Teknik Stenografi dengan Menggunakan Metode *Visual Attacks* dan S*tatistical Attacks*

Nurul Fuad, Suyono, Ir. Endang Setyati, MT
Dosen STT Surabaya

ABSTRAK

Keamanan data merupakan hal yang sangat penting dalam menjaga kerahasiaan informasi terutama yang berisi informasi sensitif yang hanya boleh diketahui isinya oleh pihak yang berhak saja, apalagi jika pengirimannya dilakukan melalui jaringan publik (Internet), apabila data tersebut tidak diamankan terlebih dahulu, maka akan sangat mudah disadap dan diketahui isinya oleh pihak- pihak yang tidak berhak.

Salah satu cara yang digunakan untuk pengamanan data adalah menggunakan sistem Stenografi. Banyak metoda stenografi yang melekatkan sejumlah besar informasi rahasia di dalam pixel pada cover image. Samaran yang digunakan oleh mayoritas utilitas steganografi mempunyai kelemahan pokok pada komunikasi rahasia. Cara mengatasi kelemahan pada algoritma stenografi tersebut adalah pengembangan serangan untuk menilai tingkat keamanannya. Ada dua macam serangan yang dapat digunakan, yaitu Visual Attacks dan Statistical Attacks. Serangan visual, untuk menjelaskan perbedaan antara noise dan visual patterns, sedangkan serangan statistik untuk mendeteksi metode stenografi yang digunakan.

***Kata Kunci** : Stenografi, Visual Attacks, Statistical Attacks. Noise, visual patterns.*

ABSTRACT

Data security is the main point to keep the confidentiality of information, specifically the data which contain of sensitive information that should be known by the eligible one. Furthermore, when on line system(network system) is used to deliver, thus if the data is not protected before it will have any risk.

One of the ways used to secure data is by using Steganography system. Thereare some methods of steganographic that utilized a large amount of confidential informations in the pixels of the cover image. Pseudonym used by the majority of steganography tools fundamentally have some weaknesses in confidentiality communication. Then, attack development is used to overcome with the weaknesses of steganographic algorithm in case of assessing security system. There are two kinds of attacks that can be used, namely Visual and Statistical Attacks. Visual attacks, to explain the difference between noise and visual patterns, while the statistical attacks to detect steganographic method which is used.

***Keywords:** stenographic, Visual Attacks, Statistical Attacks. Noise, visual patterns.*

KAJIAN TEORI

1. Pengertian Steganografi

Steganografi merupakan ilmu yang mempelajari, meneliti, dan mengembangkan seni menyembunyikan suatu informasi. Steganografi dapat digolongkan sebagai salah satu bagian dari ilmu komunikasi. Pada era informasi digital, steganografi merupakan teknik dan seni menyembunyikan informasi dan data digital dibalik informasi digital lain, sehingga informasi digital yang sesungguhnya tidak kelihatan. Namun saat ini telah di-ketahui ada metode yang dapat melakukan serangan- serangan terhadap stenografi dengan memanfaatkan kelemahan stenografi.

Metode tersebut yaitu: Visual Attacks dan Statistical Attacks. Serangan visual (*visual attacks*), untuk menjelaskan perbedaan antara noise dan visual patterns, sedangkan serangan statistik (statistical attacks) untuk mendeteksi metode steganografi yang digunakan. Karena

telah ditemukannya metode penyerangan pada stenografi, timbul masalah bagaimana memberikan keamanan pada suatu data agar data selain dapat disembunyikan, dapat pula terjaga kerahasiaan isinya dari pihak yang tidak berwenang untuk mengaksesnya. Dengan pertimbangan tersebut, maka pengenkripsian data dilakukan sebelum data disembunyikan. Pengenkripsian yang dilakukan adalah dengan menggunakan algoritma Rijndael sedangkan LSB (Least Significant Bit).

Serangan visual di sini menerangkan bahwa pada *least* EzStego v2.0b3, Jsteg v4, Steganos v1.5, dan S-Tools v4.0 mempunyai kelemahan *misassumption,* bahwa *least significant bits* pada data *image* adalah *noise* yang tidak terhubungkan. Selain itu juga menunjukkan metoda yang lebih obyektif untuk mendeteksi steganografi dengan cara statistik.

TUJUAN

Tujuan dari penulisan paper ini adalah untuk meningkatkan keamanan pada sistem steganografi dengan menggunakan metode ***Visual Attacks*** dan ***Statistical Attacks*****.**

Sistem File Stenografi

File Gambar

Pada komputer, suatu gambar adalah suatu array dari bilangan yang merepresentasikan intensitas terang pada point yang bervariasi (pixel). Pixel ini menghasilkan *raster data* gambar. Suatu ukuran gambar yang umum adalah 640 x 480 pixel dan 256 warna (atau 8 bit per pixel). Suatu gambar akan berisi kira-kira 300 kilobit data. Gambar digital disimpan juga secara khusus di dalam file 24-bit atau 8-bit. Gambar 24-bit menyediakan lebih banyak ruang untuk menyembunyikan informasi; bagaimanapun, itu dapat sungguh besar (dengan perkecualian gambar JPEG).

Kompresi File

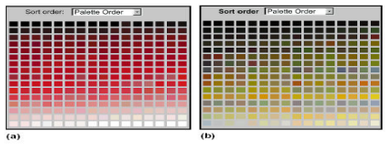
Dua kandungan dari kompresi adalah *lossless* dan *lossy*. Kedua metoda ini menghemat ruang penyimpanan tetapi mempunyai hasil yang berbeda, yang bertentangan dengan penyembunyian informasi. Kompresi *lossless* membiarkan kita merekonstruksi pesan asli yang sama; oleh karena itu, lebih disukai ketika informasi asli harus tetap utuh (seperti dengan gambar steganography). Kompresi *lossless* khusus untuk gambar yang tersimpan sebagai GIF (*Graphic Interchange Format*) dan BMP 8-bit (file bitmap Microsoft Windows dan OS/2).

Kompresi *lossy*, pada penanganan lainnya, menghemat ruangan tetapi tidak menjaga integritas gambar aslinya. Metoda ini secara khusus untuk gambar yang tersimpan sebagai JPEG (*Joint Photographic Experts Group*).

Embedding Data

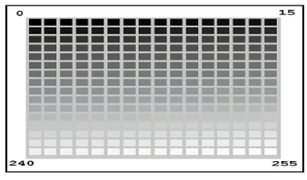
Data embedded, yang tersembunyi dalam suatu gambar membutuhkan dua file. Pertama adalah gambar asli yang belum modifikasi yang akan menangani informasi tersembunyi, yang disebut *cover image*. File kedua adalah informasi pesan yang disembunyikan. Suatu pesan dapat berupa plaintext, chipertext, gambar lain, atau apapun yang dapat ditempelkan ke dalam *bit-stream*. Ketika dikombinasikan, *cover image* dan pesan yang ditempelkan membuat *stego-image*. Suatu *stego-key* (suatu password khusus) juga dapat digunakan secara tersembunyi, pada saat decode selanjutnya dari pesan.

Dalam gambar 8-bit warna seperti file GIF, setiap pixel direpresentasikan sebagai byte tunggal, dan setiap pixel selalu menunjuk ke tabel indek warna (*palette*) dengan 256-kemungkinan warna. Nilai pixel adalah diantara 0 dan 255. Software secara sederhana menggambarkan indikasi warna pada palette merah, menggambarkan perubahan yang sulit dipisahkan dalam variasi warna: perbedaan visualisasi diantara banyak warna yang sulit. Gambar 2.1(b) menunjukkan perubahan warna yang sulit dipisahkan dengan baik.



Gambar 1. Representasi Warna Pallette
(Sumber: Exploring Steganography: Seeing the Unseen)

Banyak pakar stenography merekomendasi-kan penggunaan gambar yang meliputi 256 *shade* gray. Gambar gray-scale lebih disukai karena perubahan keteduhan sangat gradual dari byte ke byte, dan lebih sedikit perubahan nilai diantara masukan palette, dimana mereka dapat menyembunyikan informasi lebih baik. Gambar 2 menunjukkan suatu palette gray-scale dari 256 *shade*.



Gambar 2. Representasi gray-scale palette dari 256 shade
(Sumber: Exploring Steganography: Seeing the Unseen)

Rahasia di Dalam Gambar Digital

Banyak cara untuk menyembunyikan informasi didalam gambar. cara yang ada untuk menyembunyikan informasi dalam gambar digital dengan pendekatan yang umum termasuk :

- penyisipan least significant bit
- masking dan filtering, dan
- algoritma dan transformasi.

Setiap teknik-teknik itu dapat diaplikasikan dengan derajat kesuksesan yang bervariasi pada file gambar yang berbeda.

1. Penyisipan *Least Significant Bit*

Penyisipan *Least Significant Bit* (LSB) adalah umum, pendekatan yang sederhana untuk menempelkan informasi di dalam suatu file cover. Sayangnya, hal itu sangat peka untuk kejadian yang melalaikan manipulasi gambar. Meng-konvert suatu gambar dari format GIF atau BMP, yang merekonstruksi pesan yang sama dengan aslinya (*lossless compression*) ke JPEG yang *lossy compression*, dan ketika dilakukan kembali akan menghancurkan informasi yang tersembunyi dalam LSB.

2. **Gambar 24-bit**

Untuk menyembunyikan suatu gambar dalam LSB pada setiap byte dari gambar 24-bit, dapat disimpan 3 byte dalam setiap pixel. Gambar 1,024 x 768 mempunyai potensi untuk disembunyikan seluruhnya dari 2,359,296 bit (294,912 byte) pada informasi. Jika pesan tersebut dikompres untuk disembunyikan sebelum ditempelkan, dapat menyembunyikan sejumlah besar dari informasi. Pada pandangan mata manusia, hasil *stego-image* akan terlihat sama dengan gambar cover.

Untuk contoh huruf A dapat disembunyikan dalam tiga pixel (asumsikan tidak ada kompresi). *Raster* data asli untuk 3 pixel (9 byte) menjadi

(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)

Nilai biner untuk A adalah 10000011. Sisipan nilai biner untuk A dalam tiga pixel akan menghasilkan

(00100111 11101000 11001000)
(00100110 11001000 11101000)
(11001000 00100111 11101001)

Bit-bit yang digaris bawahi hanya tiga perubahan secara aktual dalam 8 byte yang digunakan. Secara rata-rata, LSB membutuhkan hanya setengah bit dalam suatu perubahan gambar

Gambar 8-bit

Gambar 8-bit tidak diberikan untuk manipulasi LSB karena keterbatasan warnanya. Gambar cover harus lebih hati-hati diseleksi sehingga *stego-image* tidak akan mem-*broadcast* keberadaannya pada pesan yang ditempelkan. Ketika informasi disisipkan ke dalam LSB dari *raster data*, penunjuk kemasukan warna dalam palette yang diubah. Dalam suatu contoh, suatu palette sederhana empat warna dari putih, merah, biru dan hijau mempunyai posisi masukan palette yang sesuai secara berturut-turut dari 0 (00), 1 (01), 2 (10), dan 3 (11). Nilai *raster* dari empat pixel yang bersebelahan dari putih, putih, biru dan biru adalah 00 00 10 10. Penyembunyian nilai biner 1010 untuk perubahan bilangan 10 *raster data* ke 01 00 11 10, adalah merah, putih, hijau dan biru.

Implementasi LSB

Software steganografi memproses penyisipan LSB dengan membuat informasi yang tersembunyi dapat ditemukan lebih sedikit. Untuk contoh, tool EzStego menyusun palette untuk mengurangi kejadian dari warna indek bersebelahan yang kontrasnya paling banyak sebelum disisipkan pesan. S-Tool, merupakan tool steganography lainnya, yang mengambil pendekatan berbeda dengan memperkirakan cara lekat gambar cover, yang dapat berarti perubahan palette secara radikal. Seperti dengan gambar 24-bit, perubahan LSB pixel dapat membuat warna baru (Warna baru tidak dapat ditambahkan ke gambar 8-bit dalam kaitannya dengan keterbatasan palette). Sebagai gantinya, S-Tool mengurangi jumlah dari warna yang menangani kualitas gambar, sehingga perubahan LSB tidak secara drastis merubah nilai warna.

Sebagai contoh, nilai 8 warna diperlukan untuk setiap warna jika nilai 000 sampai 111 disimpan. Pengurangan jumlah warna yang unik ke 32 *ensures* bahwa nilai ini dapat digunakan dan jumlah dari warna tidak akan melebihi 256 (256/8 = 32). Setiap dari 32 warna yang unik dalam palette dapat diperluas ke 8 warna yang mempunyai nilai LSB dari merah, hijau, biru (RGB) dari 000 ke 111. Hasil warna multiple dalam palette yang terlihat sama visualisasinya tetapi itu dapat bervariasi dengan satu bit.

Tool ini mendapatkan pendekatan yang sama dengan gambar gray-scale. Bagaimanapun, hasil *stego-image* seperti yang diaplikasikan dengan S-Tool tidak lagi gray-scale. Sebagai gantinya hanya dengan warna yang bersebelahan seperti yang dilakukan EzStego. S-Tool memanipulasi palette untuk menghasilkan warna

yang berbeda satu bit. Untuk contoh, dalam gambar gray-scale yang normal, putih akan berpindah ke hitam dengan triple RGB berikut

(255 255 255), (254 254 254),…,
(1 1 1), (0 0 0)

Setelah diproses dengan S-Tool, nilai untuk putih akan tersebar atas range dari atas ke delapan warna sebagai

(255 255 255), (255 255 254), dan (255 254 255)

Visualisasi dari *stego-image* dapat dilihat sama seperti gambar cover gray-scale, tetapi aktualnya menjadi suatu gambar 8-bit warna.

3. Masking dan Filtering

Teknik *masking* dan *filtering*, hanya terbatas ke gambar 24-bit dan gray-scale, informasi disembunyikan dengan menandai suatu gambar dengan cara seperti *paper watermark*. Teknik *watermarking* dapat di aplikasikan dengan resiko rusaknya gambar dalam kaitannya dengan *lossy compression,* sebab mereka lebih menyatu ke dalam gambar.

Watermark digital dapat berupa informasi sebagai copyright, kepemilikan, atau lisensi, seperti yang ditunjukkan dalam Gambar 3. Dalam steganography, objek dari komunikasi adalah pesan yang tersembunyi. Di dalam watermark digital, objek dari komunikasi adalah cover.

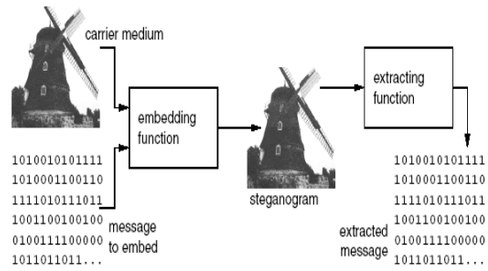
Gambar 3. Gambar yang di Watermarking
(Sumber: Exploring Steganography: Seeing the Unseen)

Untuk membuat gambar watermark dalam Gambar 3. dengan meningkatkan *luminance* dari area *masked* 15%. Jika diubah *luminance* dengan persentasi yang lebih kecil, mask akan tidak terdeteksi oleh mata manusia. Sekarang kita dapat menggunakan gambar watermark untuk menyembunyikan plaintext atau informasi yang di-*encode*-kan.

4. PENYERANGAN SISTEM STEGANOGRAFI

System steganografi ditunjukkan pada Gambar 4. Pengirim membuat suatu steganogram menggunakan fungsi *embedding,* dimana fungsi tersebut mempunyai dua parameter sebagai berikut :

1. Media pembawa yang isisnyabersifat random
2. Pesan yang embedded.



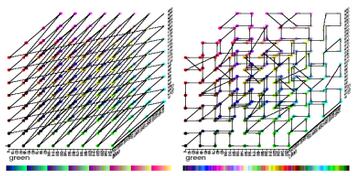
Gambar 4. Sistem Steganografi
(Sumber : Attacks on Steganographic Systems)

Beberapa utilitas steganografi menggunakan kunci rahasia. Ada dua jenis kunci rahasia : kunci steganografi dan kunci kriptografi [4]. Kunci steganografi mengontrol proses embedding dan ekstracting. Tanpa kunci itu, bagian ini tidak diketahui, dan masing-masing sample yang digunakan untuk mendeteksi penempelan oleh penyerangan statistik adalah dengan pencampuran tempat yang digunakan dan tempat yang tidak digunakan, yang dapat merusak hasilnya. Kunci kriptografi digunakan untuk mengenkripsi pesan sebelum ditempelkan. Kedua aplikasi rahasia ini diperoleh dari algoritma aktual dalam bentuk suatu parameter kunci. Untuk tidak memasangkan keamanan algoritma steganografi dari penampilan pesan yang tersembunyi, menggunakan *pseudo random bit-strings* untuk menghasilkan pesan-pesan itu. Seperti mempunyai semua properti statistik dari pesan yang dienkripsi. Penyerangan statistik diaplikasikan pada Jsteg menggunakan model statistik yang berbeda.

5. EzStego

Utilitas EzStego (oleh Romana Machado) melekatkan pesan dalam file-file GIF. File-file GIF berisi lukisan warna sampai dengan 256 warna yang berbeda dari 2^{24} warna yang mungkin, dan *Lempel Ziv Welch* (LZW) dimampatkan dalam matrik [3,6,8] dari indek lukisan. EzStego melekatkan pesan ke dalam pixel tanpa informasi yang lengkap. Hal ini meninggalkan lukisan warna tanpa dimodifikasi. Algoritma steganografi menciptakan suatu penyortiran salinan lukisan itu. Dengan cara itu akan sulit untuk membedaan antara dua warna bersebelahan di dalam lukisan yang disortir itu. Penyortiran secara *luminance* adalah tidak optimal, setidak-tidaknya sebab dua

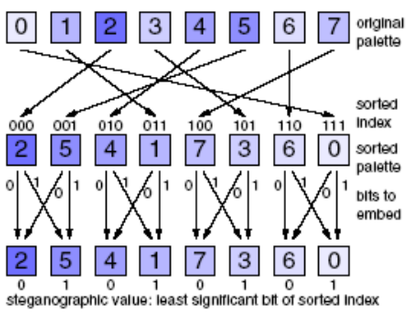
orang mewarnai dengan seri yang sama dapat berbeda secara radikal. Kita dapat menginterpretasikan masing-masing warna sebagai titik di dalam suatu ruang tiga dimensi, yakni dalam kubus warna RGB (merah, hijau, biru).



Gambar 5. Order warna dalam suatu lukisan (kiri) dan penggunaan penyortiran oleh EzStego (kanan)

Gambar 5 menunjukkan order warna-warna dalam kubus RGB. Pada sisi kiri warna-warna kelihatan lebih terurut daripada sebelah kanan. Order warna pada lukisan ini banyak dalam kasus order secara numerik. Pada sisi kanan, warna telah disortir oleh EzStego mengikuti suatu jalur yang paling pendek pada kubus RGB.

Fungsi *embedding* EzStego bekerja garis per garis pada pixel berurutan yang tidak terputus-putus dari kiri atas sampai kanan bawah. Setelah *embedding*, masing-masing pixel memegang satu nilai steganografi (misalnya 1 bit dari pesan yang dilekatkan). Nilai steganografi dari suatu pixel adalah indek *least significant bit* dalam lukisan yang telah disortir. Fungsi embedding memenuhi nilai steganografi dengan bit yang dilekatkan (jika bit yang dilekatkan tidak ada) dan mengganti warnanya dengan sebelahnya dalam lukisan yang telah disortir, jika diperlukan.

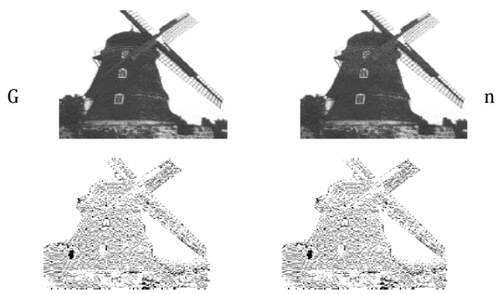


Gambar 6. Fungsi Embedding EzStego

Gambar 6, menunjukkan fungsi embedding EzStego dengan lukisan yang sudah direduksi. Sebagai contoh, ditemukan indek 7 untuk sebuah pixel yang ditentukan dalam image pembawa. Jika ingin melekatkan a'1' digantikan oleh indek 3, dan jika ingin melekatkan a'0' tidak perlu merubah apapun. Sebab indek warna 7 dalam lukisan original adalah indek 101 (=5) dalam lukisan yang telah disortir. Kedua warna itu bersebelahan, karena itu sangat sulit membedakannya. Sebuah perubahan dari indek 7 ke indek 3 (dan sebaliknya) tidak dapat dilihat dengan mata biasa, kecuali jika dibandingkan secara langsung dengan gambar aslinya.

6. Serangan Visual (*Visual Attacks*)

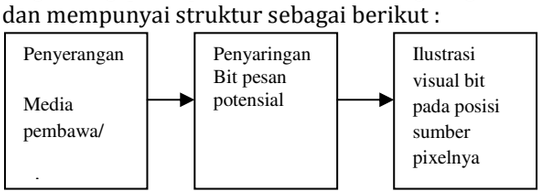
Mayoritas algoritma steganografi melekatkan pesan-pesan pengganti secara hati-hati bit yang diseleksi dengan bit pesan. Secara nyata, ini adalah sulit untuk memberi ciri yang acak dan muatan image oleh mesin, dan lebih sulit lagi mencirikan *least significant bit* dan bit random. Itu adalah sangat sulit untuk menetapkan muatan image dengan cara formal. Bagaimanapun, pembatasan menjadi kabur dan pada imajinasi masing-masing. Penglihatan manusia terlatih untuk mengenali berbagai hal. Kemampuan ini digunakan untuk serangan visual. Gambar 3.5 merepresentasikan *least significant bit* untuk gambar 3.4 yang mana menunjukkan bahwa benar-benar bukan merupakan serangan pada steganografi. Kita masih dapat melihat LSB kedua image, dan kita tidak dapat mengidentifikasi steganogram dengan mata kita, meskipun separuh bagian atas sisi kanan berisi pesan steganografi.



Gambar 7. Least significant bit dari Gambar 3.4, hitam untuk LSB=0, putih untuk LSB=1

Ide Serangan Visual

Gagasan serangan visual adalah untuk memindahkan semua bagian-bagian yang mencakup image pesan. Mata manusia sekarang dapat menciri apakah ada pesan potensial atau hanya muatan image. Proses penyaringan tergantung pada perkiraan utilitas steganografi, dan mempunyai struktur sebagai berikut :



Gambar 8. Arus Penyerangan Visual

Menempelkan Saringan untuk Serangan Visual

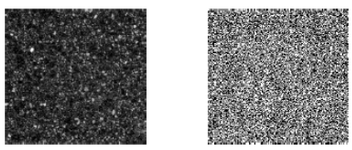
Menempelkan saringan untuk serangan visual secara grafik ditunjukkan field nilai pixel ketika fungsi penyaringan diaplikasikan kepadanya. EzStego menggunakan warna pixel, yang digambarkan oleh lukisan, untuk menentukan bit yang dilekatkan. Penempelan saringan untuk serangan visual pada EzStego menggantikan lukisan asli oleh lukisan hitam dan putih. Ini ditunjukkan pada Gambar 10

Gambar 9. Penugasan fungsi warna pengganti; warna yang mempunyai indek dalam lukisan yang disortir menjadi hitam, sisanya menjadi putih.

Eksperimen

Contoh-contoh berikut dengan jelas menunjukkan suatu asumsi menjadi nyata bahwa LSB adalah sepenuhnya acak dan oleh karena itu mungkin dapat digantikan.

EzStego; penyaringan image dari Gambar7 : tanpa embedded (kiri), 50% kapasitas dari pembawa digunakan untuk embedding



Gambar 10. Image GIF dari hiasan lantai media pembawa,dan image penyaringannya

EzStego – penempelan yang berlanjut

Pesan-pesan yang tidak menggunakan panjang maksimum yang dimungkinkan, meninggalkan sisa dari media pembawa tanpa perubahan. EzStego tidak mengenkripsi muatan pesan. Untuk memudahkan mengenali dimana letak pesan disembunyikan ditunjukkan pada Gambar 10, tetapi itu tergantung pada muatan image, seperti ditunjukkan pada Gambar 11.

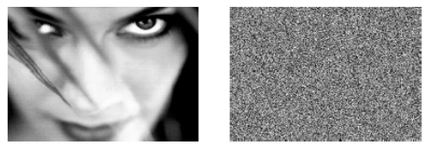
S-Tools - penempelan yang menyebar

S-Tools menyebar suatu pesan di atas keseluruhan media pembawa. Berlawanan dengan EzStego, tidak ada garis pembatas yang jelas antara sisa yang tidak dirubah, dengan menunda pesan yang lebih pendek, dan perubahan pixel secara steganografi. Keduanya bercampur. Pada image sebelah kanan Gambar 12, Gambar 13, Gambar 14 ada delapan warna, satu bit pada setiap tiga komponen warna, sebab S-Tools melekatkan sampai tiga bit tiap pixel.

Steganos – penempelan berlanjut dengan isian atas

Steganos menggunakan media pembawa dengan sepenuhnya dalam tiap-tiap kasus. Hal itu akan memenuhi pesan yang lebih pendek, seperti ditunjukkan pada Gambar 3.13. Penyaringan steganogram tidak pernah berisi muatan awal image.

Gambar 11. Image warna asli BMP sebagai media pembawa, dan image penyaringannya.



Gambar 12. S-Tools; steganogram dengan ukuran maksimum dari teks embedded, dan image penyaringannya

Jsteg – penempelan pada domain yang diubah

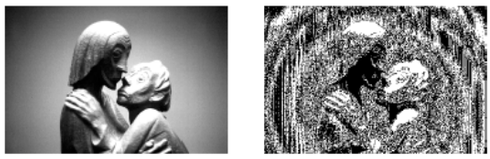
Jsteg melekat dalam image JPEG. Dalam image JPEG, image berisi perubahan dalam koefisien frekuensi untuk mencapai penyimpanan yang sama. Tidak ada seranangan visual pada pengertian yang diperkenalkan, sebab satu bit yang steganografi mempengaruhi sampai 256 pixel.

Serangan Statistik (*Statistical Attacks*)

1. Ide Serangan Chi-square

Fungsi embedding EzStego banyak menyisipkan indek penyortiran LSB. Penulisan berlebihan LSB mengubah nilai masing-masing yang hanya berbeda LSBnya. Pasangan-pasangan nilai ini disebut PoV dalam sambungan. Jika penulisan LSB distribusinya sama, frekuensi kedua nilai dari masing-masing PoV menjadi sama. Gambar 15, menggunakan contoh Gambar 6. untuk menjelaskan bagaimana frekuensi warna-warna gambar berubah, ketika digunakan untuk melekatkan distribusi pesan yang sama. Ide penyerangan statistic ini untuk membandingkan secara teori distribusi frekuensi harapan dalam stegnogram dengan beberapa distribusi sample dalam media pembawa yang mungkin berubah.

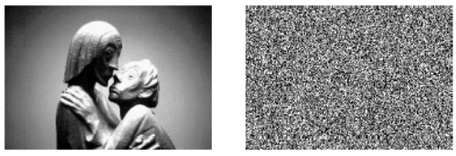
Gambar 13. S-Tools; steganogram dengan kapasitas 50% digunakan media pembawa, dan image penyaringan



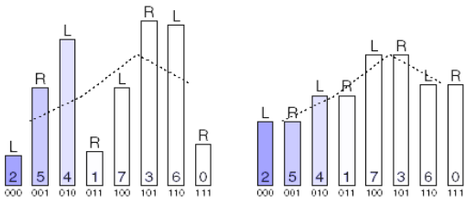
Gambar 14. Image warna asli BMP sebagai media pembawa, dan image penyaringan

Suatu titik kritis adalah bagaimana cara memperoleh secara teoritis distribusi frekuensi harapan (frekuensi kejadian yang kita harapkan setelah penerapan perubahan steganografi. Frekuensi ini harus tidak diperoleh dari sample acak, sample acak ini mungkin sudah diubah oleh operasi yang steganografi. Tetapi dalam banyak kasus kita tidak mempunyai yang asli untuk membandingkan dengan atau untuk memperoleh frekuensi harapan. Orijinalnya, secara teori frekuensi harapan adalah rata-rata perhitungan dua frekuensi dalam PoV. Garis yang dihancurkan pada Gambar 3.14 menghubungkan nilai rata-rata perhitungan. Sebab fungsi embedding menyisipkan kembali LSB, hal ini tidak merubah jumlah dua frekuensinya. Pengambilan jumlah nilai frekuensi ganjil ditranfer secara korespondensi ke nilai frekuensi genap dalam masing-masing PoV, dan sebaliknya. Seperti penjumlahan tetap, rata-rata perhitungan adalah sama untuk suatu PoV dalam kedua-duanya, media pembawa yang asli dan masing-masing korespondensi steganogram.

Derajat kesamaan dari distribusi sample yang diamati dan secara teori distribusi frekuensi harapan adalah pengukuran probabilitas dari beberapa embedding yang sedang berlangsung. Derajat kesamaan ini menentukan penggunakan uji Chi-square. Uji ini beroperai pada kategori pemetaan observasi.



Gambar 15. Steganos; steganogram hanya dengan satu byte teks embedded, dan image penyaringan



Gambar 16. Histogram warna sebelum dan sesudah embedding pesan dengan EzStego

Dengan langkah-langkah sebagai berikut :

2. Misalkan ada k kategori dan sample acak pengamatan. Masing-masing pengamatan harus masuk dalam satu dan hanya satu kategori. Kategori itu adalah semua indek lukisan, warna yang mana ditempatkan pada indek dalam lukisan yang telah disortir. Tanpa membatasi generalisasi, konsentrasikan pada nilai ganjil PoV dari penyerangan media pembawa. Secara teori minimum frekuensi harapan harus lebih dari 4, kita boleh menggabung kategori-kategori untuk menjaga kondisi ini.
3. Secara teori frekuensi harapan dalam kategori *i* setelah embedding distribusi pesan yang sama adalah

$$n_i^* = \frac{|\{\text{colour}|\text{sortedIndexOf}(\text{colour}) \in \{2i, 2i+1\}\}|}{2}$$

4. Pengukuran kejadian dalam sample acak

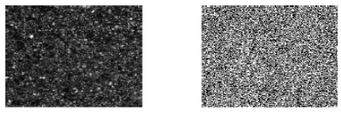
$$n_i = |\{\text{colour}|\text{sortedIndexOf}(\text{colour}) = 2i\}|$$

5. Statistik χ^2 ditentukan dengan derajat kebebasan $k-1$

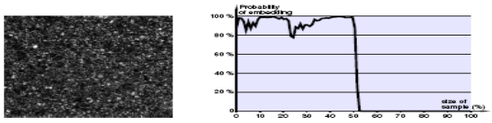
$$\chi^2_{k-1} = \sum_{i=1}^{k} \frac{(n_i - n_i^*)^2}{n_i^*}$$

6. p adalah probabilitas statistik dengan syarat $n_i = n_i^*$

$$p = 1 - \frac{1}{2^{\frac{k-1}{2}}\Gamma\left(\frac{k-1}{2}\right)} \int_0^{\chi^2_{k-1}} e^{-\frac{x}{2}} x^{\frac{k-1}{2}-1} dx$$



Gambar 17. Hiasan lantai sebagai steganogram EzStego, dan penyaringan; penyerangan visual tidak dapat mencirikan antara setengah steganografi dan setengah asli



Gambar 18. Probabilitas embedding dengan EzStego dalam image hiasan lantai.

Eksperimen

EzStego – Embedding berlanjut.

Gambar 3.15 melukiskan suatu steganogram, dimana suatu pesan rahasia 3600 byte telah dilekatkan, pesan yang sama pada Gambar 3.4, Gambar 3.15 kelihatan lebih cantik seperti Gambar 3.8, dalam kaitannya dengan muatan gambar. Serangan fisual dapat menjangkau batasnya. Diagram pada gambar 3.16 menunjukkan nilai p dari uji Chi-square sebagai

fungsi penambahan sampel. Nilai *p* ini adalah probabilitas embedding. Inisialisasinya, sampel dikompres 1% dari pixel, dimulai dari batasan bagian atas. Untuk contoh ini, Persamaan (1) menghasilkan probabilitas embedding, *p* = 0,8826. Sampel berikutnya dikompres dengan penambahan 1% dari pixel., yakni 2% dari gambar yang utuh. Nilai *p* bertambah menjadi 0,9808. Selama kompres sampel pixel separuh bagian atas, yang telah dilekatkan, nilai p tidak jatuh di bawah 0,77. Pixel gambar yang separuh lebih rendah tanpa perubahan, sebab pesan yang dilekatkan bukanlah seperti itu. Sampel 52% dari pixel meliputi tanpa perubahan pixel yang cukup untuk menentukan nilai p yang utama jatuh sampai 0. (Disini, yang utama berarti probabilitasnya lebih kecil dari presisi numerik 80 bit perhitungan floating point digunakan untuk implementasinya).

S-Tools – penempelan yang menyebar.

S-Tools menyebarkan bit-bit di atas keseluruhan media pembawa. Oleh karena itu, tipe diagram Gambar 3.16 tidak digunakan untuk S-Tools, Tabel 1 mengenali efektifitas uji statistik dengan menerapkan pada beberapa file yang tanpa penempelan, 50% penempelan, atau 99,5% penempelan, secara berturut-turut. Kenyataannya uji sederhana ini terlalu lemah untuk mendeteksi perubahan penyebaran. Uji yang lebih sensitif mengambil kombinasi yang sesuai dari k kategori atau kategori yang berbeda. Beberapa eksperimen menunjukkan hasil yang bermanfaat dengan hanya 33% dari teks yang dilekatkan dalam image warna, tetapi uji untuk penempelan teks yang lebih sedikit dengan ε sampai kerapatan 0,5.

Steganos – penempelan berlanjut dengan isian atas.

Tabel 2 memberikan hasil eksperimen yang sama pada steganos. Jika penempelan hanya satu byte dengan steganos, diperoleh probabilitas kesalahan kecil yang sama seperti jika menggunakan kapasitas 100% media pembawa. Ini berkaitan dengan fakta bahwa arus kode digunakan untuk enkripsi pesan rahasia memenuhi pesan dengan lapisan byte sampai kapasitas media pembawa penuh.

Jsteg – penempelan dalam domain yang berubah.

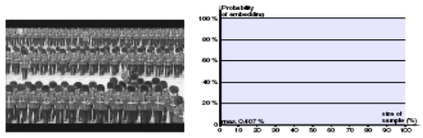
Penyerangan visual tidak bekerja pada Jsteg. Karena Jsteg (seperti EzStego) melekatkan bit secara terus-menerus, seperti pada Gambar 3.16, Gambar 17, Gambar 18, dan Gambar 19. Ini menunjukkan bahwa uji statistik sangat efektif mengenai Jsteg.

PENUTUP

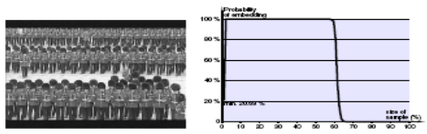
Simpulan

Strategi embedding sistem-stego dengan menyisipkan LSB pada media pembawa, seperti yang telah diuraikan di atas dapat diambil kesimpulan sebagai berikut.

1. Serangan visual yang telah diuraikan menunjukkan bahwa LSB dalam gambar-gambar tidak sepenuhnya acak, tetapi berkorelasi satu dengan yang lain. Secara jelas dapat dibedakan jika gambar-gambar itu dipresentasikan menggunakan filter penempelan untuk serangan visual yang diuraikan di atas.



Gambar 19. Image JPEG sebagai media pembawa; tanpa sesuatu yang menempel, dan uji statistik yang menghasilkan probabilitas penempelan yang sangat rendah

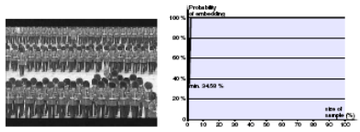


Gambar 20. Jsteg; steganogram dengan penempelan 50%

2. Penulisan kembali frekuensi LSB yang sama tentang kejadian yang berbeda dengan probabilitas yang sangat tinggi. Melalui uji statistik kesamaan dengan jelas dapat dideteksi.

Dari eksperimen di atas, uji statistik lebih baik daripada serangan visual, karena lebih sedikit ketergantungan pada cover yang digunakan dan dapat secara penuh diotomatiskan, dengan demikian dapat diaplikasikan pada skala yang besar.

Dengan tidak menyisipkan semua LSB, tetapi hanya suatu bagiannya dan dengan memilih bit-bit ini secara acak, tingkat kesalahan kedua serangan visual dan statistik meningkat. Tetapi dengan pengukuran itu, throughput sistem steganografi berkurang.



Gambar 21. Jsteg; steganogram dengan ukuran penempelan teks maksimum

DAFTAR PUSTAKA

1. FAP Petitcolas, RJ Anderson, MG Kuhn, "*Attacks on Copyright Marking Systems*", http://www.cl.cam.ac.uk/~fapp2/papers/ih98-attacks/ di akses 13 Juli 2012
2. Neil F. Johnson, Sushil Jajodia, 1998, "*Steganalysis of Images Created Using Current Steganography Software"*, in David Aucsmith (Ed.): Information Hiding, LNCS 1525, Springer-Verlag Berlin Heidelberg. pp. 32–47
3. M. R. Nelson: "*LZW Data Compression. Dr. Dobb's Journal*", October 1989.
4. Birgit Pfitzmann, 1996. "*Information Hiding Terminology*", in Ross Anderson (Ed.): Information Hiding. First International Workshop, LNCS 1174, Springer-Verlag Berlin Heidelberg pp. 347–350
5. Robert Tinsley, 1996 "*Steganography and JPEG Compression*", Final Year Project Report, University of Warwick,
6. Andreas Westfeld, Andreas Pfitzmann, "*Attacks on Steganographic Systems*"**,** Dresden University of Technology, Department of Computer Science, D-01062 Dresden, Germany, http://wwwrn.inf.tudresden.de/~westfeld/attacks.html