EKSPLORASI STEGANOGRAFI : KAKAS DAN METODE

Meliza T.M.Silalahi

Program Studi Teknik Informatika Institut Teknologi Bandung
Ganesha 10, Bandung
if16116@students.if.itb.ac.id

ABSTRAK

Steganografi merupakan ilmu dan seni menyembunyikan informasi. Seiring dengan perkembangan teknologi digital, steganografi pun mengalami perkembangan. Namun, perkembangan teknologi serta daya komputasi jugalah yang menyebabkan keamanan menjadi aspek yang tidak bisa dikesampingkan begitu saja. Terbukti dengan banyaknya serangan yang memanfaatkan kelemahan pada perangkat lunak yang mengimplementasikan metode steganografi. Dibutuhkan metode yang kuat, aman dan cocok untuk menyembunyikan informasi, meskipun pada kenyataannya terdapat keuntungan dan kerugian tersendiri atas keberhasilan tersembunyinya informasi maupun terbongkarnya suatu informasi.

Makalah ini khusus membahas steganografi dengan gambar sebagai media penyembunyi informasi. Akan dilakukan eksplorasi beberapa metode yang ada dalam steganografi, kakas yang menggunakan metode tersebut serta serangan-serangan atas celah kelemahan kakas steganografi tersebut.

Kata kunci : Steganografi, metode, kakas, serangan

1. PENDAHULUAN

Kriptografi sudah menjadi salah satu cara pengamanan informasi. Informasi disamarkan (dienkripsi) dengan metode,algoritma dan kunci tertentu sehingga menghasilkan pesan samar yang tidak dimengerti maknanya. Hal ini dapat menimbulkan kecurigaan bagi pihak lawan. Terlebih lagi bila pihak lawan berhasil mendekripsi pesan samar tersebut menjadi informasi asli. Informasi menjadi tidak aman dan tidak rahasia lagi. Diperlukan suatu cara agar pihak lain tidak curiga akan keberadaan informasi . Atas dasar inilah lahir suatu ilmu dan seni penyembunyian informasi yang bernama steganografi.

Steganografi berasal dari bahasa Yunani, yaitu "*steganos*" yang artinya "tulisan tersembunyi *(covered writing*)". Dalam implemetasinya, steganografi membutuhkan dua properti, yaitu media penampung dan pesan rahasia. Media penampung yang umum digunakan adalah gambar, suara, video, atau teks. Pesan yang disembunyikan dapat berupa sebuah artikel, gambar, daftar barang, kode program atau pesan lain[1].

Untuk itu, steganografi memiliki setidaknya tiga bagian, yaitu :

a. *Hiddentext* : pesan yang disembunyikan
b. *Covertext* atau *cover-object* : pesan atau objek yang digunakan untuk menyembunyikan pesan rahasia.
c. *Stegotext* atau *stego-object* : pesan atau objek yang sudah berisi pesan rahasia.

Terdapat tiga aspek yang perlu diperhatikan dalam menyembunyikan pesan: kapasitas, keamanan dan ketahanan/kekuatan. Kapasitas merujuk kepada besarnya informasi yang dapat disembunyikan oleh media penutup, keamanan merujuk kepada ketidakmampuan pihak lain untuk mendeteksi keberadaan informasi yang disembunyikan, dan ketahanan/kekuatan merujuk kepada sejauh mana medium steganografi dapat bertahan sebelum pihak lain menghancurkan informasi yang disembunyikan[2]

1. METODE

Ada berbagai metode / algoritma yang digunakan untuk menyembunyikan informasi dalam gambar *(cover image)*. Di antaranya adalah sebagai berikut [3] :

a. Least Significant Bit (LSB)

Metode ini memodifikasi langsung nilai byte dari *cover image*. Bit-bit pesan digunakan untuk

mengganti bit-bit kurang berarti (*least significant bit*) dari *cover image*.

Misalkan, *cover image* C berukuran 1.024 x 768. Tiap byte dapat menyimpan 3 bit di setiap pixelnya. Jadi *cover image* C memiliki potensi menyembunyikan 2,359,296 bits (294,912 bytes) informasi. Bila pesan dikompress dan disembunyikan sebelum digabungkan,informasi yang disembunyikan akan lebih banyak. Dengan mata, hasil *stego-image* akan akan terlihat identik dengan *cover image*.

Lebih rincinya sebagai berikut :
Huruf A dapat disimpan dalam 3 pixel (asumsi tidak ada kompres)
Data asli untuk 3 pixel (9 bytes) menjadi :
(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)

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

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

Bit yang bergaris bawah adalah tiga yang berubah dalam 8 byte yang digunakan. Pada rata-rata LSB membutuhkan hanya sebagian dari bit dalam gambar yang dapat diubah. Data disembunyikan dalam bit terakhir maupun kedua terakhir bit yang signifikan dan tetap saja mata manusia tidak akan dapat membedakannya.

b. Masking dan filtering
Teknik ini terbatas pada gambar 24-bit dan keabu-abuan (*gray scale*). Informasi disembunyikan dengan memberi tanda pada gambar (*marking*) . Masking lebih kuat bila dibandingkan dengan LSB, apalagi bila mengalami kompresi, *cropping* dan beberapa proses gambar *(image processing)* lainnya. Teknik ini menyatukan informasi pada area berarti (*significant area*)sehingga pesan tersembunyi lebih menyatu dengan *cover image*, tidak hanya sekedar menyembunyikannya pada level *'noise'*.

c. Algoritma dan Transformasi
Manipulasi dengan menggunakan LSB memang mudah untuk menyembunyikan informasi, namun sangat rentan apabila mengalami proses gambar *(image processing)*. Algoritma dan transformasi yang digunakan berbeda antara satu kakas dengan kakas yang lain. Salah satunya adalah transformasi DCT *(discrete cosine transform)*. DCT biasanya digunakan untuk gambar yang memiliki tipe JPEG.

3. Eksplorasi Kakas / perangkat lunak dan metode

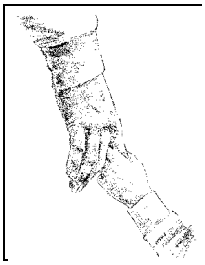
Berikut akan dibahas satu per satu kakas yang mengimplementasikan berbagai metode steganografi :

3.1 Steganography 1.8

Gambar 1 Steganography 1.8

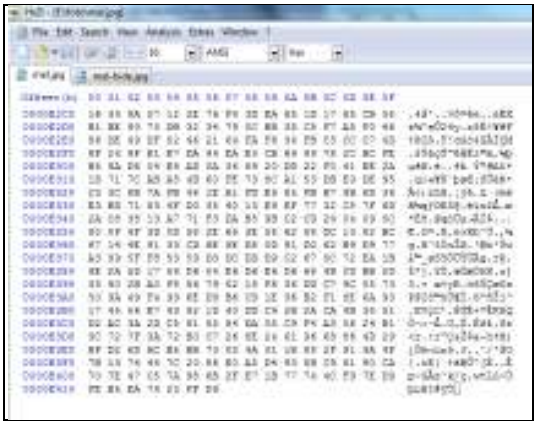
Pencobaan ini menggunakan kakas steganography 1.50, *cover image*-nya adalah "mel.jpg" dan pesan yang disembunyikan dapat berupa teks ataupun file. Dalam hal ini pesan tersembunyi berupa file "holding-hands".

Gambar 2 Cover Image

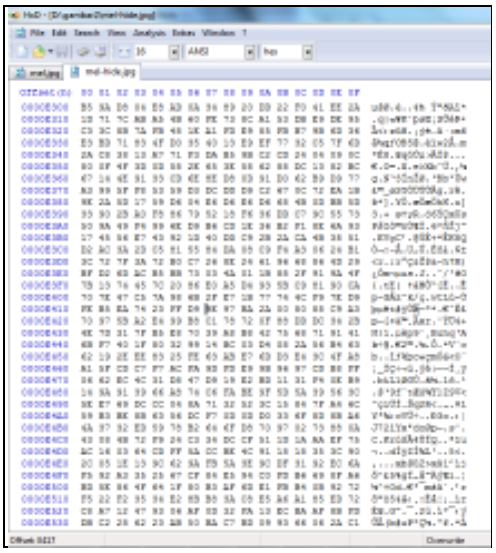


Gambar 3 Informasi Tersembunyi

Bila dilihat dengan kasat mata, tampak tidak ada perbedaan antara *cover image* dengan gambar yang telah disisipkan informasi rahasia. Sekarang bandingkan bit-bit heksa kedua gambar tersebut dengan bantuan kakas Hex Editor Neo.



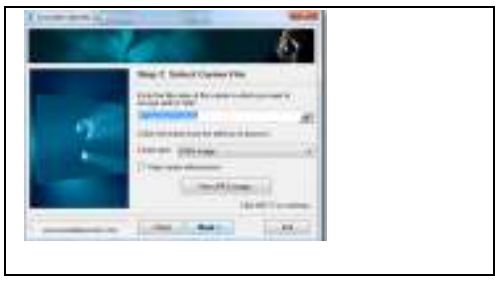
Gambar 4 Heksa Cover Image



Gambar 5 Heksa Stegano-Object

Akhir file *cover image* ditandai dengan offset FF D9 , baris 0000E410. Pada gambar yang telah disisipkan pesan terlihat adanya penambahan bit-bit setelah akhir file *cover image*. Metode ini dinamakan Fuse. Mudah untuk diimplementasikan, namun keamanannya sangat minim. (lebih jelasnya akan dibahas pada serangan).

3.2 Invisible Secrets



Gambar 6 Langkah - langkah Invisible Secrets 2002

Bila dilihat dengan kasat mata, kedua gambar ini tidak berbeda. Metode yang digunakan pada kakas ini adalah LSB.

Gambar 7 Cover Image

Gambar 8 Stegano-Object dengan Invisible Secrets 2002

Invisible Secrets 2002 memiliki beberapa keunggulan, diantaranya:

- Menggunakan algoritma kriptografi simetri yang kuat (contohnya : Blowfish, Twofish, RC4, AES, dan lain-lain).
- Mengkompresi data terlebih dahulu sebelum disembunyikan. Sebuah langkah yang

sederhana dan penting untuk mengurangi ukuran dan redudansi data yang tersembunyi.

- Memungkinkan untuk menyembunyikan file random untuk meningkatkan '*noise*'.

Namun, sama saja halnya dengan Fuse (pada kakas sebelumnya), keamanannya masih dipertanyakan. Hal ini dibuktikan dengan mudahnya file yang telah disisipkan pesan tersembunyi dideteksi. Baik dengan menggunakan analisai matematis maupun dengan menggunakan kakas lain yang dirancang untuk mendeteksi keberadaan pesan tersembunyi. Hal-hal terkait serangan akan dibahas kemudian.

4. Serangan

4.1 Serangan terhadap Steganography 1.8

Tabel 1 Tanpa Sandi Lewat

70 7e 47 c5 7a 98 6b 2f e7 1b 77 74 4c f9 7e d9fe b5 ea 74 23 ff d9 9e 97 ba 2a 00 80 88 c9 a370 97 5b a2 e4 99 b8 c1 78 72 0f 88 dd dc 34 2b4e 7d 31 7f b5 e8 70 39 a8 b8 42 75 68 71 91...40 00 58 01 00 00 6c 3c 39 6c 30 6b 6c 31 30 6e 38 38 6a 3a 38 3c 00 cd 12 00

Tabel 2 Sandi Lewat 'a'

70 7e 47 c5 7a 98 6b 2f e7 1b 77 74 4c f9 7e d9fe b5 ea 74 23 ff d9 9e 97 ba 2a 00 80 88 c9 a370 97 5b a2 e4 99 b8 c1 78 72 0f 88 dd dc 34 2b4e 7d 31 7f b5 e8 70 39 a8 b8 42 75 68 71 91...40 00 58 01 00 00 38 6b 6b 39 3f 3d 6a 31 6b 38 6e 39 6a 3e 69 30 00 cd 12 00

Tabel 3 Sandi Lewat 'rahasia'

70 7e 47 c5 7a 98 6b 2f e7 1b 77 74 4c f9 7e d9fe b5 ea 74 23 ff d9 9e 97 ba 2a 00 80 88 c9 a370 97 5b a2 e4 99 b8 c1 78 72 0f 88 dd dc 34 2b4e 7d 31 7f b5 e8 70 39 a8 b8 42 75 68 71 91...40 00 c8 00 00 00 69 6b 3c 3b 3f 3a 3c 6e 39 3e 6d 31 3a 3c 39 6c 00 cd 12 00

Bit berwarna hitam adalah *cover image*, bit berwarna biru adalah pesan tersembunyi dan bit berwarna merah adalah sandi lewat.Bila diperhatikan dari strukturnya dapatlah dengan mudah bahwa ada file tersembunyi.

Pihak lain dapat dengan mudah mengetahui keberadaan pesan tersembunyi cukup dengan melihat strukutrnya. Oleh karena ini, metode fuse tergolong sangat lemah dan mudah dideteksi. Dari kakas steganograhy ini sendiri juga masih minim akan kriptografinya yang hanya menggunakan hash dari 16 byte (128bit). Bila kunci kriptografinya ditemukan, pesan tersembunyi dapat dibaca dengan program yang dapat mengubah file berformat heksa ke file aslinya. Pada eksplorasi ini, karena keterbatasan waktu, penulis belum dapat merancang dan mengimplementasikan program yang dimaksud.

4.2 Serangan terhadap Invisible Secrets 2002

Terdapat beberapa kelemahan pada Invisible Secrets 2002, di antaranya adalah sebagai berikut :

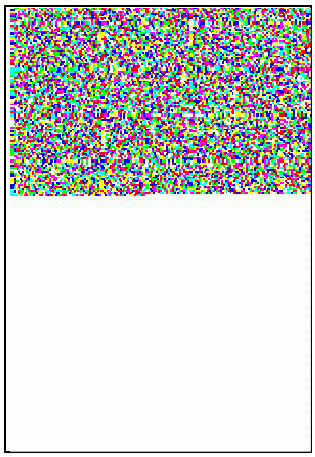
- Steganografi PNG dan JPG sangat buruk. Data / informasi yang akan disembunyikan diletakkan pada *comment field* dari tipe gambarnya. Untuk JPG diletakkan di awal sedangkan PNG diletakkan di akhir. Sebagai perbandingan, BMP dan WAV menggunakan 1-bit LSB dan HTML menggunakan spasi atau tab (0/1) yang ditambahkan di akhir baris.
- Informasi yang disembunyikan,baik yang dienkripsi terlebih dahulu, tetap saja masih memiliki struktur linear dan tetap (meskipun bermula dari akhir baris gambar). Hal ini menyebabkan sangat mudah mengetahui adanya pesan tersembunyi bila menggunakan kakas ini.
- Pada gambar berformat BMP, LSB yang tidak digunakan untuk menyembunyikan data akan diset menjadi 1 atau 0. Seharusnya tidaklah perlu melakukan hal ini karena semakin mengundang kecurgiaan. Sama halnya seperti mengatakan, "Hai, lihatlah ke sini!".

Sebagai contoh :

Gambar 9 Gambar Asli

Gambar 10 LSB gambar asli

Gambar 11 LSB dan pesan tersembunyi. LSB yang tidak digunakan diset menjadi 0 sehingga berwarna hitam

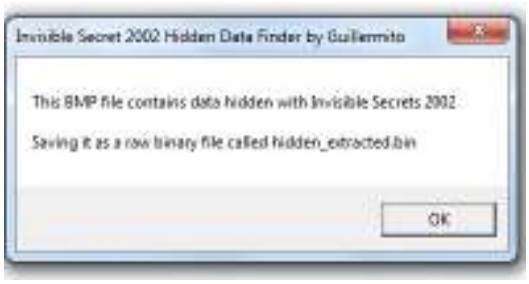


Gambar 12 LSB dan pesan tersembunyi. LSB yang tidak digunakan diset menjadi 1 sehingga bewarna putih

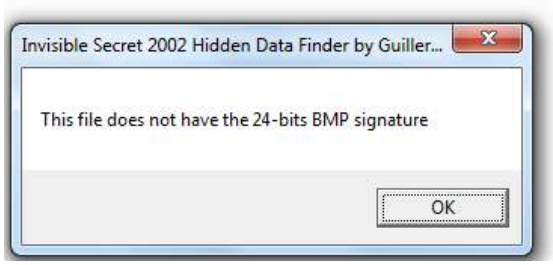
Guillermito[4] berhasil mengemukakan gagasan untuk memecahkan kakas steganografi ini. Idenya adalah sebagai berikut :

- Pergi ke baris terakhir
- Ekstrak semua LSB dalam suatu urutan
- Ketika sampai pada akhir baris, pergi ke baris sebelumnya dan mulai lagi.
- Setelah data yang telah diekstrak terkumpul semuanya, cek ukuran header dan dapatkan ukuran selanjutnya.
- Lakukan terus pada semua blok data sampai menemukan bit-bit yang identik. Bila sudah menemukannya, kemungkinan ada sesuatu yang disembunyikan oleh kakas ini.

Guillermito berhasil mengimplementasikannya dalam bahasa assembly. Bila program dijalankan, semua gambar yang mengandung steganografi dari kakas ini akan terdeteksi.



Gambar 13 Uji Coba Gambar yang Mengandung Steganografi Invisible Secrets 2002



Gambar 14 Uji Coba Gambar yang Tidak Mengandung Steganografi Invisible Secrets 2002

5 Kesimpulan

Berdasarkan hasil eksplorasi metode dan kakas steganografi, dapat disimpulkan sebagai berikut :

1. Dalam merancang dan mengimplementasikan metode steganografi pada suatu kakas, perlu mempertimbangkan aspek keamanan (dalam hal ini adalah kriptografi). Hal ini disebabkan oleh minimnya keamanan kakas steganografi yang telah dieksplorasi. Perlu mengingat apabila pesan tersembunyi telah diketahui keberadaannya, pesan tersebut tidak dapat langsung diketahui (memerlukan dekripsi terlebih dahulu).
2. Terdapat dua metode dan dua kakas dalam eksplorasi makalah ini, yaitu metode Fuse pada Steganografi 1.8 dan metode LSB pada Invisible Secrets 2002. Apabila dibandingkan terdapat kelemahan dan keunggulan masing-masing seperti yang telah dijelaskan pada pem bahasan.
3. Untuk memecahkan steganografi (steganalisis) perlu mengetahui terlebih dahulu metode apa yang dipakai. Lalu, analisa dan rancang program untuk mendeteksi keberadaan informasi yang disembunyikan.

4. Jangan terlalu percaya pada kakas steganografi yang mengatakan bahwa kakasnya terbukti aman, tidak dapat dideteksi, dan lain sebagainya. Tidak, sampai Anda benar-benar tidak berhasil mencoba memecahkannya. Lebih baik menggunakan algoritma atau program yang dirancang sendiri. Tentu saja hal ini bergantung kepada seberapa penting informasi yang disembunyikan tersebut.

DAFTAR PUSTAKA

[1] Munir, Rinaldi, "Kriptografi", Institut Teknologi Bandung, 2009.

[2] Provos Niels, Honeyman Peter," Hide and Seek: An Introduction to Steganography",IEEE SECURITY &PRIVACY, May/June 2003.

[3] Neil F. Johnson, Sushil Jajodia,"Exploring Steganography: Seeing the Unseen", IEEE Computer, Feb 1998, pp 26-34.

[4] Analyzing steganography softwares http://www.guillermito2.net/stegano/ Tanggal Akses : 24 Maret 2010