Journal of Computer Networks, Architecture and High Performance Computing
Volume 1, Number 1, Januari 2019
https://doi.org/10.47709/cnapc.v1i1.45

Submitted : Jan 02, 2019
Accepted : Jan 16, 2019
Published : Jan 18, 2019

Analisa Metode Least Significant Bit Dalam Penyisipan Pesan Terenkripsi Terhadap Citra Digital

Penda Sudarto Hasugian

Teknik Informatika, STMIK Pelita Nusantara Medan, Jl. Iskandar Muda No.1 Medan

Email : penda.hasugian@gmail.com

Abstract- Berkembangnya jaringan komunikasi membuat pertukaran informasi semakin sering terjadi, baik dalam bentuk teks, audio, dan video. Semakin banyaknya pesan yang yang ingin dikirimkan maka keamanan dan kerahasiaan dari pesan tersebut semakin sulit dijaga. Keamanan dan kerahasiaan adalah aspek yang penting pada proses pertukaran informasi atau pesan. Untuk melindungi kerahasiaan pesan bisa menggunakan teknik kriptografi. Kriptografi adalah ilmu yang mempelajari teknik – teknik matematika yang berhubungan dengan aspek keamanan seperti kerahasiaan, integritas data, serta otentikasi. Sedangkan Steganografi adalah salah satu teknik yang dilakukan dalam mengamankan pesan yaitu dengan cara menyembunyikan pesan ke sebuah media digital. Pada penelitian ini dilakukan analisis untuk mengimplementasikan keamanan file txt dan rtf dengan kombinasi algoritma kriptografi One Time Pad dengan steganografi least significant bit (LSB). Pada tahap kriptografi pesan yang telah berhasil diinput akan dienkripsi dengan kunci yang telah diberikan kemudian dilanjutkan dengan proses penyisipan pesan di tahap steganografi.. Pesan yang dapat dibaca oleh sistem adalah pesan berformat .txt dan .rtf. Pada Algoritma LSB penyisipan dilakukan dengan menggantikan bit terakhir citra dengan bit pesan yang akan disisipkan. Pada tahap enkripsi dan penyisipan akan dihasilkan stego image, stego image merupakan citra yang sudah disisip oleh pesan. Kemudian pada tahap ekstraksi dan dekripsi yang dihasilkan adalah plaintext, plaintext merupakan pesan yang belum diberikan kunci apapun.

1. Pendahuluan

Seiring berkembangnya media internet dan aplikasi yang menggunakan internet semakin betambah pula kejahatan dalam sistem informasi. Caranya yaitu dengan menggunakan image, sound maupun video dalam menyembunyikan informasi rahasia di dalam suatu wadah penampung informasi dengan sedemikian rupa, sehingga keberadaan informasi rahasia yang sisipkan tidak terlihat.Wadah penampung rahasia tersebut dapat berbentuk berbagai jenis file multmedia digital seperti teks, citra, audio dan video Oleh karena itu, digunakan steganografi dengan menerapkan metode Least Significant Bit sebagai teknik yang digunakan menyembunyikan informasi yang berupa nilai–nilai bit. Bit-bit data yang akan disembunyikan atau diamankan dengan Least Significant Bit ke dalam media pesan. Setelah di analisa dan diimplementasikan maka, diperoleh bahwa nilai–nilai bit yang disisipkan ke dalam media pesan masih tampak seperti normal sehingga tidak menimbulkan kecurigaan bagi orang yang mebaca. One Time Pad adalah merupakan salah satu algoritma kriptografi kunci simetris. Algoritma One Time Pad m sebagai kunci untuk melakukan enkripsi dan dekripsi. Dasar teori matriks yang digunakan dalam One Time Pad antara lain adalah perkalian antar matriks dan melakukan invers pada matriks. Diseluruh dunia internet sudah berkembang menjadi salah satu media komunikasi data yang sangat populer. Kemudahan dalam pengunaan dan fasilitas yang sangat lengkap merupakan keunggulan yang dimiliki oleh internet dan bukan rahasia umum dikalngan masyarakat mengguna inetrnet pada saat sekarang ini.

2. Landasan Teori

2.1 Metode Least Significant Bit

Steganografi mempunyai 2 metode, yaituMost Significant Bit (MSB)dan Least Significant Bit (LSB). Untuk Most Significant Bit adalah sebuah metode penyisipan dengan cara menggantinilai bit-bit atas dari data media penampung dengan nilai bit-bit dari data yang akandisembunyikan. Sedangkan Least Significant Bit adalah kebalikan dari Most Significant Bit, yaitu sebuah metode menyembunyikan data pada bit bawah (LSB) pada data pixel yang menyusun file tersebut (Stalling,2003).

Metode Most Significant Bit hanya mampu menyimpan informasi dengan ukuran terbatas. Seperti untuk file atau citra 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 menyisipkan 3 bit data.

Contohnya Huruf A Dapat disisipkan dalam 3 pixel,

Journal of Computer Networks, Architecture and High Performance Computing is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

Journal of Computer Networks, Architecture and High Performance Computing
Volume 1, Number 1, Januari 2019
https://doi.org/10.47709/cnapc.v1i1.45

Submitted : Jan 02, 2019
Accepted : Jan 16, 2019
Published : Jan 18, 2019

Misalnya data raster original adalah sebagai berikut:
00100111 11010011 11001000
R G B

Sedangkan representasi biner huruf A adalah 01100001. Dengan menyisipkan 4 bit kedalam nilai R dan 4 bit kedalam nilai B pada pixel diatas dengan menggunakan metode LSB maka akan dihasilkan:

R awal = 00100111, setelah disisipkan 4 bit dari kiri 0110.
Maka R baru = 00100110.
B awal = 11001000, setelah disisipkan 4 bit dari kiri 0001.
Maka B baru = 11000001.

Terlihat hanya empat bit rendah yang berubah, sehingga untuk mata manusia tidak akan tampak perubahannya.

2.2 Kriptografi

Kriptografi berasal dari bahasa Yunani, menurut bahasa dibagi menjadi dua kripto dan graphia, kripto berarti secret (rahasia) dan graphia berarti writing (tulisan). Menurut teminologinya kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari satu tempat ke tempat lain[7].

2.3 Algoritma One Time Pad

One time pad termasuk dalam kelompok kriptografi simestris. *One-time pad* (*pad* = kertas bloknot) berisi deretan karakter-karakter kunci yang dibangkitkan secara acak. *Cipher* ini diimplementasikan melalui sebuah kunci yang terdiri dari sekumpulan *random* karakter-karakter yang tidak berulang (Munir, 2007). Setiap huruf kunci dijumlahkan *modulo* 26 dengan huruf pada plaintext. Pada *One Time Pad*, tiap huruf kunci digunakan satu kali untuk satu pesan dan tidak digunakan kembali.Panjang stream karakter kunci sama dengan panjang pesan.Satu-satunya algoritma kriptografi yang tidak dapat dipecahkan adalah *one time pad*. *One-Time Pads (OTP).One time pad* ditemukan pada tahun 1917 oleh Major Joseph Mauborgne. *Cipher* ini termasuk ke dalam kelompok algoritma kriptografi simetri. *One time pad* (*pad* = kertas *bloknot*) berisi barisan karakter-karakter kunci yang dibangkitkan secara acak. Aslinya, satu buah one time pad adalah sebuah pita (*tape*) yang berisi barisan karakter-karakter kunci. Satu *pad* hanya digunakan sekali (*one time*) saja untuk mengenkripsi pesan, setelah itu pad yang telah digunakan dihancurkan supaya tidak dipakai kembali untuk mengenkripsi pesan yang lain. Aturan enkripsi yang digunakan persis sama seperti pada *CipherVigenere*. Pengirim pesan menggunakan setiap karakter kunci untuk mengenkripsikan satu karakter *plainteks*. Enkripsi dapat digambarkan sebagai penjumlahan *modulo* 26 dari satu karakter plainteks dengan satu karakter kunci *one time pad* :Adapun rumus enkripsi (sumber : Munir, 2007) adalah sebagai berikut :

$$Ci = (Pi + Ki) \text{ Mod } 26$$

Keterangan :
Pi : karakter plainteks
Ki : karakter kunci
Ci : karakter *Cipher*teks

Sedangkan rumus dekripsi dapat dilihat pada rumus dibawah ini
Pi = (Ci – Ki) mod 26 untuk Ci >= Ki
Pi = (Ci + 26- Ki) mod 26 untuk Ci <= Ki
Keterangan :

Pi : karakter plainteks
Ki : karakter kunci
Ci : karakter *Cipher*teks

Suatu algoritma dikatakan aman, apabila belum ada tidak ada cara untukmenemukan plaintext-nya. Sampai saat ini, hanya algoritma One Time Pad (OTP) yang dinyatakan tidak dapat dipecahkan meskipun diberikan sumber daya yang tidak terbatas. Prinsip enkripsi pada algoritma ini adalah dengan mengkombinasikan masing-masing karakter pada plaintext dengan satu karakter pada kunci. Oleh karena itu, panjang kunci setidaknya harus sama denganpanjang plaintext. Secara teori , adalah hal yang tak mungkin untuk mendeskripsi chipertext tanpa mengetahui kuncinya. Sebab jika kunci yang digunakan salah, akan diperoleh hasil yang salah juga, atau bukan plaintext yang seharusnya. Kemudian setiap kuncinya hanya boleh digunakan untuk sekali pesan. Pengambilan kunci harus dilakukan secara acak supaya tidak dapat diterka lawan dan jumlah karakter kunci harus sebanyak jumlah karakter pesan.

Journal of Computer Networks, Architecture and High Performance Computing is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

Journal of Computer Networks, Architecture and High Performance Computing
Volume 1, Number 1, Januari 2019
https://doi.org/10.47709/cnapc.v1i1.45

Submitted : Jan 02, 2019
Accepted : Jan 16, 2019
Published : Jan 18, 2019

3. Analisa

3.1. Penerapan LSB

Proses Enkripsi Dan Penyisipan

Sebagai contoh kita akan mengamankan pesan dengan isi karakternya adalah "PENDA". Proses pengamanan pesan dapat dilihat pada langkah-langkah berikut ini:

1. Masukkan pesan, *format* yang bisa dibaca adalah txt dan rtf. Misalkan pesan yang akan dimasukkan adalah *format* .txt dengan isi teksnya adalah "PENDA".
2. Masukkan kunci untuk melakukan enkripsi dengan maksimal 20 huruf, adapun kunci yang digunakan yaitu MAKAN
3. Masukkan citra dengan *format* bmp, kemudian citra akan di*resize* menjadi 300x300 piksel.



Gambar 1. Citra yang akan disisip

4. Pesan akan dienkripsi dengan kunci yang dibuat. Pesan akan dikonverikan kedalam table. Nilai pesan asli dan kunci dapat dilihat pada Tabel 1 dibawah ini.

Tabel 1 : Tabel Utama

A	**B**	**C**	**D**	**E**	**F**	**G**
0	1	2	3	4	5	6
H	**I**	**J**	**K**	**L**	**M**	**N**
7	8	9	10	11	12	13
O	**P**	**Q**	**R**	**S**	**T**	**U**
14	15	16	17	18	19	20
V	**W**	**X**	**Y**	**Z**		
21	22	23	24	25		

Tabel 2. Tabel Konversi Alphabet Ke Nilai One Time Pad

Pesan	P	E	N	D	A
	15	4	13	3	0
Kunci	M	A	K	A	N
	12	0	10	0	13

Setelah mendapatkan nilai dari pesan asli dan nilai dari kunci maka kita akan melakukan proses enkripsi dengan menggunakan rumus : $C_i=(P_i + K_i) \bmod 26$

Tabel 3. Tabel Hasil Enkripsi

Nilai Plaintext	Nilai Kunci	Ci = (Pi + Ki) Mod 26	Nilai Chipertext	Chipertext
15	12	(15 + 12) mod 26	1	B
4	0	(4 + 0) mod 26	4	E
13	10	(13 + 10) mod 26	23	X
3	0	(3 + 0) mod 26	3	D
0	13	(0 + 13) mod 26	13	N

Journal of Computer Networks, Architecture and High Performance Computing is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

Journal of Computer Networks, Architecture and High Performance Computing
Volume 1, Number 1, Januari 2019
https://doi.org/10.47709/cnapc.v1i1.45

Submitted : Jan 02, 2019
Accepted : Jan 16, 2019
Published : Jan 18, 2019

Nilai karakter *ciphertext* BEXDN akan disesuaikan dengan nilai karakter pada table ASCII, dan kemudian akan dikonversikan ke bilangan biner.

B : 0 1 0 0 0 0 1 0
E : 0 1 0 0 0 1 0 1
X : 0 1 0 1 1 0 0 0
D : 0 1 0 0 0 1 0 0
N : 0 1 0 0 1 1 1 0

Setelah mendapat bilangan binernya pesan akan disisipkan kedalam *file* citra. Citra yang diinput akan dibaca nilai piksel- pikselnya.

Tabel 4. Nilai Piksel Citra

R	G	B	R	G	B	R	G	B
100	110	25	86	134	41	34	136	52
183	80	20	130	74	15	19	149	65
150	41	19	140	124	30	100	132	45
198	63	19	197	67	27	150	64	22
156	49	15	175	24	16	15	50	53
125	75	35	180	86	31	180	40	25
95	80	21	162	80	25	185	25	15
64	97	25	130	64	14	180	19	18

Kemudian penyisipan akan dilakukan dengan mengganti nilai bit terakhir dengan nilai bilangan biner pesan.

Tabel 5. Nilai Biner Piksel Citra dan Proses Penyisipan Pesan

01000010010001010101100001000100001001110

R	G	B	R	G	B	R	G	B
01100100	01101110	00011001	01010110	10000110	00101001	00100010	10001000	00110100
10110111	01010000	00010100	10000010	01001010	00001111	00010011	10010101	01000001
10010110	01010010	00010011	10001100	01111100	00011110	01100100	10000100	00101101
11000110	00111111	00010011	11000101	01000011	00011011	10010110	01000001	00010110
10011100	00110001	00001111	10101111	00011000	00011001	00001111	00110011	00110101
01111101	01001011	00100011	10110100	01010110	00011001	10110100	00101000	00011001
01011111	01010000	00010101	10100010	01010000	00011001	10111001	00011000	00001111
01000000	01100001	00011001	10000010	01000000	00001110	10110100	00010010	00010010

Hasil Penyisipan Hasil Enkripsi Dengan Citra

01000010010001010101100001000100001001110

R	G	B	R	G	B	R	G	B
01100100	01101111	00011000	01010110	10000110	00101000	00100011	10001000	00110100
10110111	01010000	00010100	10000010	01001011	00001110	00010011	10010100	01000001
10010110	01010011	00010011	10001100	01111100	00011110	01100100	10000101	00101100
11000110	00111110	00010011	11000100	01000010	00011010	10010111	01000000	00010110
10011101	00110001	00001111	10101100	00011000	00011001	00001111	00110011	00110101
01111101	01001011	00100011	10110100	01010110	00011001	10110100	00101000	00011001
01011111	01010000	00010101	10100010	01010000	00011001	10111001	00011000	00001111
01000000	01100001	00011001	10000010	01000000	00001110	10110100	00010010	00010010

Journal of Computer Networks, Architecture and High Performance Computing is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

Journal of Computer Networks, Architecture and High Performance Computing
Volume 1, Number 1, Januari 2019
https://doi.org/10.47709/cnapc.v1i1.45

Submitted : Jan 02, 2019
Accepted : Jan 16, 2019
Published : Jan 18, 2019

4. Algoritma

4.1. Algoritma Enkripsi

Adapun algoritma yang digunakan dalam proses enkripsi ini adalah sebagai berikut :

Algoritma Enkripsi
Deklarasi : K_i, P_i, C_i as string
Input : $K_i \leftarrow$ kunci
: $P_i \leftarrow$ pesan
Output : $C_i \leftarrow$ Chipertext
Proses : mulai
$C_i=(P_i + K_i) \bmod 26$
Selesai

4.2. Algoritma Penyisipan

Adapun algoritma penyisipan pesan dan image adalah sebagai berikut**:**

Algoritma Dekripsi Penyisipan
Deklarasi : n, x_i, l, m, maxval as integer
Input : n ← ukuran pesan
: $x_i \leftarrow$ nilai pixel
: l ← panjang bit
: m ← jumlah bit yang akan disisip
: maxval ←banyaknya bit yang merepresentasikan setiap piksel

Output : $x_j \leftarrow$ hasil penyisipan gambar dan pesan
Proses : mulai
for i= 1 to n do
$x_i=x+e_i-(LSB_m(x_i-1) \bmod 2^m$
if x_i > maxval then
$x_i=x_i-2^m$
end
if x_i>maxval then
$x_i= x_i +2^m$
end

selesai

5. Kesimpulan

Setelah menyelesaikan perancangan aplikasi steganografi dengan metode *least significant bit dan One time pad*, maka penulis dapat mengambil beberapa kesimpulan sebagai berikut :

1. Least Significant Bit, yaitu suatu metode steganografi untuk mendeteksi ada tidaknya pesan rahasia yang disisipkan secara Least Significant Bit pada sebuah citra digital.
2. Cara kerja metode *enhanced* Least Significant Bit adalah dengan mengubah nilai setiap bit pada suatu *byte pixel* data citra bitmap menjadi 0 atau 255 sesuai dengan nilai bit Least Significant Bit yang telah diganti dengan bit pesan pada citra digital tersebut.

6. Referensi

[1]. *Indrajani*, S.Kom, MM. *2011*. *Perancangan Basis Data* Dalam Allin1. Elex Media. Komputindo. Jakarta
[2]. Priyanto, Rahmat., 2009, *Langsung Bisa Visual Basic.Net 2008*, Penerbit ANDI, Yogyakarta
[3]. Rinaldi Munir, 2007, ”Belajar Ilmu Kriptografi” Penerbit Andi, Yogyakarta
[4]. Jogiyanto, Hartono. Analisis dan Desain Sistem Informasi, Edisi III. Yogyakarta: ANDI.2008
[5]. Adi Nugroho, 2010, Nugroho Adi. Rekayasa Perangkat Lunak Berorientasi Objek Dengan Metode USDP. Yogyakarta: Andi
[6]. Badudu J.S dan Zain, Sutan Mohammad, 2006, Kamus Umum Bahasa Indonesia. Jakarta: Pustaka Sinar Harapan.
[7]. Lahagu, A. S., & Hutahaean, H. D. (2017). Perancangan Kios Buku Online Dengan Menerapkan Algoritma MD5 Dalam Pengamanan Record Login. *MEANS (Media Informasi Analisa dan Sistem)*, *2*(1), 23-28.

Journal of Computer Networks, Architecture and High Performance Computing is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.