

Laporan Tugas Besar 1

IF4020 Kriptografi

Steganografi



Yoga Adrian 13513030

Kevin Yauris 13513036

Feryandi Nurdiantoro 13513042

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2015**

1. Teori Singkat

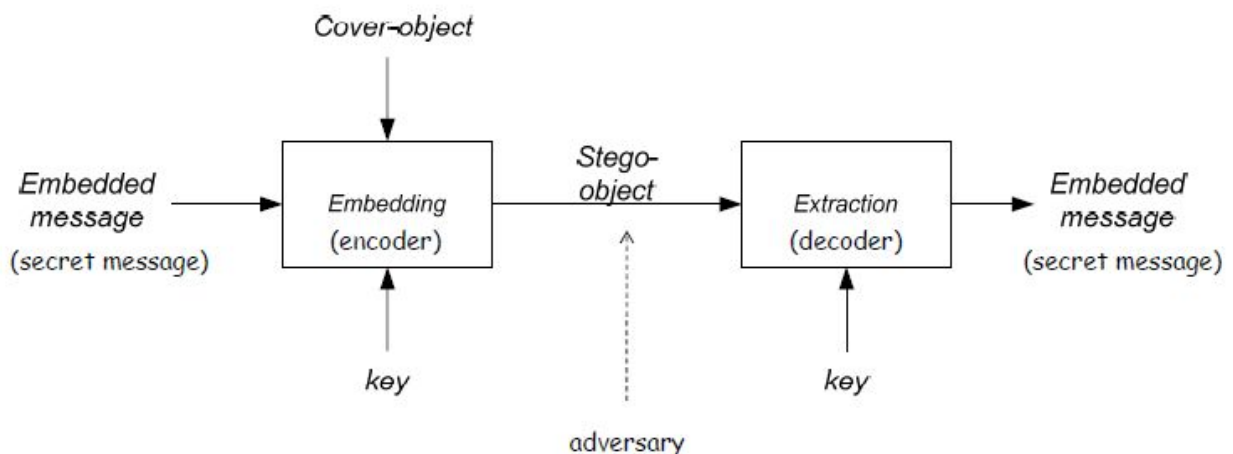
1.1. Steganografi

Deksripsi

Adalah ilmu dan seni menyembunyikan pesan rahasia dengan suatu cara sedemikian sehingga tidak seorang pun yang mencurigai keberadaan pesan tersebut. Jika dalam Kriptografi isi pesan disembunyikan, maka dalam Steganografi pesan itu sendirilah yang disembunyikan. Ada banyak macam bentuk steganografi. Banyak bentuk steganografi yang dilakukan pada masa lampau.

Namun untuk saat ini, bentuk yang paling terkenal adalah Steganografi Digital. Steganografi Digital adalah penyembunyian pesan digital di dalam dokumen digital lainnya. jadi sebuah pesan digital(bisa berupa audio, gambar, teks) disisipkan ke sebuah pesan digital lainnya (bisa berupa audio, gambar, teks).

Diagram Proses Steganografi



Proses steganografi di jelaskan pada diagram di atas.

- Embedded message (hiddentext) atau secret message: pesan yang disembunyikan. Bisa berupa teks, gambar, audio, video, dll
- Cover-object (covertext): pesan yang digunakan untuk menyembunyikan embedded message. Bisa berupa teks, gambar, audio, video, dll
- Stego-object (stegotext): pesan yang sudah berisi pesan embedded message.

- key: kunci yang digunakan untuk menyisipkan pesan dan mengekstraksi pesan dari stegotext.
- Adversary: pihak yang biasanya tidak boleh mengetahui adanya pesan tersembunyi

Kriteria Steganografi

Agar tidak dapat diketahui oleh pihak lain dan efektif, ada beberapa kriteria yang menyatakan sebuah steganografi bagus atau tidak. Yaitu:

- Imperceptible. Keberadaan pesan rahasia tidak dapat dipersepsi secara visual atau secara audio (untuk stego-audio).
- Fidelity. Kualitas cover-object tidak jauh berubah akibat penyisipan pesan rahasia.
- Recovery. Pesan yang disembunyikan harus dapat diekstraksi kembali.
- Capacity. Ukuran pesan yang disembunyikan sedapat mungkin besar

1.2. Metoda Modifikasi LSB

Deskripsi

Metode ini memanfaatkan kelemahan indra visual manusia dalam mengamati dan mengetahui perubahan warna yang terjadi pada gambar. Cara yang digunakan adalah dengan mengganti bit LSB dari pixel dengan bit pesan. Karena mengubah 1 nilai bit tidak akan berpengaruh besar dan hanya mengubah persepsi warna sangat kecil untuk mata manusia

Contoh:

Bit pada gambar	1000010 1 1110010 0 00101010 0
Bit pesan	101
Bit pesan stegano	1000010 1 1110010 0 00101010 1

Variasi Metode

Beberapa variasi metode LSB adalah,

1. Sekuensial

Dengan cara ini, pesan dimasukkan ke dalam gambar dengan urutan yang sekuensial dari awal hingga akhir pesan.

2. Acak

Dengan menggunakan suatu *seed* atau *Pseudo-Random Number Generator* untuk membuat bilangan acak yang dijadikan *seed* untuk menentukan peletakan pesan secara acak pada gambar.

3. *m*-bit LSB

Untuk memperbesar kapasitas penyimpanan data, bit yang digunakan untuk menyimpan pesan dapat digunakan sejumlah *m*. Namun dengan cara ini, kualitas gambar dapat turun dan gambar yang terisi pesan semakin mudah terlihat perbedaannya jika *m* yang digunakan cukup besar.

4. Enkripsi XOR

Pesan dienkripsi terlebih dahulu menggunakan metode XOR terhadap kunci-stego-nya. Jumlah bit kunci harus sama dengan jumlah bit pesan. Setelah itu pesan baru dimasukkan ke dalam citra.

1.3. Citra Bitmap

Deskripsi

BMP atau *bitmap image* merupakan format file yang digunakan untuk menyimpan data suatu gambar bit per bit dan independen terhadap jenis device yang membacanya. Bitmap merupakan salah satu jenis file gambar yang mudah untuk dimanipulasi karena tidak menggunakan kompresi untuk penyimpanannya sehingga bit-bit yang ada mudah dibaca dan dimanipulasi.

Struktur

Struktur bitmap yang umum diketahui untuk citra berwarna tanpa alpha adalah sebagai berikut,

Letak	Nilai Hexa	Nilai	Deskripsi
BMP Header			
0h	42 4D	"BM"	Identifikasi format file
2h	46 00 00 00	70 byte (54+16)	Ukuran file BMP
6h	00 00	Tidak digunakan	Tergantung aplikasi
8h	00 00	Tidak digunakan	Tergantung aplikasi
Ah	36 00 00 00	54 byte (14+40)	Letak dimana array byte pixel dapat ditemukan
DIB Header			
Eh	28 00 00 00	40 byte	Jumlah DIB header
12h	02 00 00 00	2 pixel (kiri ke kanan)	Lebar citra dalam bit
16h	02 00 00 00	2 pixel (bawah ke atas)	Tinggi citra dalam bit
1Ah	01 00	1 plane	Jumlah color plane yang digunakan
1Ch	18 00	24 bits	Jumlah bit per pixel
1Eh	00 00 00 00	0	BI_RGB, tidak ada kompresi yang digunakan
22h	10 00 00 00	16 bytes	Ukuran bitmap beserta padding
26h	13 0B 00 00	2835 pixels/meter horizontal	Resolusi gambar untuk percetakan
2Ah	13 0B 00 00	2835 pixels/meter vertical	
2Eh	00 00 00 00	0 colors	Jumlah warna di palet
32h	00 00 00 00	0 important colors	0 artinya semua warna, penting
Data Bitmap			

36h	00 00 FF	0 0 255	Merah, Pixel (0,1)
39h	FF FF FF	255 255 255	Putih, Pixel (1,1)
3Ch	00 00	0 0	Padding untuk membulatkan ke 1 byte
3Eh	FF 00 00	255 0 0	Biru, Pixel (0,0)
41h	00 FF 00	0 255 0	Hijau, Pixel (1,0)
44h	00 00	0 0	Padding untuk membulatkan ke 1 byte

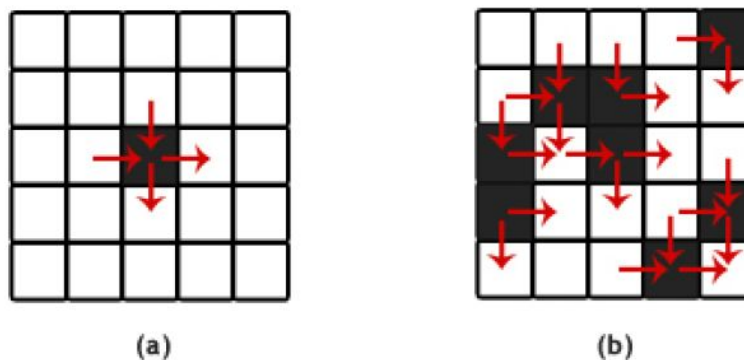
1.4. Bit-Plane Complexity Segmentation

Deskripsi

Bit-Plane Complexity Segmentation (BPCS) adalah metode Steganografi yang memiliki kapasitas lebih besar daripada metode modifikasi LSB. Jika metode modifikasi LSB hanya meletakkan pesan ke bit-plane terakhir, metode BPCS ini meletakkan pesan ke bit-plane yang memiliki kompleksitas tinggi (noise-like region).

Kompleksitas Bit-Plane

Kompleksitas bit-plane diukur dari pergantian warna hitam dan putih dalam suatu bitplane. Angka tersebut lalu dibandingkan dengan kemungkinan pergantian warna putih dan hitam dalam bit-plane yang berukuran sesuai bit-plane asal



Pada bit-plane a, pergantian warna hitam dan putih adalah 4. Pada bit-plane b, pergantian warna hitam dan putih adalah 20. Sedangkan jumlah maksimum pergantian warna hitam dan putih dalam bit-plane 5x5 adalah 40. angka tersebut didapat dari rumus $2 * sisi * (sisi - 1)$. Jadi kompleksitas bit-plane a adalah $1/40$ atau 0.025. sedangkan kompleksitas bit-plane b adalah $20/40$ atau 0.5.

Algoritma

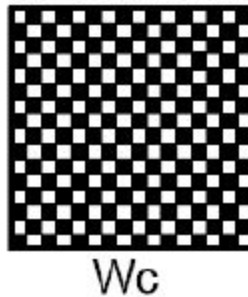
Berikut adalah langkah-langkah mengenkripsi atau membentuk stego-image. Asumsikan citra berukuran kelipatan 8. Jika bukan kelipatan 8, tambahkan pixel-pixel semu sehingga berukuran 8 x 8. pixel yang semu tersebut biasanya diisi dengan 0

1. Bag icovert-image menjadi blok 8 x 8 pixel.
2. Bentuk setiap blok 8 x 8 pixel menjadi sistem PBC yang terdiri dari 8 buah bit-plane.
3. Ubah sistem PBC menjadi sistem CGC (Canonical Gray Coding). hal ini dilakukan kepada setiap bit-plane dengan menggunakan persamaan ini.

$$\begin{aligned} g_1 &= b_1 \\ g_i &= b_{i-1} \wedge b_i \end{aligned}$$

g adalah hasil biner sistem CGC. b adalah biner sistem PBC yang akan diubah

4. Tentukan apakah setiap bit-plane merupakan informative region atau noise-like region dengan menggunakan nilai ambang . Nilai default nilai ambang adalah 0.3. Jika tergolong noise-like region, maka pesan bisa disisipkan pada bit-plane tersebut, tetapi jika termasuk informative region, maka tidak dapat digunakan untuk menyisipkan pesan.
5. Bagi pesan menjadi segmen-segmen berukuran 64-bit, lalu nyatakan segmen menjadi blok biner berukuran 8 x 8.
6. Jika blok pesan S tidak lebih kompleks dibandingkan dengan nilai ambang (yaitu termasuk kategori informative region), lakukan konjugasi terhadap S untuk mendapatkan S* yang lebih kompleks. konjugasi dilakukan dengan xor setiap bit dalam blok pesan S tersebut dengan bit-plane yang memiliki biner seperti papan catur yang dimulai dari putih.



Jika P adalah blok pesan S yang ingin di konjugasi, maka $P^* = P \wedge Wc$ dimana P^* adalah hasil konjugasi blok pesan S

7. Sisipkan segmen pesan 64-bit ke bit-plane yang merupakan noise-like region dengan cara mengganti seluruh bit pada noise-like region tersebut dengan 64-bit pesan).
8. Jika bloks S dikonyugasi, simpan pesan pada “conjugation map”.
9. Sisipkan juga pemetaan konjugasi yang telah dibuat.
10. Ubah stego-image dari sistem CGC menjadi sistem PBC.hal ini dilakukan kepada setiap bit-plane dengan menggunakan persamaan ini.

$$\begin{aligned} b_1 &= g_1 \\ b_i &= b_{i-1} \wedge g_i \end{aligned}$$

b adalah hasil biner sistem PBC. g adalah biner sistem CGC yang akan diubah

Untuk mengekstraksi pesan dari stego-image, dilakukan langkah-langkah berikut

1. Bagi stego-image menjadi blok 8 x 8 pixel.
2. Bentuk setiap blok 8 x 8 pixel menjadi sistem PBC yang terdiri dari 8 buah bit-plane.

3. Ubah sistem PBC menjadi sistem CGC(Canonical Gray Coding).
4. Hitung kompleksitas setiap bit-plane. Jika kompleksitasnya diatas nilai ambang, maka bit-plane tersebut bagian dari pesan. Tabel konyugasi yang disisipkan juga dibaca untuk melihat proses konyugasi yang perlu dilakukan pada tiap blok pesan.
5. Jika Kompleksitas bit-plane tersebut diatas nilai ambang, maka ambil setiap bit dari bit-plane tersebut. Jika pada tabel konyugasi yang disisipkan terdapat bit-plane tersebut, lakukan konyugasi terlebih dahulu. Bit-bit yang didapat dari bit-plane itulah yang akan diambil untuk diekstraksi menjadi pesan rahasia atau *embedded message*.

2. Perancangan dan Implementasi

Untuk membaca file biner agar byte-byte nya dapat dibaca.

- **FileReader**: kelas ini berguna untuk mendapatkan byte dari file biner. Pada program ini kelas ini berguna untuk mendapatkan byte dari cover-image yang ingin dijadikan tempat pesan. Kelas ini juga berguna untuk mendapatkan byte dari file pesan yang ingin disisipkan

Untuk membaca byte-byte dan memproses byte-byte tersebut sesuai bmp file. Alur dan algoritma BPCS akan diimplementasi didalam kelas-kelas ini.

- **Bitmap**: kelas ini merepresentasikan informasi file bmp sesuai byte yang sudah didapatkan. Kelas ini juga memiliki fungsi untuk mengubah informasinya sesuai pesan yang akan disisipkan. Atribut utama dari kelas ini adalah blok, karena pada algoritma BPCS, cover-image akan dibagi menjadi block 8x8
- **Block**: kelas ini merepresentasikan blok blok yang dibagi dari cover-image. Atribut utama adalah bit-plane yang berisi informasi bit-bit dari byte tiap pixel. jumlah bit-planenya bergantung dengan tipe gambar. Untuk gambar yang berwarna (BGR) setiap pixel terdapat 3 byte dalam informasi file bmp. Untuk gambar *greyscale*, setiap pixel terdapat 1 byte.
- **Plane**: Kelas ini merepresentasi kan bit-plane. kelas ini berisi data-data karakter “1” atau “0” yang merepresentasikan potongan bit dari bloknya. pesan-pesan yang sudah dibagi tiap segmen akan dimasukkan ke data kelas ini. Perhitungan kompleksitas, prosedur pembentukan PBC dan CGC, conjugate, dan pergantian data sesuai pesan yang disisipkan akan diimplementasi disini.

Dikelas-kelas itu juga terdapat fungsi untuk mengembalikan ke bitmap sesuai plane-plane yang sudah disisipi pesan.

Karena kelas Bitmap, Block, dan Plane bergantung dengan informasi file BMP (yang kemungkinan besar hanya bisa dilakukan ketika cover-image nya adalah BMP), maka terdapat kelas ImageConverter untuk mengubah file PNG menjadi BMP. sehingga konten gambarnya sama, hanya informasi seperti header yang diubah.

- **ImageConverter**: Kelas ini berguna untuk mengubah format gambar dari sebuah format ke format lain. Dalam program berguna untuk mengubah gambar PNG ke format gambar BMP.

Untuk merepresentasikan informasi pesan yang akan disisipi.

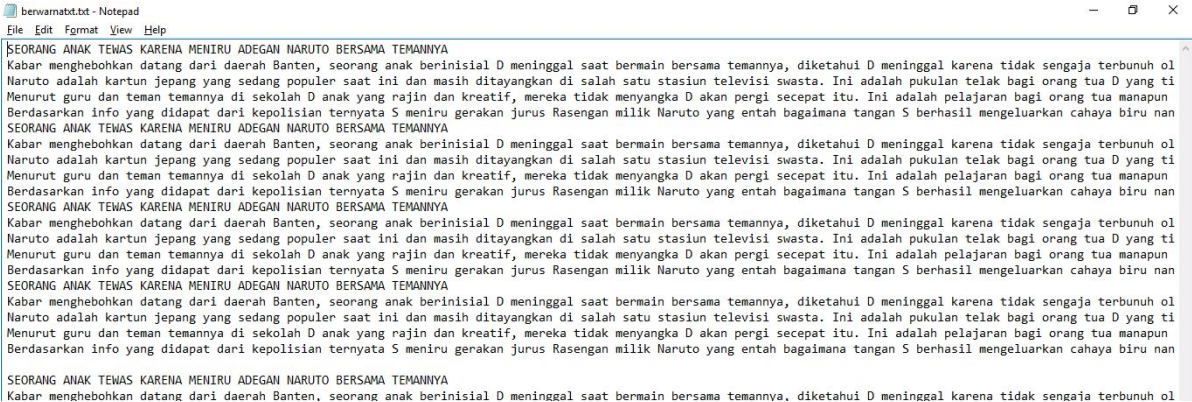
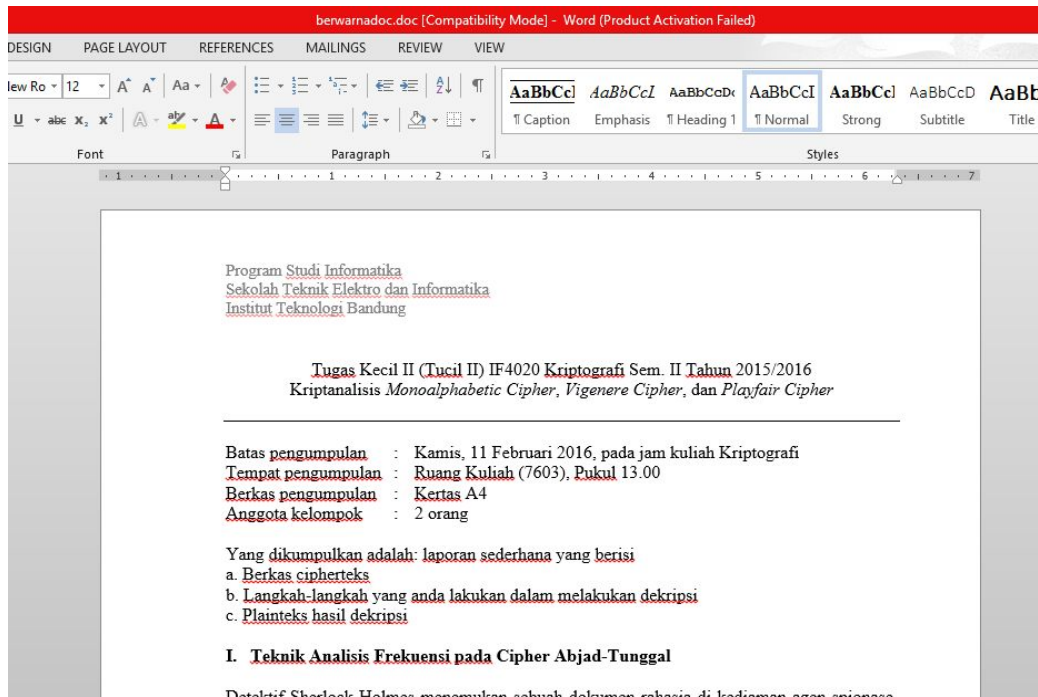
- **StringBlock**: kelas ini berguna untuk representasi pesan yang akan disisipkan. pesan akan dibagi menjadi segmen 64-bit. Atribut utama adalah Plane yang berisi tiap segment 64-bit. kelas ini juga berfungsi untuk membentuk conjugate map yang akan disisipkan juga ke stego-image


Karena pesan yang disisipkan bisa dienkripsi dengan metode kriptografi(vigenere extended), terdapat kelas:



- **CrypterTucil1**: kelas ini diambil dari tugas kecil 1 Kriptografi. kelas ini berisi algoritma enkripsi dan dekripsi kriptografi vigenere yang akan digunakan untuk mengubah byte-byte pesan yang akan disisipkan

Sedangkan untuk tampilan program dibuat dalam kelas mainMenu, kelas ini berfungsi sebagai main dan juga untuk menampilkan preview gambar dan menerima input pengguna. Terbagi atas 2 bagian utama yaitu untuk melakukan penyisipan pesan rahasia ke dalam gambar dan untuk melakukan ekstraksi pesan dari gambar yang telah disisipi pesan rahasia.



3. Pengujian Program dan Analisis Hasil

File Uji	
TXT File	10 KB
	
DOC File	67 KB
	



Homogen	
Original (Cover Image)	
	
Ukuran Maksimum	147400 B
Ukuran Gambar	512 x 256 pixel
Threshold	0.3

Hide TXT File	
File Size	10 KB
PSNR	53.208744
	
Hide DOC File	
File Size	67 KB
PSNR	32.971024
	



Heterogen	
Original (Cover Image)	
	
Ukuran Maksimum	424472 B
Ukuran Gambar	512 x 384 pixel
Threshold	0.3

Hide TXT File	
File Size	10 KB
PSNR	56.068356
	
Hide DOC File	
File Size	67 KB
PSNR	33.55837
	

Grayscale	
Original (Cover Image)	
	
Ukuran Maksimum	169944 B
Ukuran Gambar	512 x 512 pixel
Threshold	0.3

Hide TXT File	
File Size	10 KB
PSNR	53.986706
	
Hide DOC File	
File Size	67 KB
PSNR	34.74929
	

Berwarna	
Original (Cover Image)	
	
Ukuran Maksimum	532616 B
Ukuran Gambar	512 x 512 pixel
Threshold	0.3

Hide TXT File	
File Size	10 KB
PSNR	58.85451
 <p>File Size: 786436 bytes File extension : bmp</p>	
Hide DOC File	
File Size	67 KB
PSNR	37.494404
 <p>File Size: 786436 bytes File extension : bmp</p>	

Kesimpulan

Penyembunyian data pada suatu gambar dapat dilakukan dengan metode BPCS sehingga melipat gandakan ukuran yang dapat ditampung oleh gambar. Namun dibutuhkan *seed* dan fungsi random yang cukup lebih baik untuk membuat persebaran pesan pada gambar menjadi merata sehingga nilai PSNR naik dan mata dapat terkecoh dengan gambar yang telah di sisipkan pesan sehingga tidak dapat membedakannya dengan citra aslinya.