berikut:

1) Masukkan File Audio Gelombang.
2) Membaca panjang input data audio mentah.
3) Masukkan kunci karakter.
4) Enkripsi sandi vigenere 256

Alur proses dekripsi:

1) Masukkan file gelombang Cipher Audio.
2) Membaca panjang input data audio mentah.
3) Masukkan kunci dalam bentuk karakter.
4) Dekripsi Vigenere Cipher 256.

3.2. Proses enkripsi dan dekripsi playfair cipher

Sub bab ini akan membahas tentang proses enkripsi dan dekripsi Modified Playfair Cipher 256 Random Seed. Dengan enkripsi menggunakan Playfair cipher 256 Random Seed, jika pada pesan terdapat masukan berupa karakter "x", maka pada saat proses enkripsi, karakter "x" digunakan untuk meratakan huruf ganjil atau tidak berpasangan dan digunakan untuk memisahkan pasangan karakter/abjad yang sama. Oleh karena itu, penulis memodifikasi Playfair Cipher dengan mengganti karakter "x" dengan ekstensi ASCII x = 128. Alur proses enkripsi menggunakan Modified Playfair Cipher 256 Random Seed sebagai berikut:

1) Masukkan kunci biasa.
2) Cari kunci dengan seed acak menggunakan parameter frame rate.
3) Masukkan kunci yang diproses dari benih acak berupa angka.
4) Lakukan enkripsi Playfair Cipher 256. Jika beberapa karakter ganjil atau karakter sama dan berpasangan, maka akan dipasangkan dan dipisahkan.

Alur proses dekripsi:

1) Masukkan kunci sandi.
2) Cari kunci dengan seed acak menggunakan parameter frame rate.
3) Masukkan kunci yang diproses dari benih acak berupa angka.
4) Lakukan dekripsi Playfair Cipher 256.

3.3. Pengujian dan analisis

Dalam Pengujian dan Analisis, MAE (Mean Absolute Error) digunakan sebagai parameter pengukur untuk mengetahui keakuratan kinerja dalam proses dekripsi. Selain itu, MAE menentukan seberapa besar kinerja pengacakan data ketika proses enkripsi dilakukan. Nilai MAE mewakili rata-rata kesalahan absolut antara hasil peramalan dengan nilai sebenarnya. Secara matematis MAE didefinisikan sebagai berikut:

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

Penjelasan:
n = Jumlah frame data
p_1 = Audio Cipher Value
y_1 = Plain Audio Value

Peak Signal Noise Ratio (PSNR) membandingkan intensitas maksimum sinyal dari waktu ke waktu dengan tingkat kebisingan. Ini bisa menjadi metrik yang berharga untuk memperkirakan kualitas kompresi dengan membandingkan "sinyal" sumber dengan target yang dikodekan dan melaporkan seberapa banyak "noise" kompresi yang memengaruhi sinyal dalam keluaran. PSNR didefinisikan sebagai berikut:

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

Penjelasan:
L = Jumlah tingkat intensitas maksimum yang mungkin

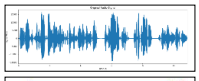
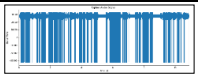
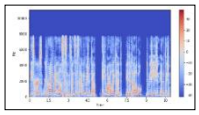
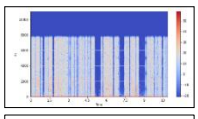
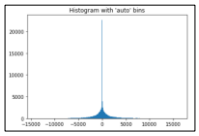
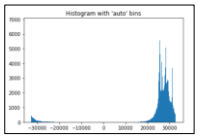
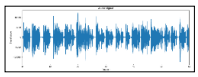
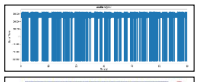
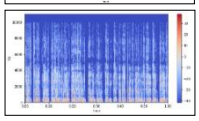
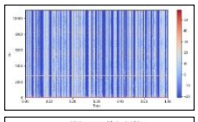
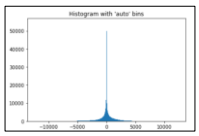
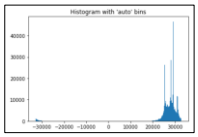
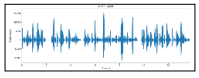
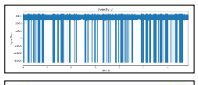
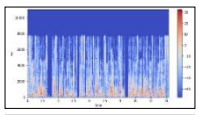
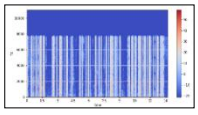
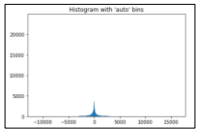
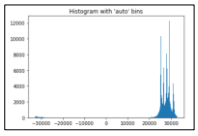
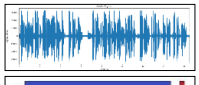
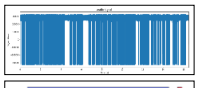
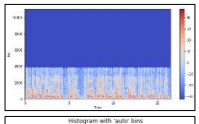
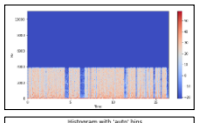
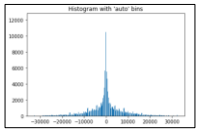
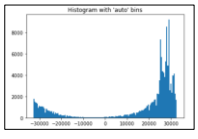
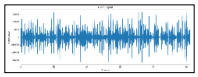
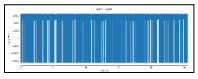
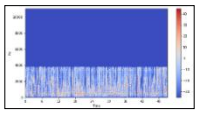
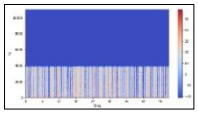
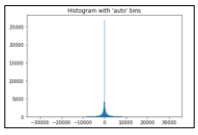
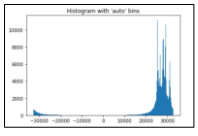
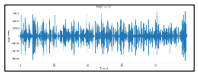
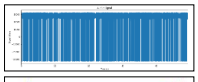
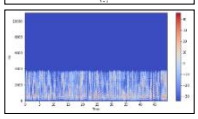
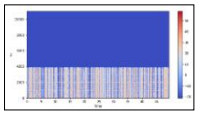
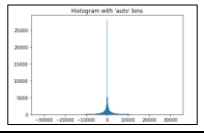
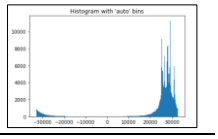
MSE adalah kesalahan kuadrat rata-rata & didefinisikan sebagai:

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

Explanation:
n = Jumlah frame data
C = Audio Cipher Value
O = Plain Audio Value

Berikut hasil analisis pengujian enkripsi dan dekripsi file audio wav menggunakan Vigenere cipher 256 menggunakan spektogram dan histogram, dapat dilihat pada Tabel 3.

Tabel 3 Hasil uji visualisasi dalam grafik

File Audio (*.wav)	Original Audio	Cipher Audio
Sample		
		
		
Sample_2		
		
		
Sample_3		
		
		
Sample_4		
		
		
Sample_5		
		
		
Sample_6		
		
		

Spektogram adalah representasi visual dari spektrum frekuensi yang ditemukan dalam suatu sinyal yang bervariasi terhadap waktu. Spektogram frekuensi audio terkadang disebut voiceprints atau voicegrams. Spektogram adalah representasi visual dari spektrum frekuensi file audio yang bervariasi terhadap waktu. Itu diperoleh dengan transformasi Fourier. Spektogram sangat berguna untuk analisis getaran dalam lingkungan yang berubah.

Histogram pada dasarnya menggambarkan perkiraan distribusi probabilitas suatu variabel. Untuk membuat histogram, rentang nilai variabel yang mungkin dibagi menjadi serangkaian interval yang disebut bin. Tempat sampah harus berdekatan satu sama lain dan sering kali (tetapi harus) memiliki lebar yang sama. Kemudian penghitungan berapa banyak nilai yang masuk ke dalam setiap interval menentukan tinggi setiap nampan sedemikian rupa sehingga tingginya sebanding dengan jumlah kasus di setiap nampan. Histogram dapat digunakan untuk menentukan kriteria pencarian. Analisis histogram adalah metrik akurat untuk mengukur kualitas sinyal audio terenkripsi. Skema enkripsi yang baik harus mengenkripsi file audio asli menjadi suara acak.

Pada dataset terdapat 6 Audio yang akan diuji dan dianalisis untuk proses enkripsi dan dekripsi dapat dilihat pada Tabel 3 dan Tabel 4.

Tabel 4. Dataset yang dianalisis selama proses enkripsi

File Audio (*.wav)	Size (KB)	Frame Lenght	Encrypt Time (s)	PSNR 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

Dari 6 dataset yang telah diuji sebanyak 5 kali, dari Tabel 4 terlihat hasil proses enkripsi Sample.wav

dengan durasi 10 detik, ukuran file 340 KB, panjang frame 174240, dan dari 5 kali percobaan, rata-rata hasil proses enkripsi membutuhkan waktu 1,60 detik. Sample_2.wav, dengan durasi 60 detik, ukuran file 2520 KB, panjang frame 1323008, dan dari 5 kali percobaan, rata-rata hasil proses enkripsi membutuhkan waktu 10,37 detik. Sample_3.wav, dengan durasi 13 detik, ukuran file 429 KB, panjang frame 219840, dan dari 5 kali percobaan, rata-rata hasil proses enkripsi membutuhkan waktu 3,61 detik. Sample_4.wav, dengan durasi 16 detik, ukuran file 259 KB, panjang frame 132007, dan dari 5 kali percobaan, rata-rata hasil proses enkripsi membutuhkan waktu 1,18 detik. Sample_5.wav, dengan durasi 51 detik, ukuran file 798 KB, panjang frame 408960, dan dari 5 kali percobaan, rata-rata hasil proses enkripsi membutuhkan waktu 3,64 detik. Sample_6.wav, dengan durasi 49 detik, ukuran file 770 KB, panjang frame 394560, dan dari 5 kali percobaan, rata-rata hasil proses enkripsi membutuhkan waktu 3,51 detik. Dari 6 dataset yang dianalisis dengan PSNR dan MAE diperoleh tingkat perubahan/keacakan dengan rata-rata 28,345 dan 97,0625.

Tabel 4. Dataset yang dianalisis selama proses enkripsi

File Audio (*.wav)	Size (KB)	Frame Lenght	Encrypt Time (s)	PSNR 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

Dari 6 dataset yang telah diuji sebanyak 5 kali, dari Tabel 4 terlihat hasil proses dekripsi Sample.wav dengan durasi 10 detik, ukuran file 340 KB, panjang frame 174240, dan dari 5 kali percobaan, rata-rata hasil proses dekripsi membutuhkan waktu 1,54 detik. Sample_2.wav, dengan durasi 60 detik, ukuran file 2520 KB, panjang frame 1323008, dan dari 5 kali percobaan, rata-rata hasil proses dekripsi membutuhkan waktu 10,04 detik. Sample_3.wav, dengan durasi 13 detik, ukuran file 429 KB, panjang frame 219840, dan dari 5 kali percobaan, rata-rata hasil proses dekripsi membutuhkan waktu 1,97 detik. Sample_4.wav, dengan durasi 16 detik, ukuran file 259 KB, panjang frame 132007, dan dari 5 kali percobaan, rata-rata hasil proses dekripsi membutuhkan waktu 1,17 detik. Sample_5.wav, dengan durasi 51 detik, ukuran file 798 KB, panjang frame 408960, dan dari 5 kali percobaan, rata-rata hasil proses dekripsi membutuhkan waktu 3,60 detik. Sample_6.wav, dengan durasi 49 detik, ukuran file 770 KB, panjang frame 394560, dan dari 5 kali percobaan, rata-rata hasil proses dekripsi membutuhkan waktu 3,55 detik. Dari 6 kumpulan data yang dianalisis dengan PSNR dan MAE, tingkat pengembalian/perubahan rata-rata 0,0 dan 0,0, yang menunjukkan keberhasilan dekripsi. Jadi antara deskripsi audio dengan audio aslinya tidak ada perubahan atau sama dengan 0,0.

4. DISKUSI

Pada penelitian-penelitian sebelumnya, salah satu metode yang digunakan adalah RSA (Rivest Shamir Adleman). Akibatnya enkripsi tidak dapat berfungsi pada format audio yang terdaftar di aplikasi. Semakin besar ukuran file audio maka perkiraan waktu proses enkripsi juga semakin lama karena lamanya proses perhitungan, dan penerapan enkripsi file audio RSA hanya menghasilkan file berbeda yang tidak dapat digunakan sama sekali.

Namun penelitian yang dibahas kali ini akan menggunakan dua metode yaitu metode Vigenere Cipher dan metode Playfair Cipher. Vigenere Cipher digunakan untuk mengenkripsi file audio karena Vigenere Cipher merupakan kriptografi simetris yang artinya proses enkripsi dan dekripsinya menggunakan kunci yang sama. Oleh karena itu, peran Playfair Cipher adalah mengenkripsi Kunci Rahasia agar keamanan tetap terjaga. Vigenere Cipher dan Playfair Cipher adalah kriptografi klasik. Oleh karena itu metode-metode tersebut digunakan dengan harapan akan menghasilkan perhitungan dan proses komputasi yang lebih cepat. Pada file audio, bagian datanyalah yang akan dienkripsi, sehingga file audio tersebut tetap dapat berjalan sehingga informasi yang ada di dalamnya pun terenkripsi.

5. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, maka dapat disimpulkan bahwa, Enkripsi Vigenere Cipher 256 pada file gelombang audio, audio pesan yang disampaikan terdengar tidak jelas atau tidak berarti. File audio yang digunakan dengan ekstensi wave atau *wav dapat berupa audio dengan saluran mono dan stereo. Kecepatan proses enkripsi dan dekripsi dipengaruhi oleh ukuran file audio, artinya semakin besar ukuran file maka waktu enkripsi dan dekripsi akan semakin lama. Hasil MAE dan PSNR menunjukkan kinerja hasil enkripsi dan dekripsi pada gelombang audio. Nilai rata-rata proses enkripsi pada PSNR sebesar 28.345 dan MAE sebesar 97.0625. Nilai rata-rata proses dekripsi pada PSNR dan MAE adalah 0,0 yang menunjukkan bahwa proses dekripsi berhasil.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Fakultas Sains dan Teknologi, Program Studi Teknik Informatika, Universitas Muhammadiyah Kalimantan Timur, atas segala dukungannya, termasuk fasilitas dan pendanaan.

DAFTAR PUSTAKA

[1] R. Shelke and Dr. M. 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] Dr. E. A. Albahrani, " A New Audio Encryption Algorithm Based on Chaotic Block Cipher ", 2017th ed. NTICT, 2017.

[5] H. Bijlani, D. Gupta, and M. Lovanshi, "Audio Encryption Optimization," 2018, doi: 10.1109/CSNT.2018.38.

[6] 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.

[7] 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

[8] 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.

[9] Lindawati and R. 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

[8] A. Harsa, A. Yusika, and A. Ansharie, "Enkripsi Data Audio Menggunakan Metode Kriptografi RSA" 2017.

[9] J. A. N. Purba, D. Sinaga, and S. R. Purba, "Implementasi Algoritma Paillier Cryptosystem Pengamanan Audio," *Seminar Nasional Teknologi Komputer & Sains (SAINTEKS)*, 2019. [Online]. Available: https://seminar-id.com/semnas-sainteks2019.html

[10] 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.

[11] Dr. E. A. Albahrani, "A New Audio Encryption Algorithm Based on Chaotic Block Cipher," *Iraq: IEEE Institute of Electrical and Electronics Engineers*, 2017.

[13] C. Albin, D. Narayan, R. Varu, and V. Thanikaiselvan, "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://ejurnal.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, Jan. 21, 2019. https://github.com/microsoft/MS-SNSD (accessed Jun. 16, 2023)..