Pemanfaatan Steganografi untuk Menyampaikan Pesan Rahasia Berupa Musik Midi

Martha Monica (13510080)
Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganeca 10 Bandung 40132, Indonesia
martha.monica@students.itb.ac.id

***Abstrak*— Steganografi merupakan salah satu algoritma yang digunakan untuk menyembunyikan informasi rahasia. Makalah ini akan membahas mengenai salah satu algoritma baru yang dapat digunakan untuk menyembunyikan pesan rahasia dengan memanfaatkan penggunaan musik midi.**

***Kata kunci*—steganografi, midi, kriptografi, pesan rahasia.**

I. PENDAHULUAN

Kriptografi merupakan ilmu yang mempelajari teknik-teknik, baik secara matematis ataupun intuitif, yang berhubungan dengan aspek keamanan informasi seperti bagaimana menjaga kerahasiaan dari suatu data dan informasi, bagaimana menjaga keabsahan dari suatu data, bagaimana agar integritas suatu data atau informasi dapat tetap dipertahankan, serta berbagai macam cara untuk autentikasi data / informasi.

Saat ini, penggunaan ilmu kriptografi berkembang sangat pesat. Tidak sedikit orang yang tertarik untuk mengembangkan berbagai macam cara dalam ilmu kriptografi ini. Semua ini berkaitan dengan salah satu tujuan mendasar dari ilmu kriptografi yaitu bagaimana kita dapat menjaga kerahasiaan dari suatu informasi.

Perkembangan ilmu kriptografi bisa dibilang cukup stabil dan pesat. Semenjak ilmu ini pertama kali ditemukan, banyak sekali orang yang ikut serta memberikan kontribusi dalam menciptakan berbagai teori dan algoritma baru dalam menjaga kerahasiaan informasi tertentu. Namun, seiring itu pula, berbagai algoritma itu kemudian diujicoba dan sebagian besar berhasil dipecahkan.

Salah satu pengembangan dari ilmu kriptografi ini adalah steganografi yang merupakan ilmu yang berkaitan dengan bagaimana cara untuk menyembunyikan informasi tertentu dengan menyisipkan pesan rahasia di dalam pesan/ media lain. Terdapat berbagai macam algoritma ditemukan untuk mengimplementasikan bagaimana penggunaan steganografi dalam penyembunyian pesan baik dalam bentuk teks maupun media digital lainnya seperti foto, musik dan video.

II. MIDI

MIDI (Musikal Instrument Digital Interface) merupakan salah satu standar teknis yang menjelaskan sebuah protocol, antar muka digital, dan konektor yang menghubungkan beberapa instrumen musik elektronik, komputer, dan berbagai piranti sejenis. Sebuah file musik berekstensi MIDI berisi pesan yang mencakup notasi, pitch, dan kecepatan, sinyal – sinyal pengendali yang berisi parameter seperti volume, vibrato, audio panning dan cues, serta sinyal clock yang digunakan untuk mengatur dan mensinkronisasikan tempo dari beberapa device berbeda.

Byte pada musik bertipe MIDI berkisar dari 0 hingga 255 sehingga apabila direpresentasikan dalam bentuk biner, akan kita peroleh jangkauan nilai yang mungkin adalah 00000000 hingga 11111111. Sedangkan apabila kita menggunakan basis bilangan hexadecimal, akan kita peroleh jangkauan musik midi berkisar antar nilai 00 hingga FF.

III. ALGORITMA

Pada makalah ini, dijelaskan salah satu algoritma baru yang dapat digunakan untuk menyembunyikan pesan rahasia dengan memanfaatkan penggunaan algoritma steganografi. Mungkin selama ini, cara yang digunakan untuk menyembunyikan pesan rahasia tertentu, digunakan cara dengan menyisipkan pesan tersebut ke dalam berkas lainnya. Pada makalah ini, algoritma yang diajukan adalah menyembunyikan informasi rahasia yang bisa berupa file teks atau file digital lainnya menjadi sebuah file bertipe MIDI.

3.1 Enkripsi file

Pada bagian ini akan dijelaskan bagaimana proses enkripsi yang dilakukan untuk menyembunyikan suatu pesan / data rahasia menjadi suatu file bertipe MIDI.

3.1.1 Konversi file input menjadi binary file

Pada algoritma ini setiap karakter file input akan dikonversi menjadi representasi binernya di mana untuk satu karakter akan direpresentasikan dalam 8 digit bilangan biner. Implementasi dari konversi file input menjadi binary file dijelaskan melalui pseudocode sebagai berikut

```
function toByte(String a) → byte[]
{Fungsi ini mengembalikan representasi byte dari suatu string}

DEKLARASI
    i : integer
    j: integer
    tmp : byte[]

ALGORITMA
    for i=1 to a.length() do
        tmp = getByte(a[i])

        for j=0 to (7-tmp.Length()) do
                toByte[i*8+j] = 0
        endfor
        //bagian ini memastikan representasi biner untuk
        setiap karakter panjang yang sama yaitu 8 digit

        for j=0 to (tmp.Length()) do
                toByte[i*8+tmp.Length()+j] = tmp[j]
        endfor

    endfor
```

3.1.2 Konversi dari binary file ke dalam representasi hexadecimal

Setelah file input dikonversi menjadi file dalam bentuk biner, file-file ini kemudian masuk ke dalam konversi ke dalam bentuk hexadecimal. Hal ini memiliki tujuan agar file-file biner tadi dapat dibuat dalam representasi musik di mana dalam representasi dalam bidang musik, jangkauan tangga nada direpresentasikan dalam nada dasar A-G, sedangkan pada representasi bilangan basis hexadecimal, representasi yang mungkin hanyalah 0-9 dan huruf A-F sehingga penggunaan basis hexadecimal ini masih dapat dikorelasikan dengan penggunaan tangga nada. Algoritma konversi dari binary file ke dalam hexadecimal dapat dijelaskan melalui pseudocode sebagai berikut

```
function fourByteToHexa(byte [] a) → byte
{Fungsi ini mengembalikan satu digit representasi hexadecimal dari empat digit representasi binernya}

DEKLARASI
    i : integer
    c : byte
ALGORITMA
    c = 0
    for i=1 to 4 do
        c = c * 2
        c = c + a[i]
    endfor
    //setelah bagian ini, akan kita peroleh hasil konversi 4 digit bilangan biner tersebut menjadi representasi hexadecimal dalam bentuk angka.

    if ( c>9) then
    // merubah representasi tersebut dalam bentuk angka
        c = c + 55
        // hal tersebut bertujuan agar nilai c = 10 akan berubah menjadi A yang bernilai 65 dalam ASCII
    else
        c = c + 48
        // hal tersebut bertujuan untuk merubah nilai 0 – 9 menjadi karakter 0 – 9
    endif
```

```
function byteToHexa(byte [] a) → byte
{Fungsi ini mengembalikan representasi hexadecimal dari representasi biner}

DEKLARASI
    i : integer
    j : integer
    tmp : byte[4]

ALGORITMA
    for i=1 to (a.length() div 4) do
        tmp[0] = a[i*4 + 0]
        tmp[1] = a[i*4 + 1]
        tmp[2] = a[i*4 + 2]
        tmp[3] = a[i*4 + 3]
        byteToHexa[i] = fourByteToHexa(tmp)
    endfor
```

3.1.2 Konversi file hexadecimal menjadi file MIDI dan file teks eksternal

Setelah didapatkan representasi hexadecimal dari suatu data, maka yang dilakukan adalah memainkannya dalam bentuk midi ataupun kita bisa menyimpan representasi dalam bentuk tangga-tangga nada dasar yang siap kita ubah menjadi partitur. Pada algoritma ini, representasi musik tersebut dibuat dengan aturan sebagai berikut. Karakter A-F melambangkan nada yang akan dimainkan sedangkan karakter angka melambangkan berapa kali nada tersebut

dibunyikan secara berurutan. Apabila ditemukan dua buah karakter alfabet yang muncul berurutan maka, karakter alfabet pertama dibunyikan sekali. Apabila sebuah karakter alfabet diikuti dengan karakter angka n, maka kunci tangga nada yang mewakili karakter alfabet tersebut akan dibunyikan sebanyak n kali. Apabila sebuah karakter angka m ditemukan berurutan dengan karakter angka n, maka kunci tangga nada yang mewakili karakter alfabet sebelumnya akan dibunyikan sebanyak m kali, diikuti oleh tanda berhenti selama satu ketukan, kemudian kunci tangga nada sebelumnya dibunyikan lagi sebanyak n kali. Pada saat memainkan MIDI, bunyi dari tangga nada tersebut diwakili oleh karakter alfabet x dan dibunyikan sesuai nada dasar x3 dengan kecepatan sesuai keinginan pengguna. Implementasi dari algoritma berikut dapat dituliskan melalui pseudocode sebagai berikut

```
procedure playMidi(byte [] a,int vel)
{Fungsi ini memainkan representasi hexadecimal yang
 telah dibuat ke dalam musik midi}

DEKLARASI
    i : integer
    j : integer
    tmp : char

ALGORITMA
    tmp = 0
    for i=1 to (a.length()) do
        if (a[i]<65) then
            if (i == 1) then
                play (“00”,vel)
                tmp = ‘C’
                for j = 2 to (chr) a[i] do
                    play (“C3”, vel)
                    // hal ini ditujukan apabila karakter
                    pertama yang dibaca merupakan
                    angka. Musik yang dimainkan
                    pertama kali adalah berhenti satu
                    ketuk dan memainkan nada C3
                    sebanyak j kali
                endfor

            else
                for j = 2 to (chr) a[i] do
                    play((chr)tmp+”3”,vel)
                    // memainkan nada sebelumnya
                    sebanyak j-1 kali
                endfor
            endif
        else
        //karakter yang ditemukan karakter alfabet
            tmp = (chr) a[i]
            play((chr)tmp+”3”,vel)
            //hal ini untuk mengantisipasi munculnya
            karakter alfabet secara berbarengan
        endif
    endfor
procedure writeToFile(byte [] a,int vel)
{Fungsi ini menuliskan representasi not ke dalam file}

DEKLARASI
    i : integer
    j : integer
    tmp : char

ALGORITMA
    tmp = 0
    for i=1 to (a.length()) do
        if (a[i]<65) then
            if (i == 1) then
                write(‘.’)
                tmp = ‘C’
                for j = 2 to (chr) a[i] do
                    write(‘C’)
                    // hal ini ditujukan apabila karakter
                    pertama yang dibaca merupakan
                    angka. Musik yang dituliskan pertama
                    kali adalah berhenti satu ketuk dan
                    memainkan nada C sebanyak j kali
                endfor

            else
                for j = 2 to (chr) a[i] do
                    write((chr)tmp)
                    // memainkan nada sebelumnya
                    sebanyak j-1 kali
                endfor
            endif
        else
        //karakter yang ditemukan karakter alfabet
            tmp = (chr) a[i]
            write((chr)tmp)
            //hal ini untuk mengantisipasi munculnya
            karakter alfabet secara berbarengan
        endif
    endfor
```

3.2 Dekripsi file

Pada bagian ini akan dijelaskan bagaimana proses dekripsi yang dilakukan terhadap file yang dipilih

3.2.1 Konversi file input menjadi binary file

Pada algoritma ini setiap karakter file input akan dikonversi menjadi representasi binernya di mana untuk satu karakter akan direpresentasikan dalam 4 digit bilangan biner. Aturan dekripsi pada file input untuk dekripsi disesuaikan dengan proses penulisan tangga nada pada proses enkripsi sebelumnya.Implementasi dari konversi file input menjadi binary file dijelaskan melalui pseudocode sebagai berikut

```
function fileToHex(String a) → byte[]
{Fungsi ini mengembalikan representasi hexadecimal dari file yang ingin di dekripsi}

DEKLARASI
    i : integer
    count: integer
    tmp : byte
    cek : boolean

ALGORITMA
    i=0
    j=0
    cek = false
    while (i<a.Length()) do
        if (a[i]='.') then
            if (i=0) then
                tmp='C'
            count = 0
        else
            if (a[i] = tmp) then
                count = count +1
            else
                if (count = 1 ) then
                    fileToHex[j] = ord(tmp)
                    j = j +1
                else
                    fileToHex[j] = ord(tmp)
                    j = j + 1
                    fileToHex[j] = ord(count)
                    count = 0
            endif
        endif
        i = i+1
    endwhile
```

3.1.2 Konversi dari representasi hexadecimal menjadi binary file

Setelah file input dikonversi menjadi bentuk hexadecimal, file-file ini kemudian masuk ke dalam konversi ke dalam bentuk biner. Algoritma konversi dari hexadecimal ke dalam biner dapat dijelaskan melalui pseudocode sebagai berikut

```
function hexaTofourBtye(byte a) → byte[]
{Fungsi ini mengembalikan satu digit representasi hexadecimal menjadi empat digit representasi binernya}

DEKLARASI
    i : integer

ALGORITMA
    if ( a>65) then
        a = a - 55
    else
        a = a - 48

    i = 3
    while (a>1) do
        hexaToFourByte[i] = a mod 2
        i = i - 1
        a = a div 2
    endwhile
    hexaToFourByte[i] = a
    i = i – 1

    while (i>=0) do
        hexaToFourByte[i] = 0
        i = i - 1
    endwhile
```

```
function hexaToByte(byte[] a) → byte []
{Fungsi ini mengembalikan representasi hexadecimal menjadi representasi biner}

DEKLARASI
    i : integer
    j : integer
    tmp : byte[4]

ALGORITMA
    c = 0

    for i=1 to a.length() do
        tmp = hexaToFourByte(a[i])
        hexaToByte[i*4 + 0] = tmp [0]
        hexaToByte[i*4 + 1] = tmp [1]
        hexaToByte[i*4 + 2] = tmp [2]
        hexaToByte[i*4 + 3] = tmp [3]
    endfor
```

3.2.2 Konversi binary file menjadi representasi aslinya

Proses ini mengembalikan hasil konversi setiap karakter pada file input yang direpresentasikan menjadi 8 digit file binary. Implementasi dari algoritma berikut dapat dituliskan melalui pseudocode sebagai berikut

```
function toFile(byte[] a) → String
{Fungsi ini mengembalikan representasi byte dari suatu string}
```

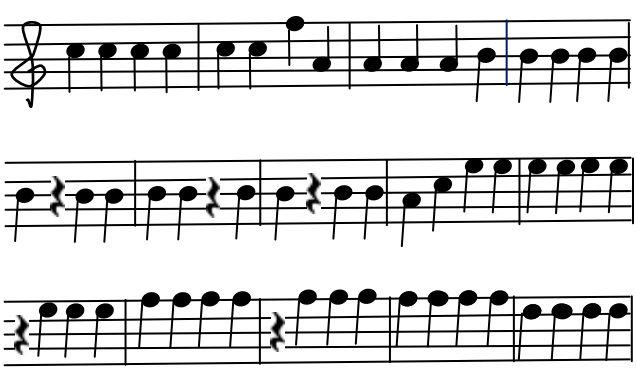
```
DEKLARASI
    i : integer
    j: integer
    tmp : byte[]

ALGORITMA
    for i=1 to (a.length()/8) do
        tmp = copyByte(a[i*8],a[i*8 + 7]
        toFile[i] = getChar (tmp)
    endfor
```

IV. PEMBAHASAN DAN ANALISIS

Penggunaan algoritma konversi dari suatu file menjadi sebuah file musik bertipe MIDI merupakan salah satu cara baru yang dapat digunakan untuk menyembunyikan pesan atau data rahasia. Ketika seseorang mendengerkan file bertipe MIDI, orang tersebut belum tentu akan menyangka bahwa musik tersebut merupakan pesan/ data rahasia yang diproses dengan algoritma steganografi.

Hasil implementasi dari penggunaan algoritma ini adalah suara dalam bentuk MIDI dan file yang menunjukkan nada-nada yang dimainkan dalam representasi karakter alfabet kunci nada dasar yang dimainkan. Jika kita perhatikan dari pola penulisan/ penghasilan nada musik MIDI yang digunakan, cara yang digunakan untuk menerjemahkan hasil hexadecimal terhadap file musik midi yang digunakan masih sangat sederhana sehingga file musik yang dihasilkan akan berbentuk sebuah file musik MIDI yang tidak terlalu menarik untuk didengar.

Kita ambil saja salah satu hasil pengujian yang menghasilkan C6FA4B6422ACE63F47D4. Hal ini kemudian akan direpresentasikan dalam bentuk tangga nada MIDI yang dimainkan sebagai berikut
CCCCCCFAAAABBBBBB.BBBB.BB.BBACEEEEEE.EEEFFFF.FFFFFFFDDDD
Bentuk ini jika direpresentasikan dalam bentuk midi yang direpresentasikan dalam bentuk partitur menjadi



Dari hasil yang diperoleh di atas, dapat kita temukan bahwa musik yang tercipta belum tentu enak untuk didengar. Karena ada kemungkinan untuk keluar karakter angka secara berturut-turut yang cukup besar, pola musik tersebut bisa saja menghasilkan musik dengan nada yang sama selama selang waktu yang cukup lama.

Walaupun hasil yang dihasilkan belum bisa kita pastikan akan menghasilkan sebuah musik MIDI yang memiliki melodi/ harmoni yang enak untuk didengar, kita memanfaatkan musik ini untuk menutupi data/pesan rahasia yang ingin kita sampaikan tersebut. Alasan bahwa kebanyakan orang tidak akan menyangka bahwa musik MIDI yang dihasilkan ini merupakan konversi dari sebuah data yang ingin kita rahasiakan inilah yang kita manfaatkan. Apabila sebuah file musik MIDI tidak enak untuk didengarkan biasanya akan menimbulkan asumsi bahwa orang yang membuat komposisi musik tersebut belum berpengalaman dalam membuat komposisi musik. Akan sangat jarang bagi pendengar musik yang akan langsung terpikir bahwa musik tersebut merupakan pesan rahasia.

Selain itu, implementasi yang digunakan dalam penciptaan musik MIDI tersebut dirancang dengan algoritma yang berbeda dari representasi hexadecimal ke musik MIDI kebanyakan di mana biasanya dua buah digit hexadecimal dapat kita representasikan menjadi satu buah nada dalam musik MIDI sehingga algoritma yang digunakan akan menjadi lebih sulit untuk ditebak polanya.

Penggunaan algoritma ini cukup sederhana karena menggunakan representasi dalam byte sehingga dapat digunakan untuk menyembunyikan berbagai jenis data mulai dari data sederhana yang hanya bertipe file teks maupun data-data digital yang bertipe gambar, suara , maupun tulisan. Perbedaannya hanya akan terlihat pada lama waktu pemrosesan yang dibutuhkan untuk melalukan enkripsi file tersebut ke dalam file MIDI yang tergantung pada banyaknya representasi biner dari suatu file.

Proses dekripsi dilakukan terhadap file eksternal yang dihasilkan dari proses enkripsi sebelumnya. Apabila ingin dilakukan proses dekripsi dari file MIDI, dapat dilakukan sedikit modifikasi dari algoritma dekripsi yang dijelaskan pada makalah ini dengan memanfaatkan sinyal processing dari file suara.

Proses dekripsi tersebut dilakukan dengan memutarbalikkan alur konversi yang digunakan pada saat melakukan dekripsi sehingga dapat dihasilkan hasil yang serupa dengan file yang digunakan untuk proses enkripsi sebelumnya.

Pendekatan yang dilakukan pada algoritma baru yang dirancang pada makalah ini bisa dibilang berbeda dari pendekatan yang dilakukan untuk pemrosesan algoritma steganografi atau kriptografi kebanyakan. Oleh karena itu, penggunaan algoritma ini bisa terbilang cukup efektif untuk digunakan dalam penyembunyian data/pesan rahasia tertentu.

Dengan memanfaatkan unsur seni ke dalam usaha penyembunyian rahasia, diharapkan orang yang mendengar hasil dari musik MIDI tersebut masih tidak menyadari

bahwa musik yang didengarnya tersebut merupakan pesan rahasia sehingga informasi/data rahasia yang ingin disampaikan tersebut hanya dapat diterima oleh orang yang mengetahui cara untuk melakukan dekripsi dari file tersebut.

V. KESIMPULAN

Pemanfaatan steganografi dalam menyembunyikan pesan rahasia masih tergolong cara yang cukup efektif. Salah satu cara yang bisa digunakan adalah menyembunyikan file/data rahasia menjadi suatu file musik. File / data rahasia yang ingin disembunyikan dapat dikonversi menjadi representasi hexadecimal yang kemudian dengan aturan tertentu kita representasikan dalam file musik berekstensi MIDI. Hal ini masih cukup aman untuk digunakan karena tidak banyak orang yang akan menyadari bahwa sebuah file MIDI yang mereka dengarkan itu merupakan hasil konversi dari sebuah pesan/data rahasia.

REFERENSI

[1] Huber, David Miles. "The MIDI Manual". Carmel, Indiana: SAMS, 1991

[2] http://grouplab.cpsc.ucalgary.ca/cookbook/index.php/VisualStudio/HowToPlayMIDIInstruments diakses pada Minggu, 24 Maret 2013

[3] http://www.wavosaur.com/download/midi-note-hex.php diakses pada Senin, 25 Maret 2013

[4] http://www.pianostreet.com diakses pada Senin, 4 Maret 2013

[5] www.binaryhexconverter.com diakses pada Senin, 4 Maret 2013

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah ini merupakan tulisan asli, bukan adaptasi dari jurnal pihak lain,dan bukan plagiarisme.

Bandung, 26 Maret 2013

Martha Monica
(13510080)