ISSN : 1693 – 1173

Pengamanan Data Teks dengan Kriptografi dan Steganografi

Wawan Laksito YS [5)]

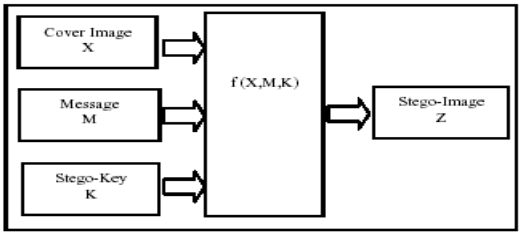
Abstrak

Keamanan data teks ini sangatlah penting untuk menghindari manipulasi data yang tidak diinginkan seperti pengeditan, pembacaan, akses dan publikasi ilegal. Teknik Pengamanan dengan melakukan pengacakan data awal dengan metoda enkripsi dilanjutkan dengan menyisipkan hasil enkripsi kedalam gambar (image). Penyisipan teks ke dalam gambar dilakukan dengan mengganti sebagian bit derajat keabuan gambar dengan bit data teks. Ekstrating data dilakukan dengan mengekstrak dari gambar dan dilakukan deskrepsi sehingga dimunculkan data asli.

I. Pendahuluan

Kemudahan dan keunggulan yang dimiliki data digital menyebabkan semakin pesatnya laju penggunaan dan pemanfaatan data digital dalam berbagai bidang kehidupan. Namun di balik keunggulan itu tetap ada celah yang dimanfaatkan oleh sebagian manusia untuk dapat memanipulasinya. Manipulasi di sini meliputi modifikasi isi data, akses illegal dan publikasi atau penyebar luasan yang tidak sah.

Teknik steganografi menyembunyikan pesan ke dalam sebuah penampung atau medium, dalam kasus ini adalah *cover image*. Pesan rahasia disisipkan ke sebuah *cover image* yang menghasilkan sebuah *stego image*. Penambahan *stego key* digunakan untuk memperkuat pengamanan data pada pesan rahasia. Skema steganografi seperti terlihat pada gambar di bawah ini



[5)] Staf Pengajar STMIK Sinar Nusantara Surakarta

Proses penyisipan pesanmenggunakan metoda *Least Significant Bit (LSB)* yang dimodifikasi. LSB merupakan salah satu teknik yang paling umum digunakan dalam steganografi karena kesederhanaan dalam implementasinya dan teknik ini memberikan pengaruh terkecil pada cover image karena hanya mempengaruhi 1 atau 2 bit pada LSB. Walaupun demikian, membuat program semacam ini bukanlah suatu hal mudah, mengingat dan menimbang adanya banyak faktor teknis dan keamanan yang harus diperhatikan.

II. Tujuan

Membuat suatu sistem kriptografi dan steganografi yang dapat mengamankan data digital berupa teks dari manipulasi yang tidak diinginkan.

III. Pembahasan

Langkah-langkah penyandian data teks adalah sebagai berikut :

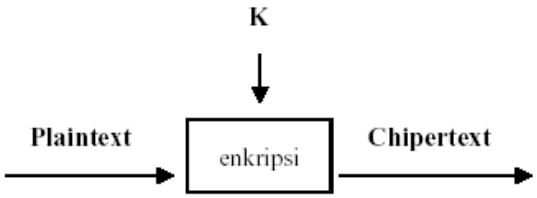
a. Pengacakan pesan (*encription*), sebelum disisipkan ke dalam media gambar dilakukan pengacakan pesan *plaintext* menghasilkan *chipertext.*
b. Penyisipan (*embedding*) *chipertext* hasil pengacakan pada media cover.

Sedangkan langkah-langkah *decoding* merupakan kebalikan dari langkah penyandian.

a. Penguraian (*extraction*) pesan dari media cover
b. Penyusunan pesan (*decreption*), setelah *chipertext* diekstrak dari media cover paka dilakukan kembali penyusunan menjadi data *plaintext.*

1.1 Encription

Dengan *cyptosystem*nya, data asli (*plaintext*) diacak menjadi data acak (*chipertext*) ataupun sebaliknya. Pada proses penyisipan pesan, sebelum disisipkan ke dalam media gambar, data teks di*encrypt* (diacak) terlebih dahulu agar data teks menjadi tidak terbaca. Pada penguraian pesan, setelah data teks diuraikan dari gambar maka *chipertext* di*decrypt* agar menjadi data asli(*plaintext*) sehingga data bisa dibaca kembali.



Metode yang digunakan dalam proses enkripsi adalah dengan cara melakukan XOR antara setiap bit plaintext dengan setiap bit kuncinya. XOR akan memiliki sifat bahwa keluaran akan sama dengan 1 bila salah satu masukan memiliki nilai 1. Operasi dilakukan per karakter.

Misalkan *plaintext* : **dengan nafasmu aku hidup**

Dan *key* : **Samson**

Maka operasi XOR akan dilakukan dengan susunan sebagai berikut:

d	e	n	g	a	n		n	a	f	a	s	n	u		a	k	U		h	i	d	u	p
S	a	m	s	o	n	S	a	m	s	o	n	S	a	m	s	o	N	S	a	m	s	o	n

plaintext* XOR *key* = *chipertext

Untuk masing-masing karakter operasinya adalah seperti berikut :

Plaintext : `01000100 01100101 01101110 01100111 .....`

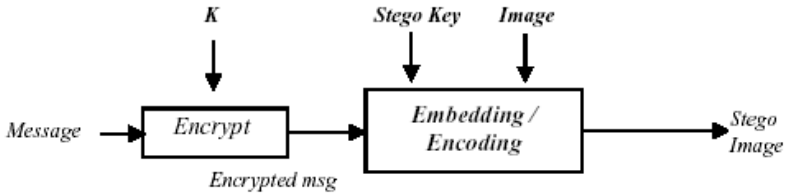
Key : `01010011 01100001 01101101 01110011 .....`

Chipertext : `00010111 00000100 00000011 00010100.…`

1.2 Embedding

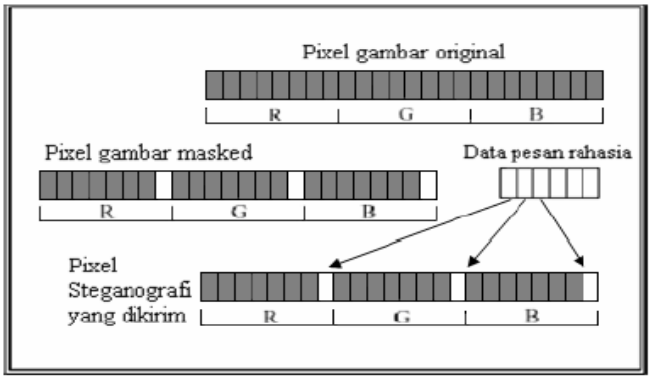
Steganografi merupakan teknik yang digunakan untuk menyisipkan data teks pada media gambar dan menguraikannya. Pada proses penyisipan pesan (*embedding text*), *chipertext* disisipkan dengan metode *LSB(Least Significant Bit)* sehingga menghasilkan gambar stego (gambar yang mengandung pesan tersembunyi). Gambar stego inilah yang merupakan hasil dari pengolahan data pada program aplikasi. Dengan gambar inilah diharapkan data teks menjadi aman. Sedangkan pada proses penguraian (*extraction*), pesan di*extract* dari media gambar agar menjadi bentuk *chipertext* yang kemudian akan dilakukan proses *decryption.*

Skema dari proses penyisipan atau *embedding* pesan ke dalam gambar adalah sebagai berikut :



Metode yang dipakai dalam proses penyisipan bit-bit pesan ke dalam bit-bit gambar adalah dengan menggunakan teknik LSB *Insertion* atau Penyisipan pada LSB. LSB (*Least Significant Bit*) adalah bit yang mempunyai nilai paling rendah, atau bit yang berada pada posisi paling kanan. Penyisipan LSB dilakukan dengan memodifikasi bit terakhir dalam satu byte data. Bit yang diganti adalah LSB karena perubahan pada LSB hanya menyebabkan perubahan nilai *byte* satu lebih tinggi atau satu lebih rendah. Misalkan data yang diubah adalah warna hijau, maka perubahan pada LSB hanya menyebabkan sedikit perubahan yang tidak dapat dideteksi oleh mata manusia. Data yang akan disisipkan adalah data teks atau karakter. Bit bit pesan disisipkan ke dalam LSB (*least significant bit*) gambar.

Proses penyembunyian pesan dilakukan dengan menyisipkan 1 bit pesan pada LSB (bit pertama) secara langsung untuk setiap posisi yang bersesuaian.



Seperti diketahui untuk file bitmap 24 bit maka setiap *pixel* (titik) pada gambar tersebut terdiri dari susunan tiga warna merah, hijau dan biru (RGB) yang masing-masing disusun oleh bilangan 8 bit (byte) dari 0 sampai 255 atau dengan format biner 00000000

sampai 11111111. Dengan demikian pada setiap *pixel* file bitmap 24 bit dapat disisipkan 3 bit data.
Contohnya huruf UB dapat kita sisipkan dalam 6 pixel.
Misalnya data gambar original adalah sebagai berikut:

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

Sedangkan representasi biner huruf UB adalah 01010101 01000010.
Dengan menyisipkan-nya pada data pixel diatas maka akan dihasilkan:

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

Untuk memperkuat teknik penyembunyian data, dapat ditambahkan beberapa metode yang merupakan modifikasi dari metode yang telah dijelaskan di atas.

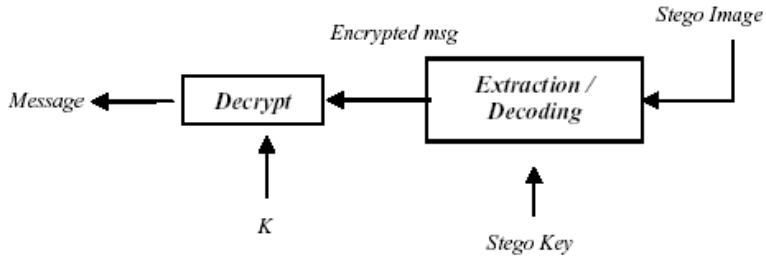
Modifikasi yang dilakukan antara lain :

- penggunaan pixel gambar yang tidak berurutan (*random pixel*),
- penyisipan pada LSB bit ke-1 dan LSB bit ke-2 secara bergantian dan acak (*random lsb*),
- mengubah seluruh *lsb* pada gambar (*modify all*).

1.3 Extraction

Proses penguraian (*extraction*), pesan di*extract* dari media gambar agar menjadi bentuk *chipertext* yang kemudian akan dilakukan proses *decryption.*

Skema dari proses penguraian atau *extraction* pesan ke dalam gambar adalah sebagai berikut :



Stego Image atau gambar yang telah berisi pesan rahasia dimasukkan dalam proses *decoding*. Pada saat *decoding* diperlukan *stego key* untuk melakukan otentikasi program. Setelah proses *encoding*, maka akan dihasilkan sebuah pesan yang terenkripsi sehingga harus dilakukan dekripsi untuk mendapatkan pesan yang dapat terbaca atau dimengerti..

Penguraian pesan dalam gambar berarti mengambil bit-bit karakter yang tersebar dalam pixel gambar. Prosedur untuk mendapatkan kembali bit-bit karakter dalam gambar adalah sebagai berikut :

a. Tentukan *f(x,y)*

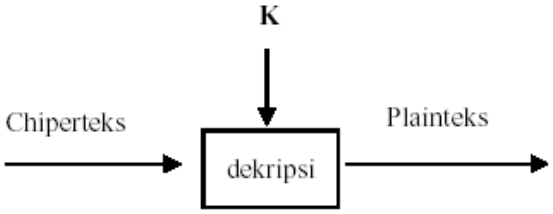
b. Untuk masing-masing komponen warna *R,G,B* dari *f(x,y)*, bangkitkan angka random, jika ganjil maka dapatkan *bit* terakhir atau LSB dari masing-masing komponen warna dengan *bit* = [*R,G,B*] *AND 1*, jika genap maka dapatkan *bit* terakhir atau LSB dari masing-masing warna dengan bit = [R,G,B] *AND 2*.

c. Susun kembali LSB pada setiap elemen *lsb* hingga menjadi pesan dengan masing-masing 8-bit untuk tiap karakter.

1.4 Decreption

Sedangkan proses dekripsi dilakukan untuk mengembalikan pesan ke pesan asal. Dekripsi merupakan kebalikan dari enkripsi.

Skema dekripsinya adalah sebagai berikut :



chiper XOR kunci = plaintext

Contoh :

Chipertext : `00010111 00000100 00000011 00010100 ……`

Key : `01010011 01100001 01101101 01110011 ……`

Plaintext : `01000100 01100101 01101110 01100111 ……`

IV. Kesimpulan

Teknik Kriptografi dapat mengacak data teks menjadi tidak terbaca. Dengan digabungkan aplikasi Steganografi yang sudah dibuat, mampu mengamankan data teks (*.txt) dengan cara menyembunyikannya dalam gambar dengan format bitmap (*.bmp) 24 bit menggunakan teknik LSB Insertion.

Agar citra image hasil penyisipan *plaintext* tidak terlalu berbeda pada pandangan kasat mata maka perlu dilakukan perhitungan/penentuan posisi dan jumlah bit yang akan diganti dengan memodifikasi teknik LSB.

Sebaiknya tidak dilakukan pengolahan citra pada gambar stego karena penyisipan menggunakan metode ini masih mempunyai kelemahan yaitu tidak tahan terhadap proses pengolahan citra seperti cropping, blur, masking, filtering dan proses pengolahan citra yang lain yang mengakibatkan hilang atau rusaknya data yang ada di dalam gambar.

Pustaka

1. Gonzalez, Rafael C. , Woods, Richard E.. **Digital Image Processing Second Editio**n, Prentice Hall. 2002
2. Johnson, Neil F. , **Steganography**. http://www.jjtc.com/stegdoc/SEC201.htm. Tanggal akses : 17-03-2006
3. Jorn Daub EDV, **File Formats Collection BMP**, http://www.daubnet.com/ formats/ BMP.html
4. Kurniawan, Yusuf. , **Kriptografi Keamanan Internet dan Jaringan Komunikasi**. Informatika Bandung. 2004
5. Wohlgemuth , Sven, **Steganography and Watermarking**, http://www.informatik.unifreiburg.de/~softech/teaching/ws01/itsec