Steganografi pada Enkripsi Image dengan Menggunakan Least Significant Bit Insertion

Ronny – NIM : 13506092

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if16092@students.if.itb.ac.id

Abstrak

Steganografi merupakan suatu metode untuk menyimpan suatu informasi yang penting, dan biasanya menggunakan digital media sebagai medianya. Pada tulisan ini, meida utama yang akan dibahas adalah image. Berbeda dengan kriptografi, steganografi dapat disimpan dengan sedemikian rupa sehingga orang awam atau orang yang sedang melakukan penyerangan terhadap file tidak akan mengetahui bahwa file tersebut berisi pesan rahasia. Ada berbagai cara untuk menyimpan suatu data rahasia dengan steganografi. Semakin lama, cara-cara tersebut berkembang dan semakin canggih pula. Terkadang juga steganografi digabungkan dengan kriptografi agar pesan yang sudah terenkripsi tersebut akan sulit terlihat dan tidak akan mudah diduga. Salah satu cara untuk melakukan stegaografi adalah menggunakan *least significant bit insertion*. Cara ini sangat efektif, karena dengan mengganti bit-bit yang tidak signifikan, image yang menjadi tempat penyimpanan data tidak akan mengalamin banyak perubahan, bahkan bisa dibilang hampir sama dengan image aslinya.Tulisan ini akan membahas mengenai penggunaan steganografi pada digital media image, dibandingkan dengan kriptografi yang hanya melakukan enkripsi pada plain teks sehingga orang awam sekalipun dapat mengetahui bahwa file tersebut telah didekripsi.Batasan masalah di sini adalah bagaimana proses steganografi dengan *least significant byte* ini dilakukan, dan juga algoritma yang diperlukannya. Selain itu akan dibahas juga bagaimana proses mengekstrak gambar tersebut dilakukan agar pesan yang dikirimkan dapat dibaca oleh penerima. Terakir adalah mengenai hasil eksperimen yang diperlukan dan juga kesimpulannya.

***Kata kunci**: Steganografi, least significant byte,enkripsi,dekripsi*

1. Pendahuluan

File-file berbentuk gambar selalu memiliki perpindahan melalui berbagai macam jaringan. Dengan besarnya perkembangan jaringan komputer dan kemajuan yang tinggi di dunia digital, jumlah file digital yang dipertukarkan pun semakin tinggi. Terkadang sebagian besar data yang dipertukarkan tersebut bersifat privat, rahasia, yang pada akhirnya menghasilkan tinginya kebutuhan akan teknik enkripsi yang kuat. Enkripsi dan steganografi adalah teknik yang sering digunakan untuk melakukan perlindungan pada data-data tersebut. Namun, pada kenyataannya tidak ada satu jenis algoritma enkripsi pun yang bisa digunakan dengan baik pada setiap jenis data yang berbeda. Pada makalah ini, akan dibahas mengenai penggunaan steganografi untuk menyembunyikan informasi pada proses enkripsi. Teknik steganografi secara cepat menjadi sangat maju dan digunakan oleh hampir setiap pengguna. Steganografi merupakan 'pelengkap' yang tepat untuk proses enkripsi karena memungkinkan user untuk melakukan penyimpanan data dalam jumlah yang besar dalam sebuah gambar. Selain itu, user juga bisa melakukan penggabungan teknik ini dengan kriptografi sehingga file yang akan dikirim terlindung jauh lebih aman. Caranya adalah dengan melakukan enkripsi terlebih dahulu, kemudian menyembunyikan hasil enkripsinya. Dengan demikian, kriptanalis harus menemukan informasinya lebih dahulu sebelum kemudian melakukan dekripsi. Kekurangan pada enkripsi biasa adalah file hasil enkripsi tersebut terlalu jelas, sehingga orang yang secara kebetulan melihat file tersebut dapat dengan mudah menyimpulkan bahwa file tersebut telah tidak boleh dilihat oleh sembarang orang. Sementara itu, adanya steganografi akan membuat proteksi tambahan pada file tersebut. Adanya informasi tersembunyi dengan metode *Least Significant Bit*(LSB) akan membuat orang yang melihat

tidak menyadari bahwa ada pesan tersembunyi di dalam tulisan dan dia tidak akan berusaha memecahkannya. Pada teknik steganografi ini, sejumlah blok horizontal dan vertikal akan dibentuk, kemudian mencampurkanya dengan file image sebelum ditransfer ke receiver. Dengan cara ini, tabel transformasi tidak perlu dikirimkan, dan seluruh informasi yang akan dikirmkan bisa disebarkan secara random pada LSB di setiap *byte* berdasarkan kunci rahasia sebelum transmisi data. Cara ini juga akan mengurangi kemungkinan image tersebut untuk dideteksi dan meningkatkan level keamanan. Selain itu, penerima juga akan memungkinkan user untuk membangun tabel transformasi (melalui beberapa blok horizontal dan vertikal tersebut) setelah berhasil mengekstrak data yang tersembunyi itu. Image yang asli akan bisa diproduksi ulang dan proses dekripsinya pun bisa dilakukan.

2. Latar belakang skema steganografi

Steganografi merupakan teknik untuk menyembunyikan data pada media digital dan juga alternatif untuk mengamankan data selain kriptografi.keduanya biasa digabungkan untuk membuat para penyerang kesulitan dan frustasi. Karena itu, teknik emkripsi yang dapat diandalkan merupakan elemen yang sangat penting utuk mencegah pemalsuan, penyerangan dari akses yang ilegal. Meskipun begitu, ada 2 perbedaan mendasar antara kriptografi dan steganografi. Kriptografi biasanya mengubah huruf-huruf dan melakukan pengacakan, sedangkan steganografi membuat seolah-oleh pesan tersebut tidak ada dengan menyelipkannya pada pesan lain. Ada banyak metode steganografi yang telah diciptakan selama beberapa tahun terakhir. Beberapa contoh dari steganografi dapat dilihat pada

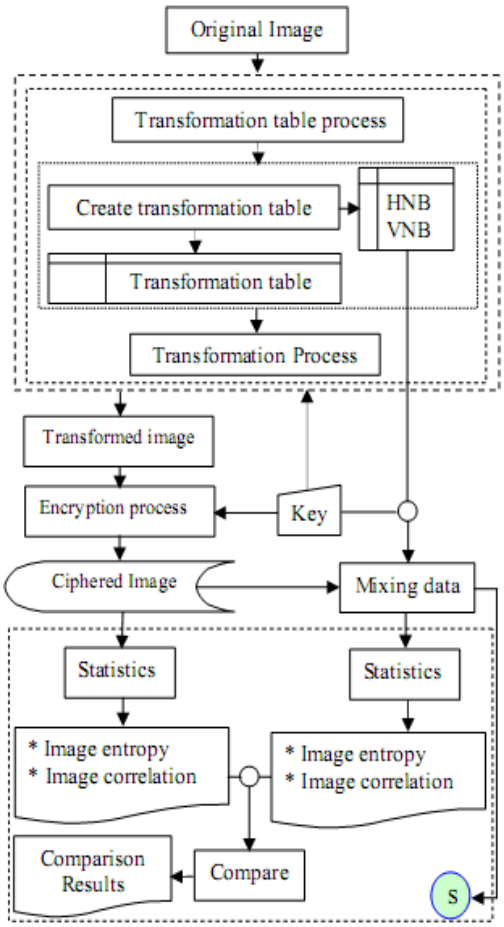
- tinta yang tak terlihat
- microdot
- digital signature
- komunikasi spektrum tersebar Johnson dan Sushil, yang menjelaskan bagaimana steganografi diaplikasikan pada proses penyimpanan informasi di gambar dan melakukan survey pada beberapa software steganografi untuk memproses gambar
- Tsung Yuan dan Wen-Hsiang, yang berhasil merancang algoritma untuk menyembunyikan data pada microsoft word.
- Sinha dan Singh juga merencanakan sebuah teknik enkripsi dengan menggunakan digital signature.
- Kemudian ada Kisik et al. Yang merencanakan sebuah algoritma steganografi yang menempelkan (embeed) pesan rahasia pada gambar bitmap dan gambar berbasis pallete. Algoritma racangannya adalah membagi gambar bitmap ke gambar *bit-plane* dari LSB-plane ke MSB-plane untuk setiap pixel, dengan asumsi bahwa bitmap plane sebagai biner. Deskripsi dari algoritma rancangannya akan dijelaskan dibagian selanjutnya, di mana algoritma yang pertama akan digunakan untuk mencampur data sebelum transmisi dilakukan(sender), dan yang kedua adalah saha untuk mengekstrak informasi dari gambar yang telah dienkripsi tersebut(receiver) sebelum dekripsinya dilakukan dan gambar aslinya diperoleh.

3. Teknik yang digunakan

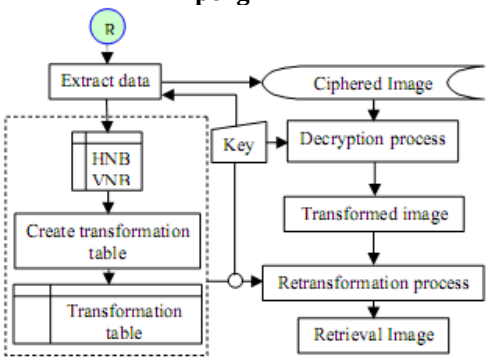
Kita menerapkan metode ini untuk bertukar informasi antara pengirim dan target penerima. Metode ini diterapkan untuk menyembunyikan informasi di dalam encrypted gambar lalu mengirim encrypted gambar kepada penerima sebenarnya untuk mengambil informasi yang tersembunyi untuk digunakan untuk menghasilkan ulang tabel transformasi rahasia yang sama, dan lalu memungkinkan penerima untuk mengembalikan gambar yang asli dengan tepat.

Metode ini digunakan sebagai proses enkripsi tambahan untuk mencegah akses yang tidak terautorisasi dideteksi. Tujuan yang lain adalah perincian mengenai bagian bagian dari proses steganografi; kunci rahasia, encrypted gambar menggambarkan gambar yang sudah melalui transformasi setelah enkripsi dan jumlah blok horizontal dan vertikal menggambarkan informasi tersembunyi.

Hasilnya adalah data campur; encrypted gambar yang termasuk data campur. Dalam proses ini, kita akan menggunakan tujuan akhir ini untuk investigasi keefektifan metode yang diajukan dalam encrypted gambar dengan dan tanpa metode ini. Gambar.berikut menerangkan model keseluruhan dari teknik yang diajukan dari pengirim dan penerima secara berurutan.



Overview metode yang digunakan pada sisi pengirim



Overview metode yang digunakan pada sisi penerima

3.1 Data mixing (sender)

Informasi pertama yang disiapkan oleh pengirim (jumlah blok horizontal dan vertical) dicampur dengan ciphered gambar(gambar yang akan dicampur) sebelum pengiriman. Posisi pemasukan dalam informasi ini akan diseleksi secara acak tergantung dari kuncinya. Ini membuat pemakai yang tahu akan kunci dapat memperoleh informasi. Algoritma berikut ini menjelaskan bagaimana informasi akan dicampur dengan ciphered gambar.

Algoritma :

1. Buka gambar chipered.
2. Masukkan kata kunci
3. Ubah blok horizontal ke bentuk bit array.(HB_bitArray)
4. Ubah blok vertical ke bentuk bit array (VB_bitArray).
5. Tentukan buff sebagai fungsi hash dari kunci.
6. Lakukan fungsi random dengan buff di atas.
7. /*mengganti bit terkecil dari salah satu pixel random dengan bit ke-I dari HB_bitArray*/
   For i = 0 to panjang HB_bitArray – 1
   r = rnd // r antara 1 dan jumlah pixel gambar -1
   bit terkecil dari pixel r = HB_bitArray(i)
8. /*mengganti bit terkecil dari salah satu pixel random dengan bit ke-I dari VB_bitArray*/
   For i = 0 to panjang VB_bitArray – 1
   r = rnd // r antara 1 dan jumlah pixel gambar -1
   bit terkecil dari pixel r = VB_bitArray(i)

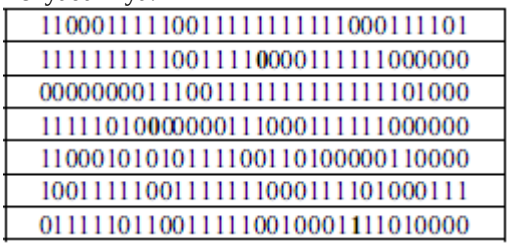
   END MIXING_DATA
   Input: Ciphered gambar, kunci
   Output: Mixed data

sebagai contoh : misalnya stream biner pada gambar 3 adalah bagian dari gambar yang akan digabung(28 pixel). Sedangkan jumlah dari HB dan VB berturut-turut adalah 150 dan 100. Di mana representasi dari angka 150 dan 100 ini (**001**0010110 & 0001100100) akan disembunyikan di gambar juga. Setiap bit tersebut kemudianakan menggantikan nilai LSB terkecil pada pixel. Sedangkan pixel itu sendiri akan dpilih secara random berdasarkan kata kunci.

110001111100111111111111000111101
111111111100111110001111111000000
000000001110011111111111111101000
111110101000001110001111111000000
110001010101111001101000001 10000
100111110011111110001111010001 11
011111011001111100100011 11010000

Sebelum dicampur

Gambar 4 menunjukkan beberapa bagian dari file gambar dengan LSB pada pixel nomor 6,17, dan 27 diganti dengan 3 bit pertama dari stream data yang merepresentasikan jumlah blok. Amati bahwa pixel 6 dan 17 telah berubah nilainya, sedangkan pixel nomor 27 tidak berubah karena nilai lama dan barunya sama. Penggantian pada pada bit ini tidak akan disadari oleh mata manusia, sehingga tidak akan ada yang menyadarinya.



Setelah dicampur

3.1 Data extracting (receiver)

Di sisi lain, sebelum dekripsi dari gambar chipered dilakukan, data yang dicampurkan tersebut harus diekstrak terlebih dahulu dari gambarnya. Jumlah blok horizontal dan vertical dari gambar yang telah dicampur(mix) tersebut juga akan diekstrak dulu. User yang mangetahui key akan mampu melakukan ekstrasi tersebut, sebab posisi memasukkan tersebut bergantung pada kunci rahasia tersebut. Berikut ini adalah algoritmanya:

1. Buka data yang tercampur
2. Masukkan kata kunci
3. Buff = fungsi hash dari kata kunci tersebut.
4. Lakukan fungsi random dengan buff di atas.
5. For i = 0 to panjang HB_bitArray – 1
   r = rnd // r antara 1 dan jumlah pixel gambar -1, lakukan ekstrasi untuk mengambil nilai bit ke-i pada HB_bitArray dari LSB di pixel gambar ke-r
   HB_bitArray(i) = bit terkecil dari pixel r.
6. Konversi HB_bitArray ke HB
7. For i = 0 to panjang VB_bitArray – 1
   r = rnd // r antara 1 dan jumlah pixel gambar -1, lakukan ekstrasi untuk mengambil nilai bit ke-i pada VB_bitArray dari LSB di pixel gambar ke-r
   VB_bitArray(i) = bit terkecil dari pixel r.
8. Konversi VB_bitArray ke VB
   END SPLITTING _DATA
   Input: Mixed data, Kunci
   Output: Ciphered gambar, HB, VB

Pada dasaranya mix dan ekstraksi ini menggunakan prinsip yang sama.

4. Hasil Eksperimen

Metode digunakan untuk mengevalusi teknik yang diajukan seperti dijelaskan dalam gambar. 1. Algooritma digunakan dalam bit mapped (bmp) dengan ukuran 300 pixel x 300 pixel dengan 256 colors. Untuk evaluasi efek dari proses insertion dari encrypted images, 3 kasus di tes. Jumlah blok dan code binary dari setiap kasus terlihat di Tabel berikut.

Cases	Key	HB	VB	Binary representation	
				HNB	VNB
Image 1	8 *byte*	30	30	00000 11110	00000 11110
Image 2	16 *byte*	60	60	00001 11100	00001 11100
Image 3	32 *byte*	100	100	00010 01001	00010 01001

4.1 Data mixing dan data extracting

Jumlah blok horizontal dan vertical adalah 1024. Karenanya, jumlah bits yang akan dikirim pada file yang dienkripsi adalah 20 bit, 10 bit untuk blok horizontal dan 10 bit untuk blok vertical. 20 bit ini akan dimasukkan secara random berdasarkan kunci rahasia yang telah dibangun seperti dijelaskan sebelumnya. Table korelasi dan entropi sebelum dan sesudah proses mixing bisa dilihat pada table berikut ini:

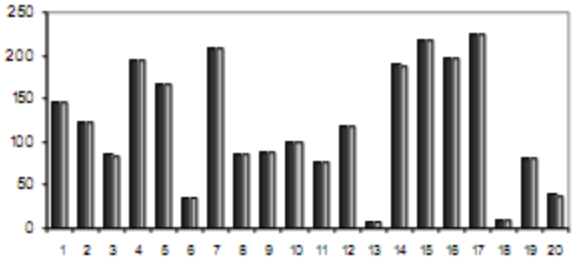
Encrypted Images		*Number of Blocks*	*Correlation*	*Entropy*
Image1	Before insertion	30×30	0.0566	5.2261
	After insertion		0.0566	5.2261
Image2	Before insertion	60×60	0.0032	5.5415
	After insertion		0.0032	5.5415
Image3	Before insertion	100×100	0.0018	5.5417
	After insertion		0.0018	5.5417

Kasus 1 : image 1

Pada kasus ini, kita menggunakan kunci 8 byte, dan NB = 30, VB = 30. Hasil dari proses ini dapat dilihat pada table da gambar berikut ini :

Insertion positions	*Pixels color before insertion*	*Pixels color after insertion*
72229	146	146
6556	122	122
76040	85	84
64569	195	194
64917	166	166
25438	35	35
51057	209	209
67030	85	85
27500	87	87
70299	100	100
51146	76	76
45253	119	118
89864	6	6
4606	189	188
86528	218	218
86696	197	197
85264	225	225
41719	9	9
17749	81	81
7848	39	38

Hasil dari posisi insertion pada kasus 1



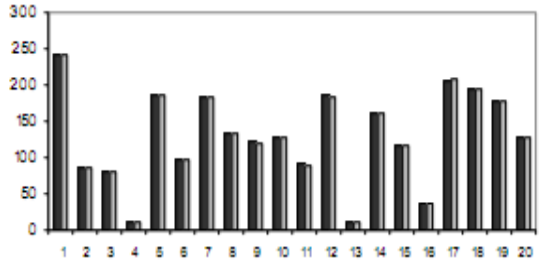
Pixel sebelum(gelap) insertion dan sesudah(terang) insertion pada kasus 1

Tabel dan gambar tersebut menunjukkan bahwa ada 15 pixel (ditandai) yang tidak mengalami perubahan warna setelah insertion.

Kasus 2 : image 2
Pada kasus ini, kita menggunakan kunci 16 byte, dan NB = 60, VB = 60. Hasil dari proses ini dapat dilihat pada tabel dan gambar berikut ini :

Insertion positions	*Pixels color before insertion*	*Pixels color after insertion*
36968	243	242
30458	86	86
13729	81	80
32070	11	10
44455	185	185
69455	96	97
4862	182	183
26790	132	133
88444	121	120
55548	128	128
74907	91	90
24186	185	184
36160	11	10
55186	160	160
30138	117	117
48566	35	35
45306	206	207
73384	194	195
41132	179	178
21318	128	128

Hasil dari posisi insertion pada kasus 2



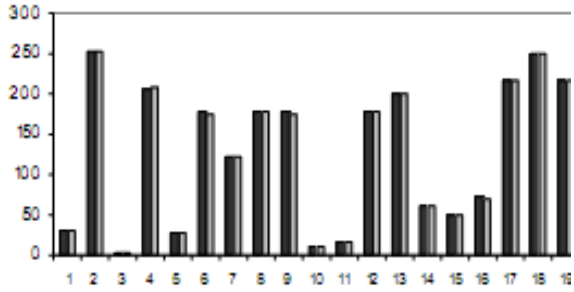
Pixel sebelum(gelap) insertion dan sesudah(terang) insertion pada kasus 2

Tabel dan gambar tersebut menunjukkan bahwa ada 7 pixel (ditandai) yang tidak mengalami perubahan warna setelah insertion. Proses ini menunjukkan bahwa proses insersi masih kuat dan mampu untuk mengunci data dalam image tersebut.

Kasus 3 : image 3
Pada kasus ini, kita menggunakan kunci 8 byte, dan NB = 90, VB = 90. Hasil dari proses ini dapat dilihat pada tabel dan gambar berikut ini :

Insertion positions	*Pixels color before insertion*	*Pixels color after insertion*
61642	30	30
31753	254	254
48268	3	2
10057	206	207
85067	28	29
85779	177	176
75791	122	122
11179	177	177
33118	177	176
49869	11	10
61289	17	16
83889	179	178
19885	200	200
37384	61	61
7797	49	49
89697	71	70
38955	217	216
73182	250	251
3873	218	218
83564	202	202

Hasil dari posisi insertion pada kasus 3



Pixel sebelum(gelap) insertion dan sesudah(terang) insertion pada kasus 3

Tabel dan gambar tersebut menunjukkan bahwa ada 9 pixel (ditandai) yang tidak mengalami perubahan warna setelah insertion.

5. Kesimpulan

Metode steganografi dirancang untuk menggabungkan/menempelkan pesan ke dalam sebuah image secara random berdasarkan kunci rahasia sebelum transmisi data. Karenanya, informasi ini akan muncul seperti file image biasa dan bisa digunakan oleh penerima untuk membangun tabel transformasi dan mengembalikan gambar aslinya.

Hasil eksperimen menunjukkan bahwa korelasi dan entropi dari gambar sebelum dan sesudah melakukan data mixing hampir sama, sehingga file gambarnya pun tidak akan terlihat berubah dan tidak akan diketahui oleh mata manusia. Inilah yang menjadi faktor yang mampu menigkatkan level keamanan dari steganografi.

DAFTAR PUSTAKA

H. Kathryn, "A Java Steganography Tool," http://diit.sourceforge.net/files/Proposal.pdf

N.F. Johnson, J. Suhil, " Exploring Steganography:Seeing the Unseen," Computing practices, 2006, 2006, http://www.jjtc.com/pub/r2026.pdf

M.M. Amin, M. Salleh, S. Ibrahim, M.R Katmin, M.Z.I. Shamsuddin, "Information hiding using steganography,"Telecommunication Technology, 2003.