

II4031 Kriptografi dan Koding

Steganografi

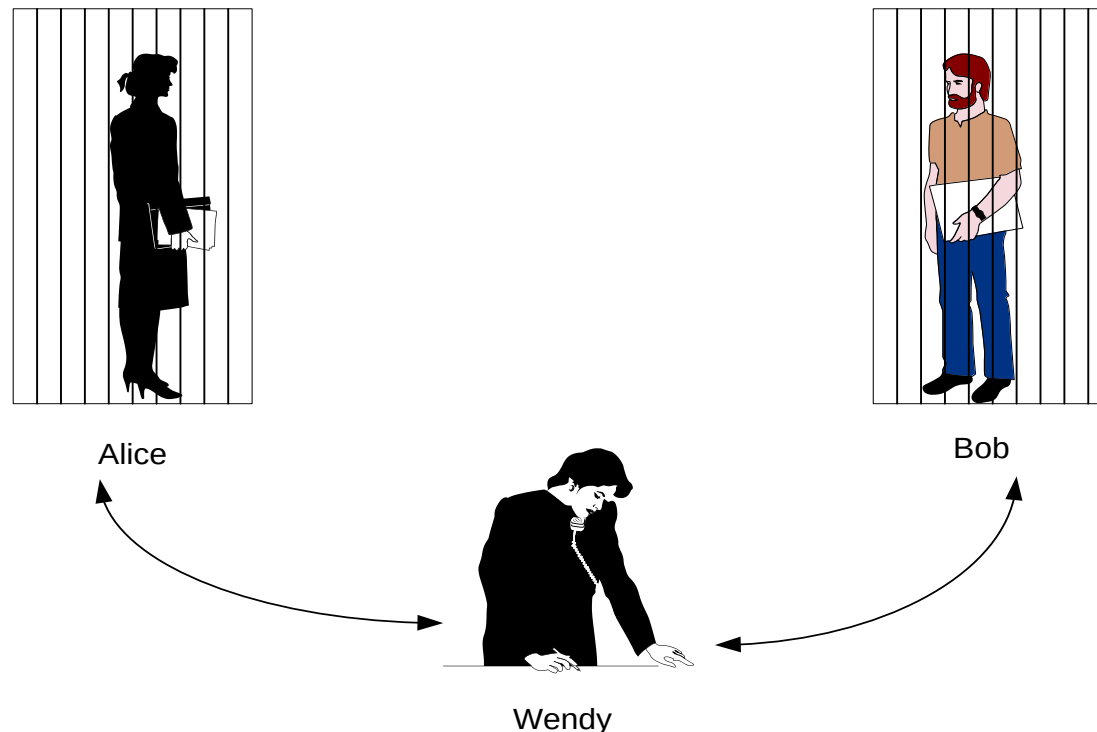


Oleh: Rinaldi Munir

Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2023

The Prisoner's Problem

- Diperkenalkan oleh Simmons – 1983
- Dilakukan dalam konteks *USA – USSR nuclear non-proliferation treaty compliance checking*



Pesan rahasia: “malam ini kita kabur”

- Bagaimana cara Bob mengirim pesan rahasia kepada Alice tanpa diketahui oleh Wendy?
- Alternatif 1: mengenkripsinya

xjT#9uvmY!rc\$7yt59hth@#

Wendy pasti curiga!

- Alternatif 2: menyembunyikannya di dalam tulisan lain

masihkah ada lara apabila memoriku ingat nestapa itu. kita ingin tetap
abadikan kisah asmara. bersamamu usiaku renta.

Wendy tidak akan curiga!

Information hiding dengan steganografi!

Apa Steganografi itu?

- Dari Bahasa Yunani: *steganos* + *graphien*

“**steganos**” (στεγανός): tersembunyi

“**graphien**” (γραφία) : tulisan

steganografi: tulisan tersembunyi (*covered writing*)

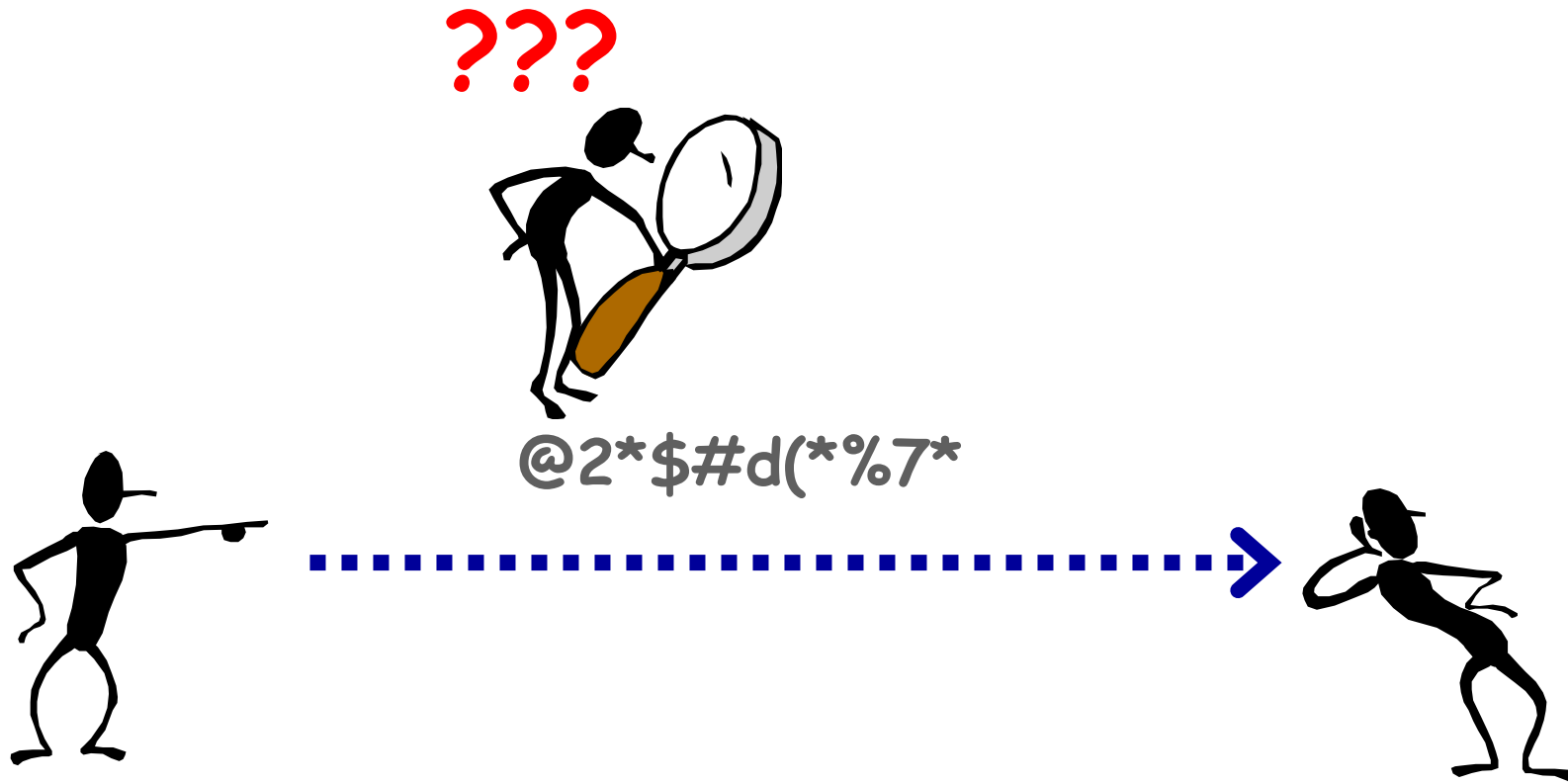
- **Steganography**: ilmu dan seni menyembunyikan pesan rahasia dengan suatu cara sedemikian sehingga tidak seorang pun yang mencurigai keberadaan pesan tersebut.

Tujuan steganografi: pesan tidak terdeteksi keberadaannya

Perbedaan Kriptografi dan Steganografi

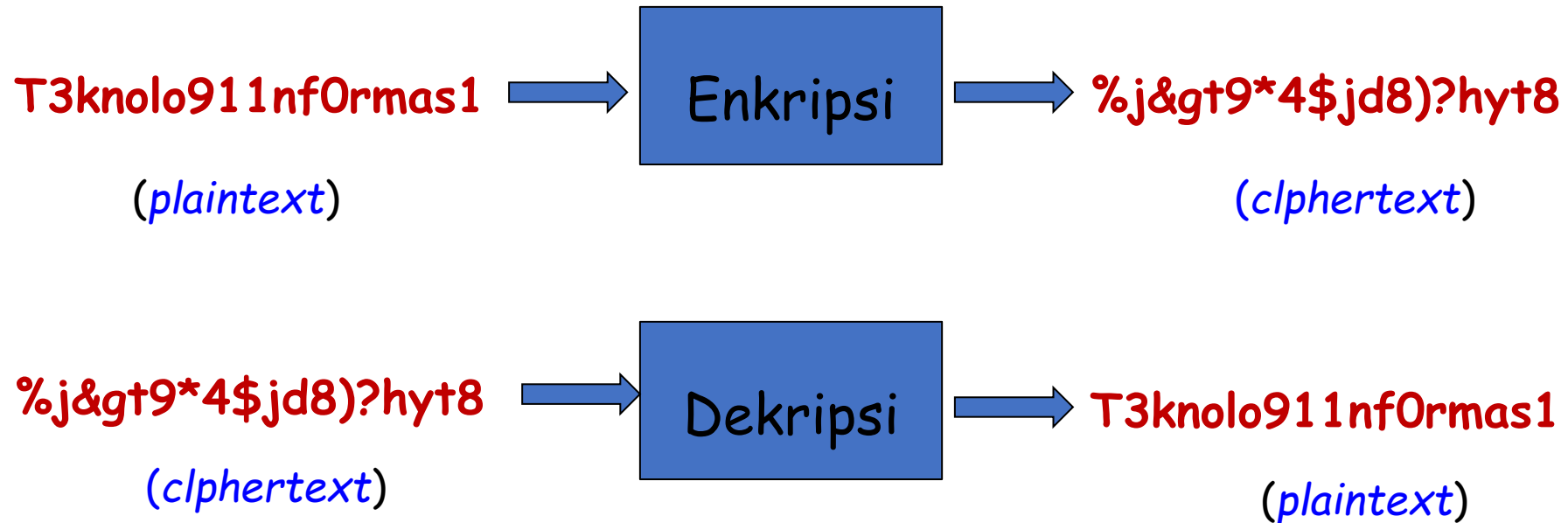
- **Kriptografi**: menyembunyikan ***makna*** atau isi (*content*) pesan
→ Tujuan: agar pesan tidak dapat dibaca oleh pihak ketiga (lawan)
- **Steganografi**: menyembunyikan ***keberadaan*** (*existence*) pesan
→ Tujuan: untuk menghindari kecurigaan (*conspicuous*) dari pihak ketiga (lawan)

Kriptografi

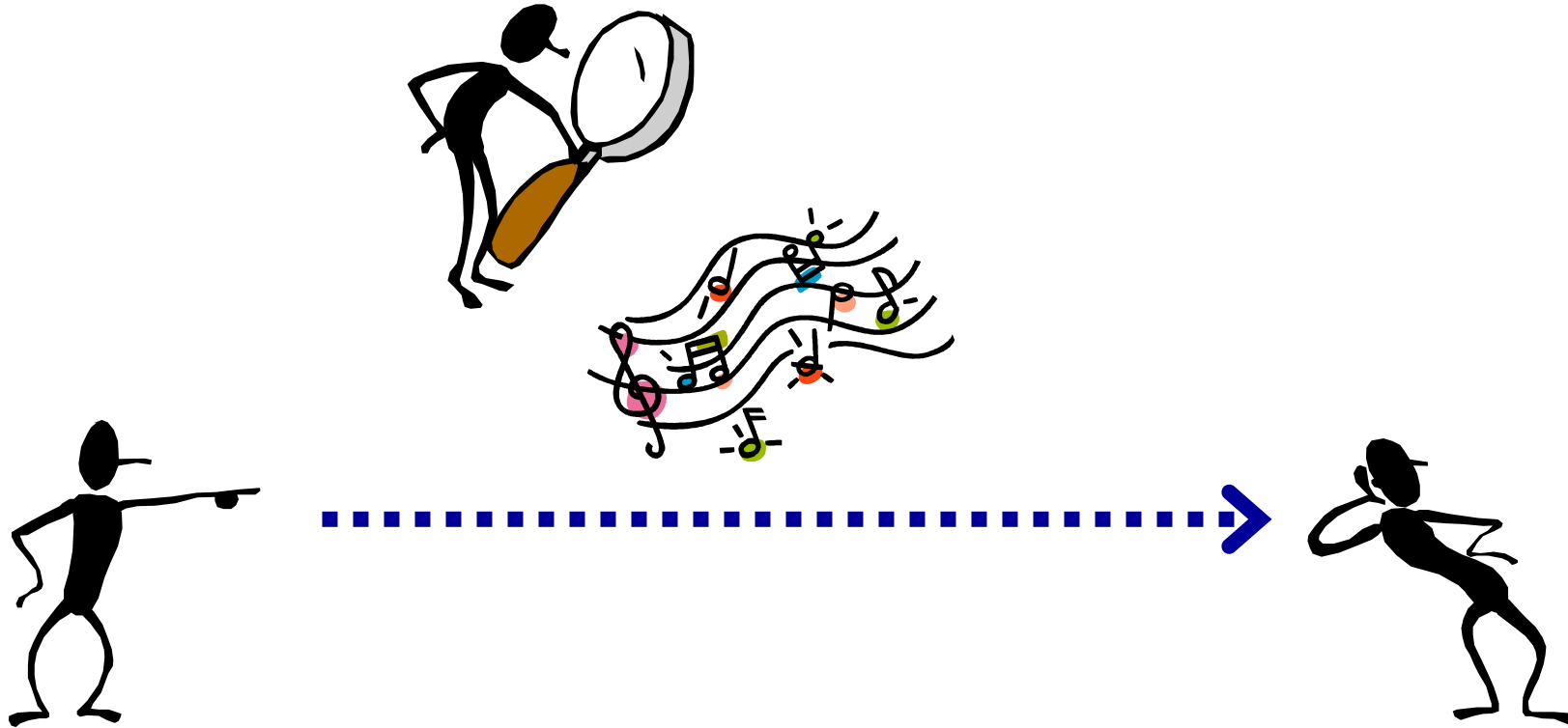


Pesan yang dienkripsi dengan kriptografi menimbulkan kecurigaan bagi pengamat.
Cipherteks dapat dideteksi keberadaannya.

Kriptografi mengenkripsi pesan sehingga maknanya tersamar



Steganografi



Stego-data tidak menimbulkan kecurigaan bagi pengamat
Pesan yang tersembunyi di dalamnya tidak dapat dideteksi.

Steganografi menyembuyikan eksistensi pesan sehingga keberadaannya tidak diketahui

T3knolo911nf0rmas1

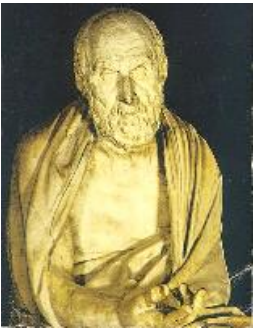


Sejarah Steganografi

- Usia steganografi setara usia kriptografi, dan sejarah keduanya berjalan bersamaan.
- Periode sejarah steganografi dapat dibagi menjadi:
 1. Steganografi kuno (*ancient steganography*)
 2. Steganografi zaman renaissance (*renaissance steganography*).
 3. Steganografi zaman perang dunia
 4. Steganografi modern

Ancient Steganography

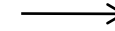
Herodatus



- **Steganografi dengan media kepala budak.**

Ditulis oleh Herodatus (485 – 525 BC), sejarawan Yunani pada tahun 440 BC di dalam buku: *Histories of Herodatus*). Kisah perang antara kerajaan Persia dan rakyat Yunani.

Herodatus menceritakan cara **Histaiaeus** mengirim pesan kepada **Aristagoras of Miletus** untuk melawan Persia. Caranya: Dipilih beberapa budak. Kepala budak dibotaki, ditulisi pesan dengan cara tato, rambut budak dibiarkan tumbuh, budak dikirim. Di tempat penerima kepala budak digunduli agar pesan bisa dibaca.



- **Penggunaan *tablet wax***

Orang-orang Yunani kuno menulis pesan rahasia di atas kayu yang kemudian ditutup dengan lilin (*wax*).

Di dalam bukunya, Heradatus menceritakan Demaratus mengirim peringatan tentang serangan yang akan datang ke Yunani dengan menulis langsung pada tablet kayu yang kemudian dilapisi lilin dari lebah.



- Penggunaan tinta tak-tampak (*invisible ink*)



Pliny the Elder.
AD 23 - 79

Pliny the Elder menjelaskan penggunaan tinta dari getah tanaman *thithymallus*. Jika dituliskan pada kertas maka tulisan dengan tinta tersebut tidak kelihatan, tetapi bila kertas dipanaskan berubah menjadi gelap/coklat

- **Penggunaan kain sutra dan lilin**
- Orang Cina kuno menulis catatan pada potongan-potongan kecil sutra yang kemudian digumpalkan menjadi bola kecil dan dilapisi lilin.
- Selanjutnya bola kecil tersebut ditelan oleh si pembawa pesan.
- Pesan dibaca setelah bola kecil dikeluarkan dari perut si pembawa pesan dengan cara BAB.

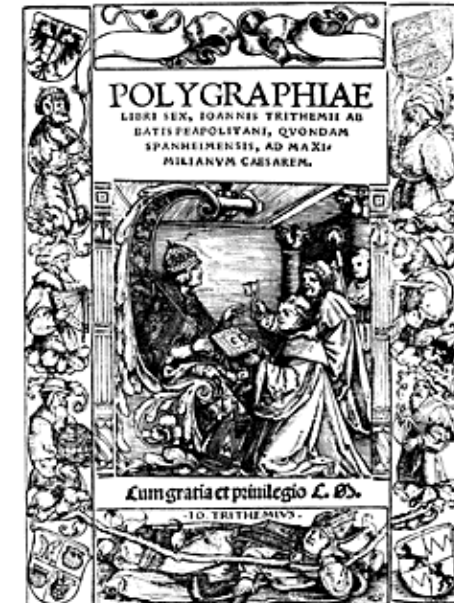
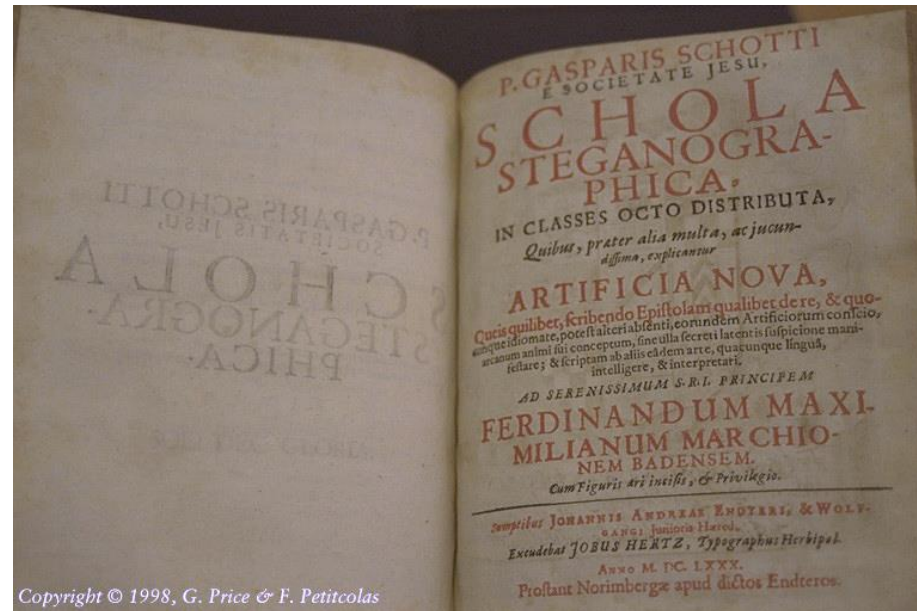


Renaissance Steganography



Johannes
Trithemius
(1404-1472)

Tahun 1499, Johannes Trithemius menulis buku **Steganographia**, yang menceritakan tentang metode steganografi berbasis karakter



Selanjutnya tahun 1518 dia menulis buku tentang steganografi dan kriptografi, berjudul **Polygraphiae**.



Giovanni Battista Porta
(1535-1615)

Giovanni Battista Porta menggambarkan cara menyembunyikan pesan di dalam telur rebus.

Caranya, pesan ditulis pada kulit telur yang dibuat dari tinta khusus yang dibuat dengan satu ons tawas dan setengah liter cuka.

Prinsipnya penyembunyiannya adalah tinta tersebut akan menembus kulit telur yang berpori, tanpa meninggalkan jejak yang terlihat.

Tulisan dari tinta akan membekas pada permukaan isi telur yang telah mengeras (karena sudah direbus sebelumnya). Pesan dibaca dengan membuang kulit telur

World War Steganography

- Penggunaan tinta tak-tampak (*invisible ink*) dalam spionase.
 - Pada Perang Dunia II, tinta tak-tampak digunakan untuk menulis pesan rahasia
 - Tinta terbuat dari campuran susu, sari buah, cuka, dan urine.
 - Cara membaca: Kertas dipanaskan sehingga tulisan dari tinta tak-tampak tersebut akan menghitam.



Seorang agen FBI sedang menggunakan sinar ultraviolet untuk membaca tulisan yang tersembunyi pada kertas yang dicurigai dari agen spionase.

- **Steganografi dalam Perang Dunia II: *Null Cipher***

Pesan berikut dikirim oleh Kedubes Jerman pada PD II:

*Apparentl**y** n**e**utral's p**r**otest **i**s t**h**oroughly d**i**scounted a**n**d i**g**nored.
I**s**man h**a**rd h**i**t. B**l**ockade **i**ssue a**f**fects p**r**etext f**o**r e**m**bargo o**n** b**y**-
p**r**oducts, e**j**ecting s**u**ets a**n**d v**e**getable o**i**ls.*

Ambil huruf kedua setiap kata, diperoleh pesan berikut: *Pershing
sails from NY June 1.*

Contoh *Null Cipher* lainnya:

Big **r**umble **i**n **N**ew **G**uinea.

The **w**ar **o**n **c**elebrity **a**cts **s**hould **e**nd **s**oon.

Over **f**our **d**ie **e**cstatic **e**lephants **r**eplicated.

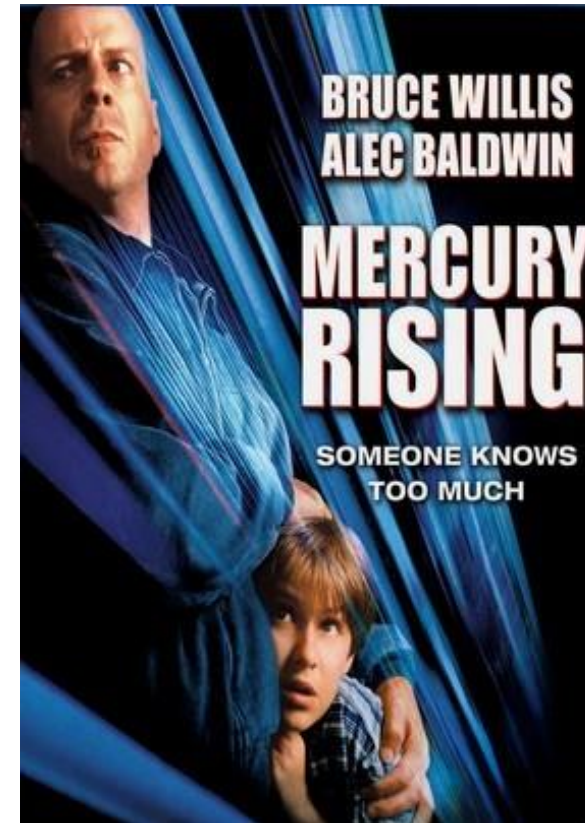
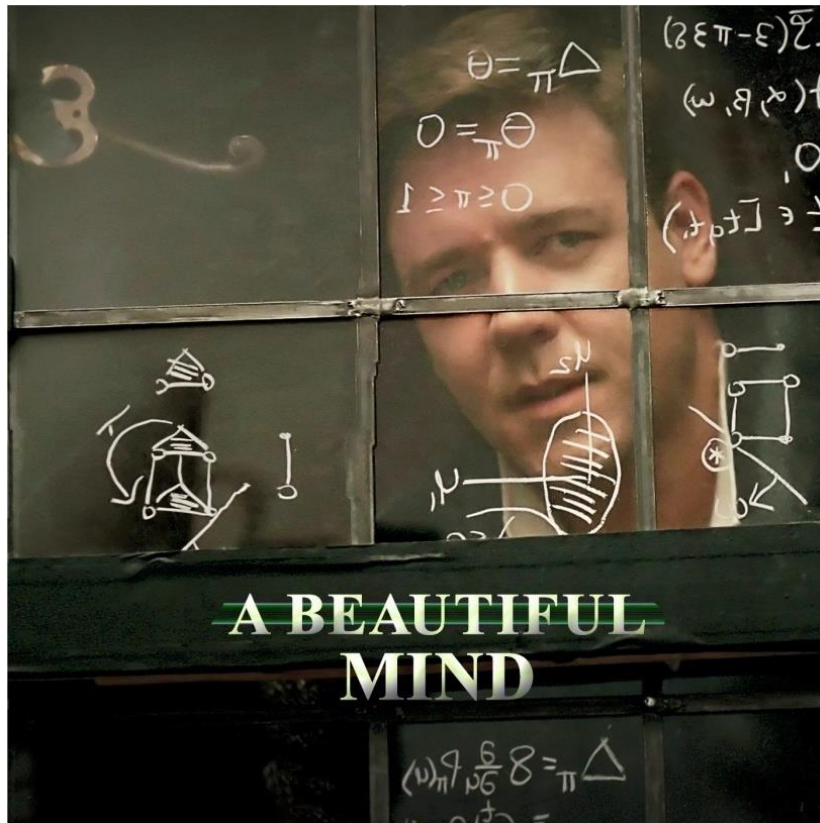
Bring two cases of deer.

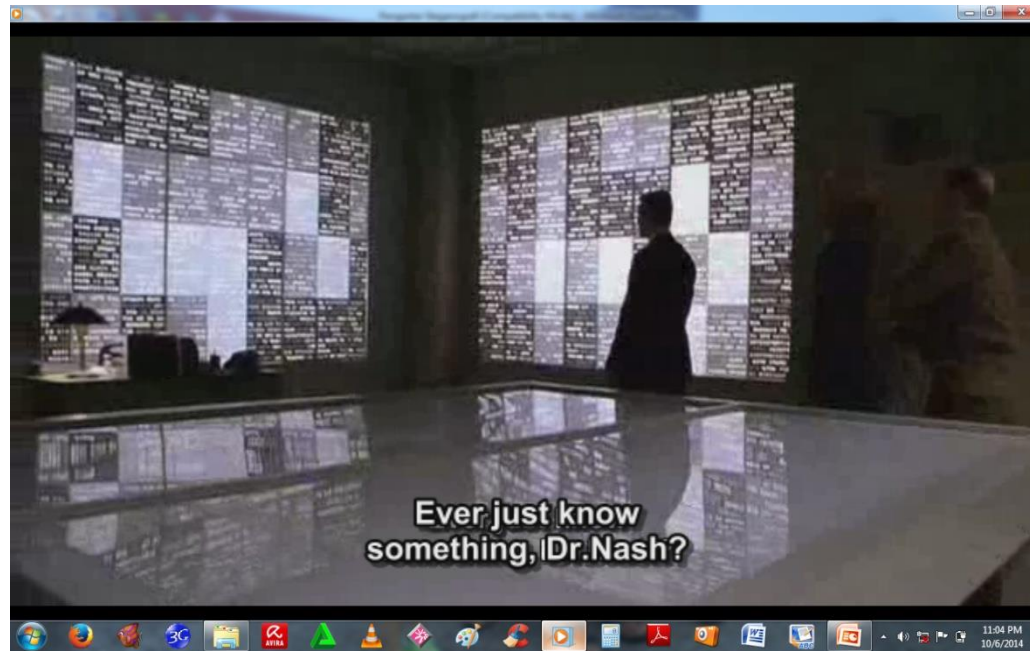
Fishing freshwater bends and saltwater coasts rewards anyone feeling stressed. Resourceful anglers usually find masterful leapers fun and admit swordfish rank overwhelming anyday.

Dengan mengambil huruf ketiga pada setiap kata diperoleh pesan berikut:

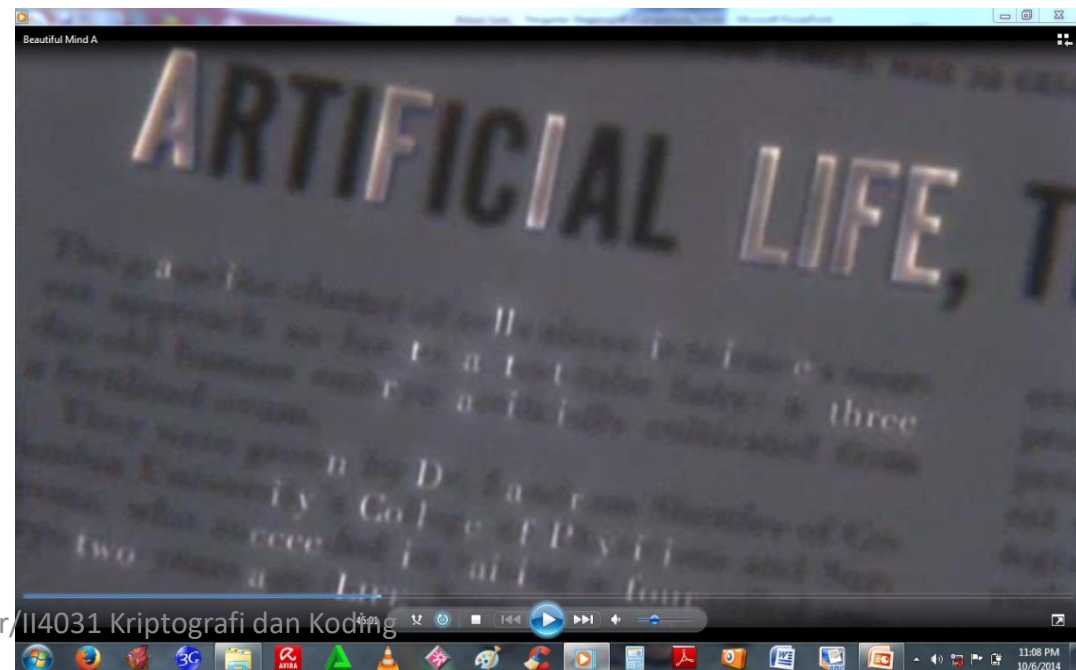
Send Lawyers, Guns, and Money.

- Steganografi di dalam film *Mercury Rising* dan *Beautiful Mind*





Beberapa adegan film *Beautiful Mind* yang memperlihatkan steganografi



Steganografi dan Terorisme

- Ilmu steganografi mendadak naik daun ketika pasca 11 September 2001 pihak FBI menuding *Al-Qaidah* menggunakan steganografi untuk menyisipkan pesan rahasia melalui video atau gambar yang mereka rilis secara teratur di Internet.



Steganografi Digital

- Steganografi digital: menyembunyikan pesan digital di dalam dokumen digital lainnya.
- *Carrier file*: dokumen digital yang digunakan sebagai media untuk menyembunyikan pesan.

1. Teks

“Kita semua bersaudara”

- Txt
- doc
- html

2. Audio



- wav
- mp3

3. Gambar (*image*)

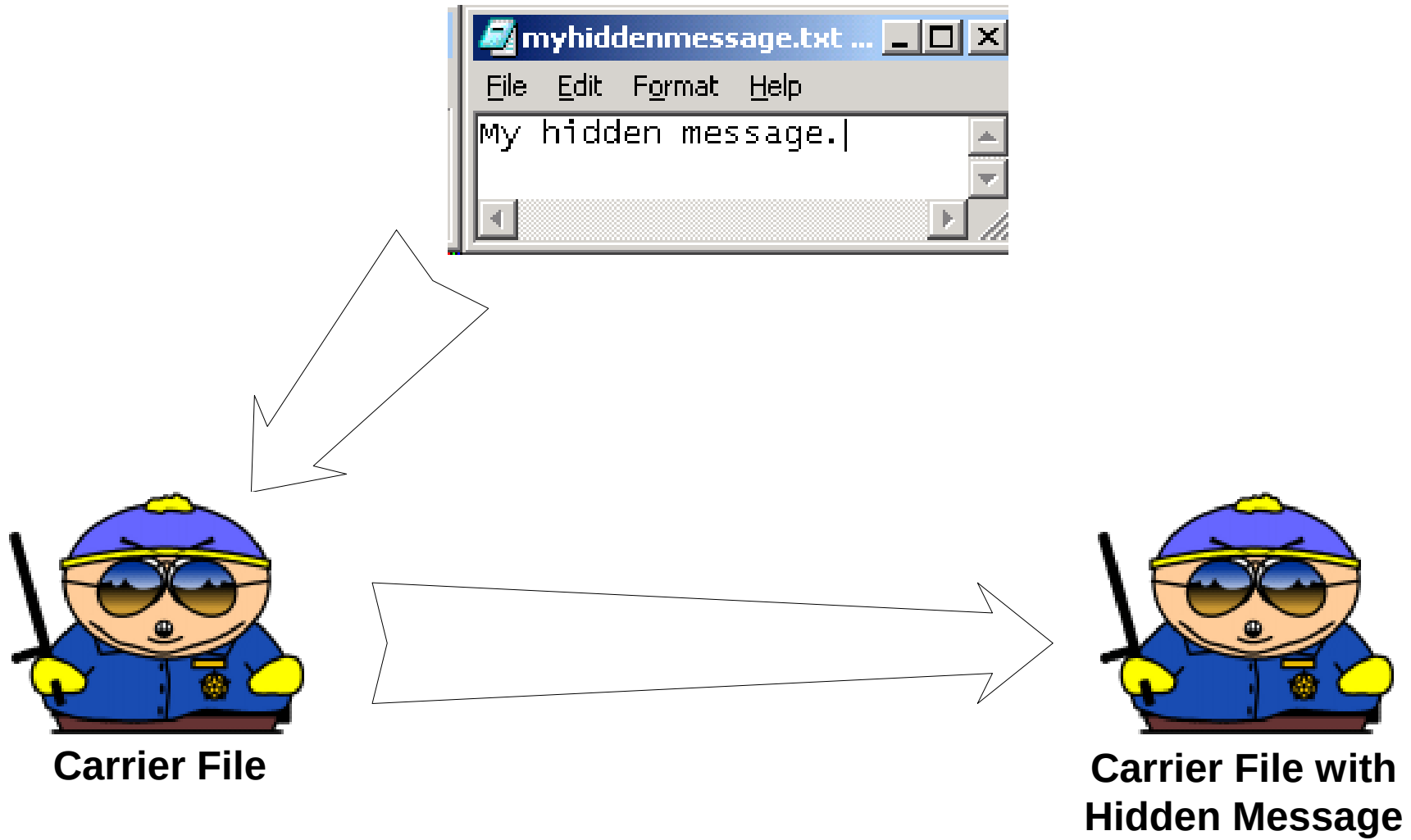


- bmp
- jpeg
- gif
- png

4. Video



- mpeg
- avi
- mp4



Terminologi Steganografi

1. *Embedded message (hiddentext) atau secret message*: pesan yang disembunyikan .

Bisa berupa teks, gambar, audio, video, dll

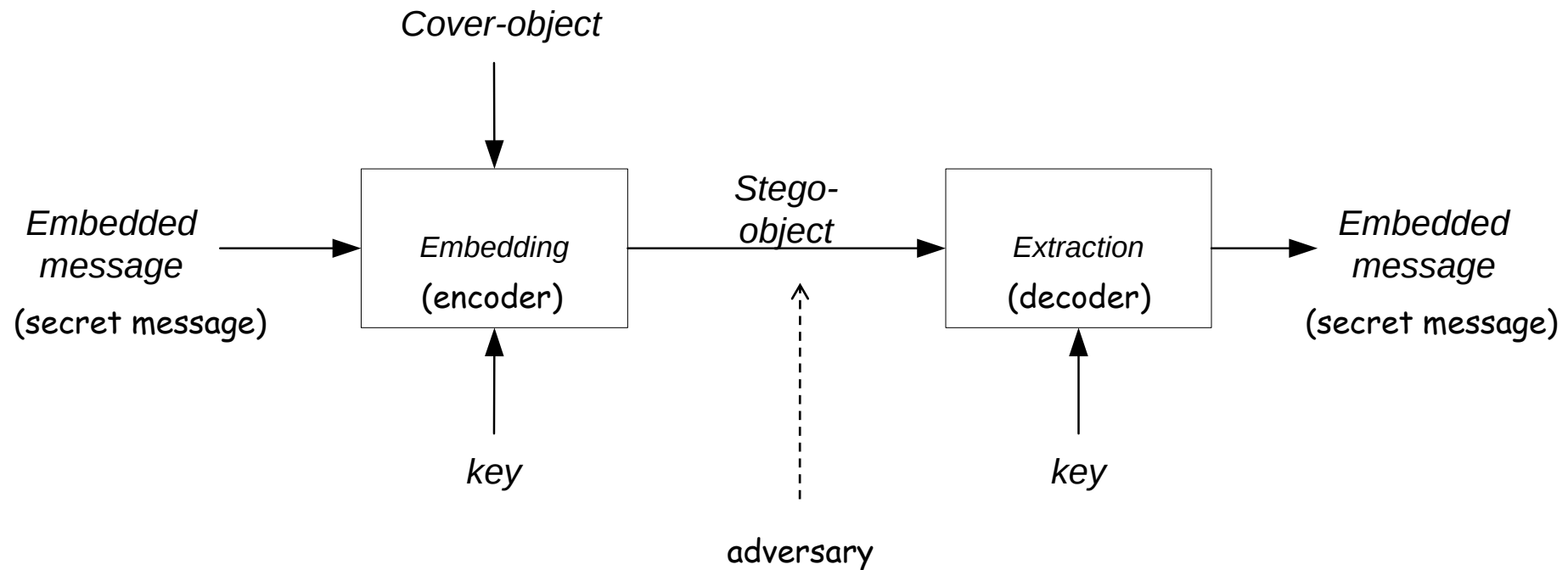
2. *Cover-object (covertext)*: pesan yang digunakan untuk menyembunyikan *embedded message*.

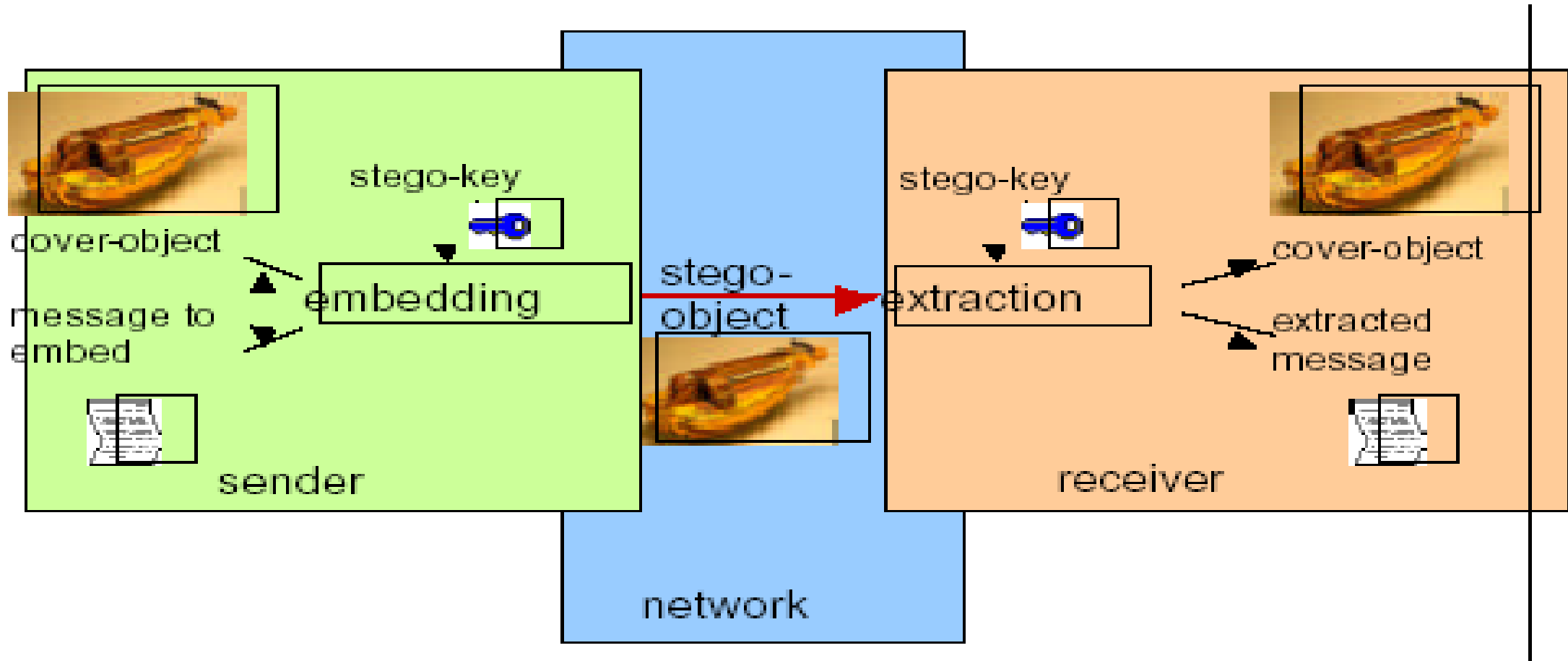
Bisa berupa teks, gambar, audio, video, dll

2. *Stego-object (stegotext)*: pesan yang sudah berisi pesan *embedded message*.

3. *Stego-key*: kunci yang digunakan untuk menyisipkan pesan dan mengekstraksi pesan dari stegotext.

Diagram Proses Steganografi





Istilah keilmuan serumpun terasa memberikan distorsi persepsi pada maksud sebenarnya. Persepsi yang segera terbentuk dengan istilah tersebut adalah pertumbuhan dari akar-akar ilmu membentuk suatu rumpun, yang berarti bahwa nuansa historis organisasi/kelompok/unit yang mewadahnya.



Embedded message

Cover-image

Stego-image

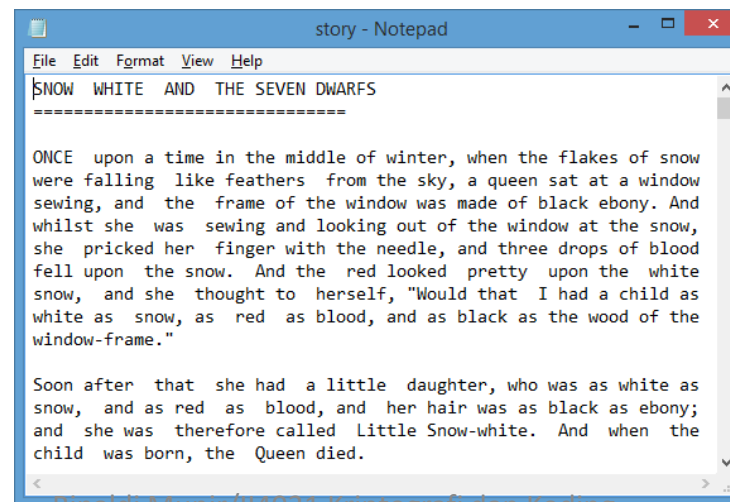
GIF image



Cover image



Stego-image



Rinaldi Murni/114031 Kriptografi dan Koding

Embedded message



Cover image



Stego-image





Cover image

Embedded image





Stego-image

Extracted image



Kriteria Steganografi yang Bagus

1. *Imperceptible*

Keberadaan pesan rahasia tidak dapat dipersepsi secara visual atau secara audial

2. *Fidelity.*

Kualitas *cover-object* tidak jauh berubah akibat penyisipan pesan rahasia.

3. *Recovery.*

Pesan yang disembunyikan harus dapat diekstraksi kembali.

4. *Capacity*

Ukuran pesan yang disembunyikan sedapat mungkin besar

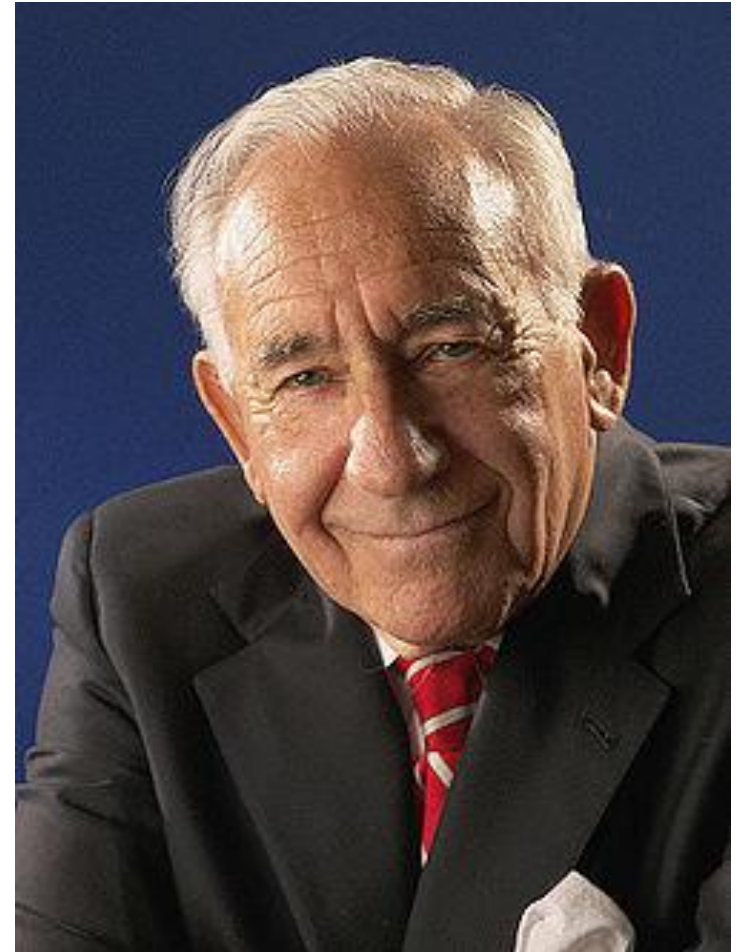
Catatan: *Robustnes* bukan isu penting di dalam steganografi

Kombinasi Kriptografi dan Steganografi

- Steganografi bukan pengganti kriptografi, tetapi keduanya saling melengkapi.
- Keamanan pesan rahasia dapat ditingkatkan dengan menggabungkan kriptografi dan steganografi.
- Mula-mula pesan dienkripsi dengan algoritma I kriptografi, selanjutnya pesan terenkripsi disembunyikan di dalam media lain (citra, video, audio, dll).

Steganography has its place in security. It is not intended to replace cryptography but supplement it. Hiding a message with steganography methods reduces the chance of a message being detected. However, if that message is also encrypted, if discovered, it must also be cracked (yet another layer of protection).

*(David Kahn, penulis buku *The Codebreakers - The Story of Secret Writing*)*



Ranah Steganografi

Berdasarkan ranah operasinya, metode-metode steganografi dapat dibagi menjadi dua kelompok

- *Spatial (time) domain methods*

Memodifikasi langsung nilai *byte* dari *cover-object* (nilai *byte* merepresentasikan intensitas/warna *pixel* atau amplitudo)

Contoh: Metode *LSB*

- *Transform domain methods*

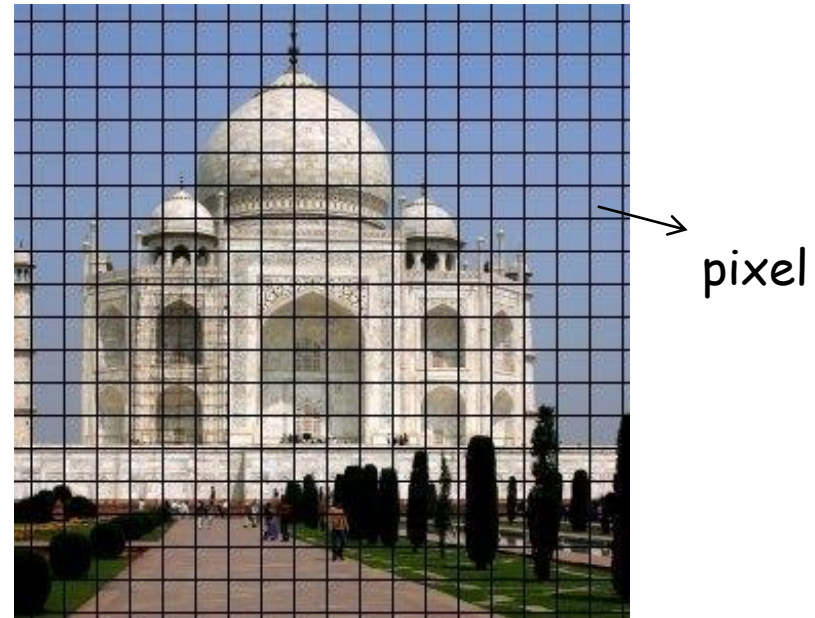
Memodifikasi hasil transformasi sinyal dalam ranah transform (hasil transformasi dari ranah spasial ke ranah lain (misalnya ranah frekuensi).

Contoh: Metode *Spread Spectrum*

Metode LSB

Citra Digital

- Citra terdiri dari sejumlah *pixel*. Citra 1200 x 1500 berarti memiliki 1200 x 1500 pixel = 1.800.000 pixel



- Setiap *pixel* panjangnya n -bit.
Citra biner $\rightarrow$ 1 bit/pixel
Citra *grayscale* $\rightarrow$ 8 bit/pixel
Citra *true color* $\rightarrow$ 24 bit/pixel

Citra Lenna



True color image
(24-bit)

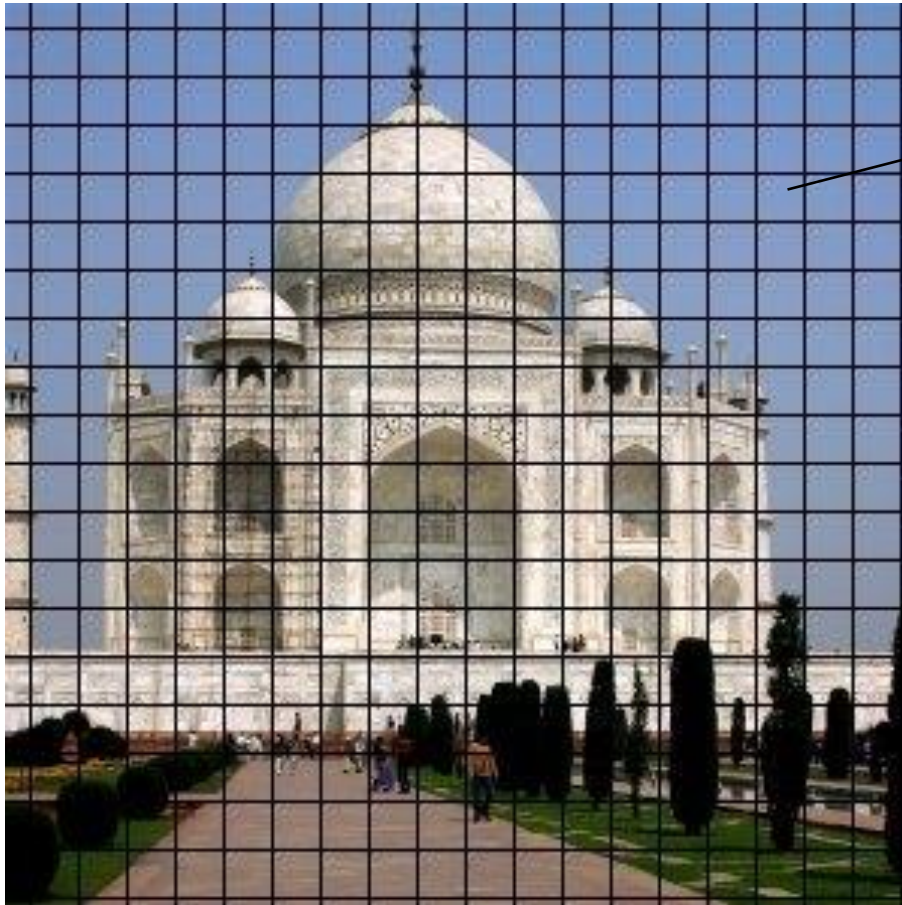


Grayscale image
(8-bit)



Bimary image
(1-bit)

Pada citra 24-bit (*real image*), 1 pixel = 24 bit,
terdiri dari komponen RGB (Red-Green-Blue)

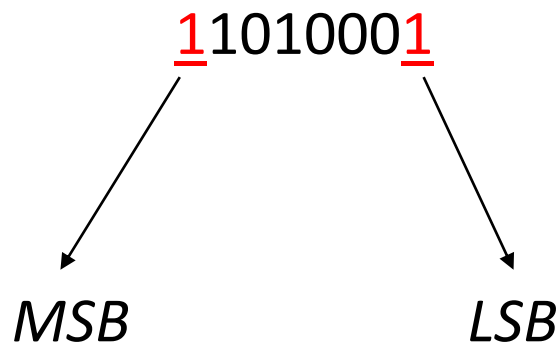


100100111001010010001010
R G B

Metode LSB

- Di dalam setiap *byte* bit-bitnya tersusun dari kiri ke kanan dalam urutan yang kurang berarti (*least significant bits* atau *LSB*) hingga bit-bit yang berarti (*most significant bits* atau *MSB*).
- Susunan bit pada setiap *byte* adalah $b_7b_6b_5b_4b_3b_2b_1b_0$.

Contoh:



LSB = *Least Significant Bit*
MSB = *Most Significant Bit*

Nilai dalam desimal = $1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 209$

- Mengubah nilai bit LSB tidak mengubah persepsi citra secara keseluruhan.
- Sebab, mengubah bit *LSB* hanya mengubah nilai *byte* satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya.

Contoh: 11010001 → 11010000
(209) (208)

Jika 11010001 misalnya menyatakan warna **merah**, maka 11010000 juga menyatakan warna **merah** yang berubah sangat-sangat sedikit → tidak bisa dibedakan oleh mata manusia

Misalkan semua bit LSB pada citra berwarna dibalikkan dari semula 0 menjadi 1; dari semula 1 menjadi 0



Sebelum



Sesudah

Adakah terlihat perbedaanya?

Misalkan semua bit LSB pada citra *grayscale* dibalikkan
Dari semula 0 menjadi 1; dari semula 1 menjadi 0



Sebelum



Sesudah

Adakah terlihat perbedaanya?

Contoh 1:

- Tinjau 1 buah *pixel* dari citra 24-bit (3 x 8 bit):

10000010

(130)

01111011

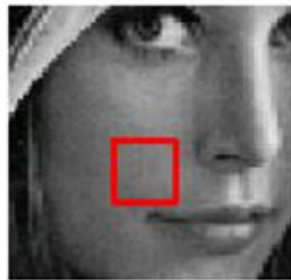
(123)

01110101

(117)

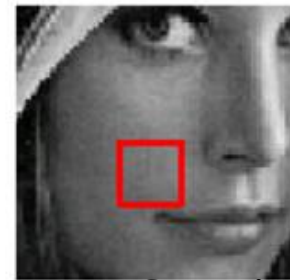
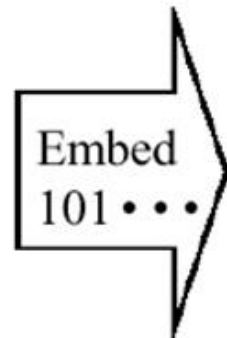
- Bit bit-bit *embedded message*: 101

- *Embed*: 00110011 10100010 11100011



original

130 = 10000010
123 = 01111011
117 = 01110101
.....



Stego-image

131 = 10000011
122 = 01111010
117 = 01110101
.....

Original: misalkan *pixel* [130, 123, 117] berwarna "ungu"
Stego-image: *pixel* [131, 122, 117] tetap "ungu" tapi berubah sangat sedikit.
Mata manusia tidak dapat membedakan perubahan warna yang sangat kecil.

Pergeseran warna sebesar 1 dari 256 warna tidak dapat dilihat oleh manusia

PESAN RAHASIA :



Sumber: TA Yulie Anneria Sinaga 13504085

Contoh 2:

- Jika pesan = 10 bit, maka jumlah *byte* yang digunakan = 10 *byte*

- Contoh susunan *byte* yang lebih panjang:

00110011 10100010 11100010 10101011 00100110
10010110 11001001 11111001 10001000 10100011

- Pesan: 1110010111

- Hasil penyisipan pada bit *LSB*:

0011001**1** 1010001**1** 1110001**1** 1010101**0** 0010011**0**
1001011**1** 1100100**0** 1111100**1** 1000100**1** 1010001**1**

Ekstraksi Pesan dari *Stego-image*

- Bit-bit pesan yang disembunyikan di dalam citra harus dapat diekstraksi kembali.
- Caranya adalah dengan membaca *byte-byte* di dalam citra, mengambil bit LSB-nya, dan merangkainya kembali menjadi bit-bit pesan.
- Contoh: Misalkan *stego-object* adalah sbb

00110011 10100011 11100011 10101010 00100110
10010111 11001000 11111001 10001001 10100011

Ekstrak bit-bit LSB: 1110010111

Menghitung Ukuran Pesan yang dapat Disembunyikan

- Ukuran pesan yang akan disembunyikan bergantung pada ukuran *cover-object*.
- Misalkan pada citra *grayscale* (1 *byte/pixel*) 256 x 256 *pixel* :
 - jumlah *pixel* = jumlah *byte* = $256 \times 256 = 65536$
 - setiap *byte* dapat menyembunyikan 1 bit pesan di LSB-nya
 - jadi ukuran maksimal pesan = 65536 bit = 8192 *byte* = 8 KB
- Pada citra berwarna 24-bit berukuran 256×256 *pixel*:
 - jumlah *pixel* $256 \times 256 = 65536$
 - setiap *pixel* = 3 *byte*, berarti ada $65536 \times 3 = 196608$ *byte*.
 - setiap *byte* dapat menyembunyikan 1 bit pesan
 - jadi ukuran maksimal pesan = 196608 bit = 24576 *byte* = 24KB

Beberapa Varian Metode LSB

1. *Sequential*

- Bit-bit pesan disembunyikan secara sekuensial pada *pixel-pixel* citra.
- Misalkan ukuran pesan = 15 bit, maka urutan *pixel-pixel* yang digunakan untuk penyembunyian bit adalah:

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	

Ekstraksi pesan dari *Stego-image*

- Pada proses ekstraksi pesan, *pixel-pixel* dibaca secara sekuensial mulai dari *pixel* pertama sampai *pixel* yang menyimpan bit pesan terakhir
- Ambil setiap *byte* dari *pixel*, ekstraksi bit LSB-nya.
- Rangkailah bit-bit LSB menjadi bit-bit pesan semula.

2. Acak

- Untuk membuat penyembunyian pesan lebih aman, bit-bit pesan tidak disimpan pada pixel-pixel yang berurutan, namun dipilih secara acak.
- Pembangkit bilangan acak-semu (*PRNG: pseudo-random number generator*) digunakan untuk membangkitkan bilangan acak.
- Umpan (*seed*) untuk pembangkit bilangan acak berlaku sebagai kunci (*stego-key*).

- Misalnya jika terdapat 64 *byte* dan 15 bit pesan yang akan disembunyikan. *Pixel-pixel* dipilih secara acak, seperti pada gambar berikut.

				5			8
	10					4	
			13		2		
7							9
		1			12		
		15					
11						3	
			6				14

Ekstraksi pesan dari *Stego-image*

- Posisi *pixel* yang menyimpan bit pesan dapat diketahui dari bilangan acak yang dibangkitkan oleh *PRNG*.
- Jika kunci yang digunakan pada waktu ekstraksi sama dengan kunci pada waktu penyisipan, maka bilangan acak yang dibangkitkan juga sama.
- Dengan demikian, bit-bit pesan yang bertaburan di dalam citra dapat dikumpulkan kembali.

3. *m*-bit LSB

- Untuk meningkatkan ukuran pesan yang disembunyikan, maka digunakan lebih dari 1 bit LSB untuk setiap *byte*.
- Susunan bit pada setiap *byte* adalah $b_7b_6b_5b_4b_3b_2b_1b_0$. Jika diambil 2-bit LSB, maka bit yang digunakan adalah bit b_1 dan bit b_0

Contoh: 11010010 → 2 bit LSB terakhir dipakai untuk menyembunyikan pesan.

- *Trade-off*: Semakin banyak bit LSB yang digunakan, semakin besar ukuran pesan yang dapat disembunyikan, tetapi semakin turun kualitas *stego-image*.
- Pesan dapat disembunyikan secara sekuensial atau secara acak pada *pixel-pixel* di dalam citra.

4. Enkripsi

- Pesan dapat dienkripsi terlebih dahulu sebelum disembunyikan ke dalam citra.
- Teknik enkripsi yang sederhana misalnya dengan meng-XOR-kan bit-bit pesan dengan bit-bit kunci. Jumlah bit-bit kunci sama dengan jumlah bit pesan.
- Bit-bit kunci dibangkitkan secara acak.
- Kunci untuk pembangkitan bit-bit kunci menjadi *stego-key*.
- Jika dipakai teknik acak dalam memilih *pixel-pixel*, maka ada dua *stego-key*: satu untuk pembangkitan bit-bit kunci, satu lagi untuk pembangkitan posisi *pixel* yang dipilih untuk menyembunyikan pesan.

How to Hack a Computer Using Just An Image

Monday, June 01, 2015 Swati Khandelwal

512 Like 8.5K Share 12.8K Tweet 923 Share 84 share 19.2K

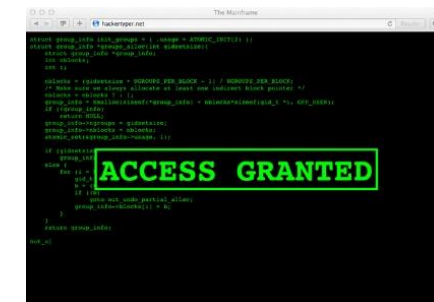


Next time when someone sends you a photo of a cute cat or a hot chick than be careful before you [CLICK](#) on the image to view — it might hack your machine.

Yes, the normal looking images could hack your computers — thanks to a technique discovered by security researcher *Saumil Shah* from India.

Dubbed "*Stegosplot*," the technique lets hackers hide malicious code inside the pixels of an image, hiding a malware exploit in plain sight to infect target victims.

Just look at the image and you are HACKED!



Program Steganografi dengan MATLAB

- Kode program Matlab berikut ini menyisipkan pesan dengan metode LSB.

A. Program penyisipan pesan

- Input:
 1. Citra berwarna (RGB)
 2. Pesan (diketik langsung dari keyboard)
- Output: citra stego (stego image)

B. Program ekstraksi pesan

- Input: Citra stego
- Output: pesan

A. Program penyisipan pesan

```
% Nama file: LSB_sisip.m
% Penyisipan pesan di dalam citra dengan metode modifikasi LSB. Panjang
% pesan (L) disimpan di dalam citra stegano.
% Asumsi: jumlah bit untuk merepresentasikan panjang pesan maksimal 30 bit.
% Input: 1. Citra berwarna (RGB)
%         2. Teks dari keyboard
% Output: citra stego (harus disimpan dalam format BMP)
% Diprogram oleh Rinaldi Munir, @Informatika STEI-ITB

clear all;
% Baca citra cover dan tampilkan
namaFile = input('Ketikkan nama file citra cover: ', 's');
coverImage = imread(namaFile);
imshow(namaFile); title ('Citra cover ');

tinggi = size(coverImage, 1);
lebar = size(coverImage, 2);
total_byte = tinggi * lebar * 3;
total_bit_embed = total_byte;

% Baca pesan
pesan = input('Ketikkan pesan: ', 's');
```

```

% Baca pesan
pesan = input('Ketikkan pesan: ', 's');

% Hitung panjang pesan
L = length(pesan);

%Konversi panjang pesan ke dalam biner
nL = dec2bin(L, 30);

%Konversi pesan ke dalam vektor biner
for i = 1 : L
    biner((i-1)*8 + 1 : i*8) = dec2bin(pesan(i), 8);
end

% Gabungkan bit-bit panjang pesan dengan bit-bit pesan
biner = [nL, biner];
m = length(biner);          % Ubah vektor biner ke numerik

if m > total_bit_embed
    disp('Ukuran citra tidak mencukup untuk penyembunyian pesan');

```

```

else
    biner = biner(:);           % Transpose vektor biner
    biner = str2num(biner);     % Ubah bit biner ke numerik

    % Lakukan penyisipan bit-bit pesan ke dalam pixel-pixel
    disp('Penyisipan pesan....');
    idx = 1; % Mulai dari bit pertama pesan
    for row = 1 : size(coverImage,1)
        for col = 1 : size(coverImage,2)
            for i = 1 : 3
                if idx <= m
                    coverImage(row, col, i) = bitset(coverImage(row, col, i), 1, biner(idx));
                    idx = idx + 1;
                end
            end
        end
    end
    end
    stegoImage = coverImage;
    % Simpan citra stego
    namaFile = input('Ketikkan nama file citra stego (bmp): ', 's');
    imwrite(stegoImage, namaFile);
    % Tampilkan citra stego
    figure; imshow(stegoImage); title ('Citra stego');
    disp('Selesai');
end

```

Running program:

>> LSB_sisip

Ketikkan nama file citra cover: peppers.bmp

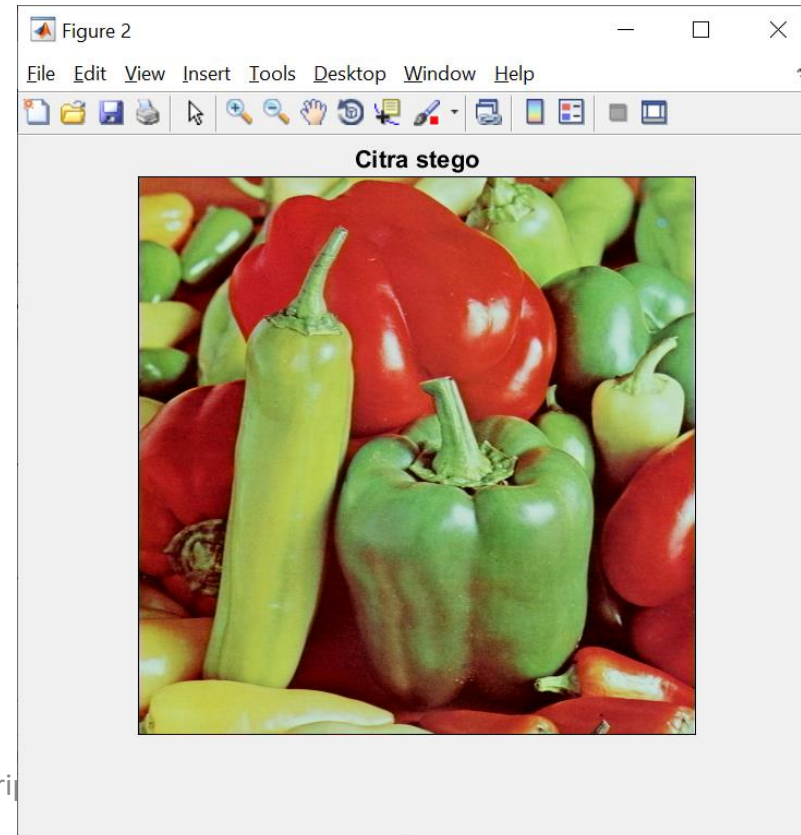
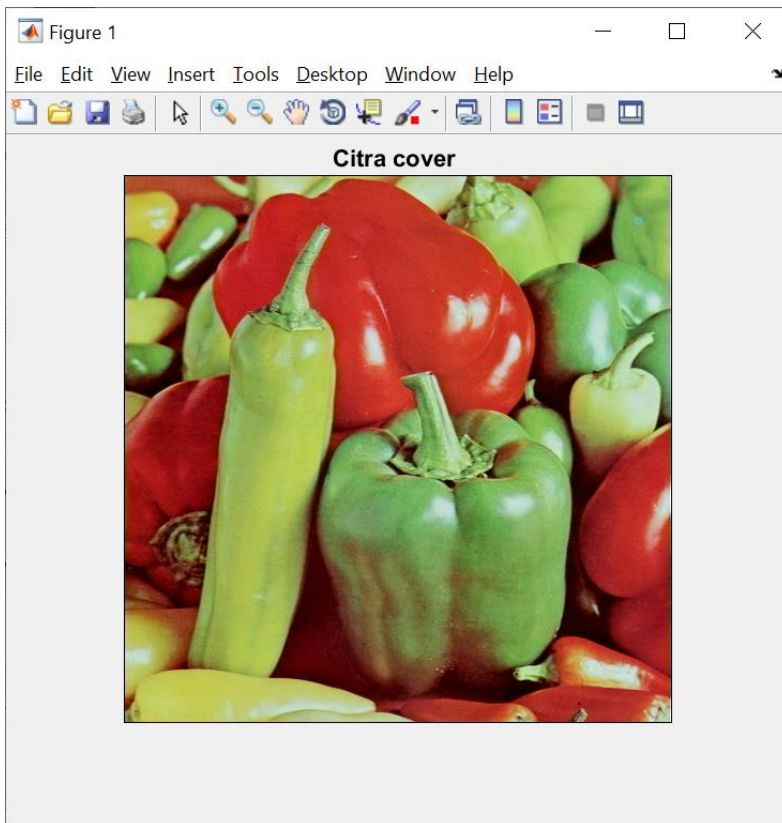
Ketikkan pesan: Mari kita jaga negara kesatuan Republik Indonesia tercinta. Oke?

Penyisipan pesan....

Ketikkan nama file citra stego (bmp): output.bmp

Selesai

>>



B. Program ekstraksi pesan

```
% Nama file: LSB_ekstrak.m
% Ekstraksi pesan dari citra stegano dengan metode modifikasi LSB. Panjang
% pesan (L) telah disimpan di dalam citra stegano.
% Asumsi: jumlah bit untuk merepresentasikan panjang pesan maksimal 30 bit.
% Input: Citra stegano (berwarna)
% Output: pesan (teks)
% Diprogram oleh Rinaldi Munir, @Informatika STEI-ITB

clear all;

% Baca citra stegano
namaFile = input('Ketikkan nama file citra stego: ', 's');
stegoImage = imread(namaFile);
imshow(namaFile); title ('Citra stego ');

% Ekstraksi bit-bit panjang pesan terlebih dahulu
% Diperlukan 30 byte = 10 pixel x 3 byte pada baris pertama citra stegano
idx = 1;
for col = 1:10
    for i = 1:3
        nL(idx) = bitget(stegoImage(1,col, i), 1);
        idx = idx + 1;
    end
end
end
```

```

% Konversi biner ke desimal
L = 0;
for idx = 1:30
    L = L + double(nL(idx)) * (2^(30-idx));
end

% Ekstraksi bit-bit pesan ke dalam vektor biner, bit-bit panjang pesan
% ikut dibaca kembali dari awal
m = L*8 + 30; % panjang bit pesan yang harus dibaca
disp('Ekstraksi pesan....');
idx = 1;
for row = 1 : size(stegoImage,1)
    for col = 1 : size(stegoImage,2)
        for i = 1 : 3
            if idx <= m
                vektorBiner(idx) = bitget(stegoImage(row,col, i), 1);
                idx = idx + 1;
            end
        end
    end
end
end
end

```

```

% Ambil hanya bit-bit pesan
vektorBiner = vektorBiner(31:m);

% Ubah vektor biner ke desimal
for i = 1 : L
    bin(i,:) = vektorBiner((i-1)*8 + 1 : i*8);
    dec(i) = bin2dec(num2str(bin(i,:)));
end
pesan = char(dec);

%Tampilkan pesan
disp('Pesan yang diekstraksi:');
disp(pesan)

```

Running program:

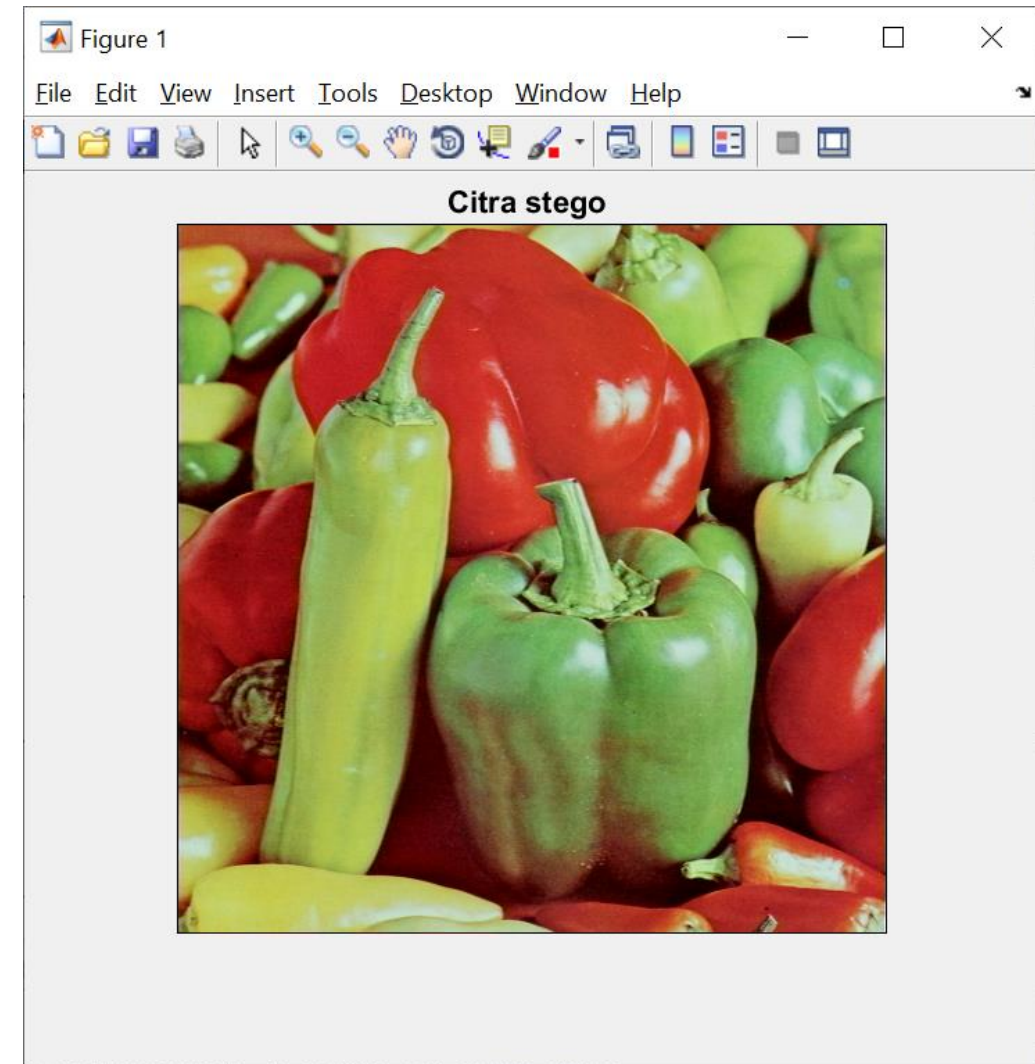
```
>> LSB_ekstrak
```

Ketikkan nama file citra stegano: output.bmp

Ekstraksi pesan....

Pesan yang diekstraksi:

Mari kita jaga negara kesatuan Republik Indonesia tercinta. Oke?



Program Steganografi dengan Python

- Kode program Python berikut ini menyisipkan pesan dengan metode LSB.

A. Program penyisipan pesan

- Input:
 1. Citra berwarna (RGB)
 2. Pesan (diketik langsung dari keyboard)
- Output: citra stego (stego image)

B. Program ekstraksi pesan

- Input: Citra stego
- Output: pesan

- Kode program Python tersebut dimodifikasi dari sumber ini:

<https://medium.com/swlh/lsb-image-steganography-using-python-2bbb2c69a2>

```
import numpy as np
from PIL import Image
```

```
def EmbeddingPesan(cover, pesan, stego):

    # Baca citra cover, ekstrak pixel-pixel sebagai sebuah larik (array)
    citra = Image.open(cover, 'r')
    citra.show()
    lebar, tinggi = citra.size
    larik_pixel = np.array(list(citra.getdata()))
    if citra.mode == 'RGB':
        n = 3

    total_pixel = larik_pixel.size//n      #jumlah pixel di dalam citra
    pesan = pesan + "stego"      #tambahkan delimiter untuk menandai akhir pesan

    #ubah pesan ke dalam biner
    pesan_biner = ''.join([format(ord(i), "08b") for i in pesan])
    jumlah_pixel_embed = len(pesan_biner)

    # periksa apakah mencukupi untuk menyisipkan pesan
    if jumlah_pixel_embed > total_pixel:
        print("ERROR: Ukuran citra tidak mencukupi untuk penyembunyian pesan")
    else:
```

```
# Sisipkan bit-bit pesan pada LSB setiap byte pixel
index=0
for p in range(total_pixel):
    for q in range(0, 2):
        if index < jumlah_pixel_embed:
            larik_pixel[p][q] = int(bin(larik_pixel[p][q])[2:9] + pesan_biner[index], 2)
            index = index + 1

larik_pixel = larik_pixel.reshape(tinggi, lebar, n)
stego_image = Image.fromarray(larik_pixel.astype('uint8'), citra.mode)
stego_image.save(stego)
print("Penyisipan pesan ke dalam citra berhasil")
```

```

def EkstraksiPesan(stego):
    # Baca citra stego, ekstrak pixel-pixel sebagai larik (array)
    citra = Image.open(stego, 'r')
    citra.show()
    larik_pixel = np.array(list(citra.getdata()))
    if citra.mode == 'RGB':
        n = 3

    total_pixel = larik_pixel.size//n          #jumlah pixel di dalam citra

    #Ekstraksi bit-bit pesan dari bit LSB setiap byte pixel
    bit_pesan = ""
    for p in range(total_pixel):
        for q in range(0, 2):
            bit_pesan += (bin(larik_pixel[p][q])[2:][-1])    #ambil bit LSB

    # Kelompokkan setiap 8-bit dari bit_pesan, simpan sebagai list
    bit_pesan = [bit_pesan[i:i+8] for i in range(0, len(bit_pesan), 8)]

```

```
# Ubah 8-bit pesan ke dalam setiap karakter
pesan = ""
for i in range(len(bit_pesan)):
    if pesan[-5:] == "stego":    # ketemu delimiter
        break
    else:
        pesan = pesan + chr(int(bit_pesan[i], 2))

if "stego" in pesan:
    print("Pesan yang diekstraksi:", pesan[:-5])
else:
    print("Tidak ada pesan tersembunyi di dalam citra")
```

```

def ProgramSteganografi():
    print("-- Steganografi pada Citra Digital --")
    print("1: Penyisipan pesan")
    print("2: Ekstraksi pesan")

    pilih = input()
    if pilih == '1':
        print("Masukkan nama citra cover (beserta path-nya)")
        cover = input()
        print("Ketikkan pesan yang akan disisipkan ke dalam citra")
        pesan = input()
        print("Masukkan nama citra output (stego image) beserta path-nya")
        stego = input()
        print("Penyisipan pesan...")
        EmbeddingPesan(cover, pesan, stego)
    elif pilih == '2':
        print("Masukkan nama citra stego beserta path-nya")
        stego = input()
        print("Ekstraksi pesan...")
        EkstraksiPesan(stego)
    else:
        print("Pilihan salah")

```

```
In [83]: ProgramSteganografi()  
  
-- Steganografi pada Citra Digital --  
1: Penyisipan pesan  
2: Ekstraksi pesan  
1  
Masukkan nama citra cover (beserta path-nya  
D:/S2 Rekayasa Pertahanan Siber Unhan/peppers.bmp  
Ketikkan pesan yang akan disisipkan ke dalam citra  
Belajar steganografi di Unhan gaesss.....  
Masukkan nama citra output (stego image) beserta path-nya  
D:/S2 Rekayasa Pertahanan Siber Unhan/output.bmp  
Penyisipan pesan...  
Penyisipan pesan ke dalam citra berhasil
```

```
In [84]: ProgramSteganografi()  
  
-- Steganografi pada Citra Digital --  
1: Penyisipan pesan  
2: Ekstraksi pesan  
2  
Masukkan nama citra stego beserta path-nya  
D:/S2 Rekayasa Pertahanan Siber Unhan/output.bmp  
Ekstraksi pesan...  
Pesan yang diekstraksi: Belajar steganografi di Unhan gaesss.....
```

NoStega: Noiseless Steganography

- Teknik baru steganografi, ditemukan oleh Desoky (2012)
- Tidak membutuhkan *cover* untuk menyembunyikan pesan
- Latar belakang: penyembunyian pesan di dalam *cover* dapat membuat kualitas *cover* menjadi terdegradasi ==> dapat diserang secara steganalisis untuk menemukan *embedded message*
- *NoStega* melakukan kamuflase dengan cara menyembunyikan pesan dalam bentuk *cover* yang terlihat alami sehingga tidak menimbulkan kecurigaan.
- *NoStega* menggunakan berbagai materi untuk melakukan kamuflase seperti grafik, email, game, catatan, dan lain-lain.

GraphStega (Graph Steganography)

- Salah satu teknik di dalam *NoStega*
- Melakukan kamuflase dengan cara mengubah pesan menjadi plot pada grafik.
- Contoh: pesan rahasia “*Use my secret key*”.
Ubah pesan ke dalam biner:

0101010101110011011001010010000001101101011110010
01000000111001101100101011000110111001001100101011
1010000100000011010110110010101111001

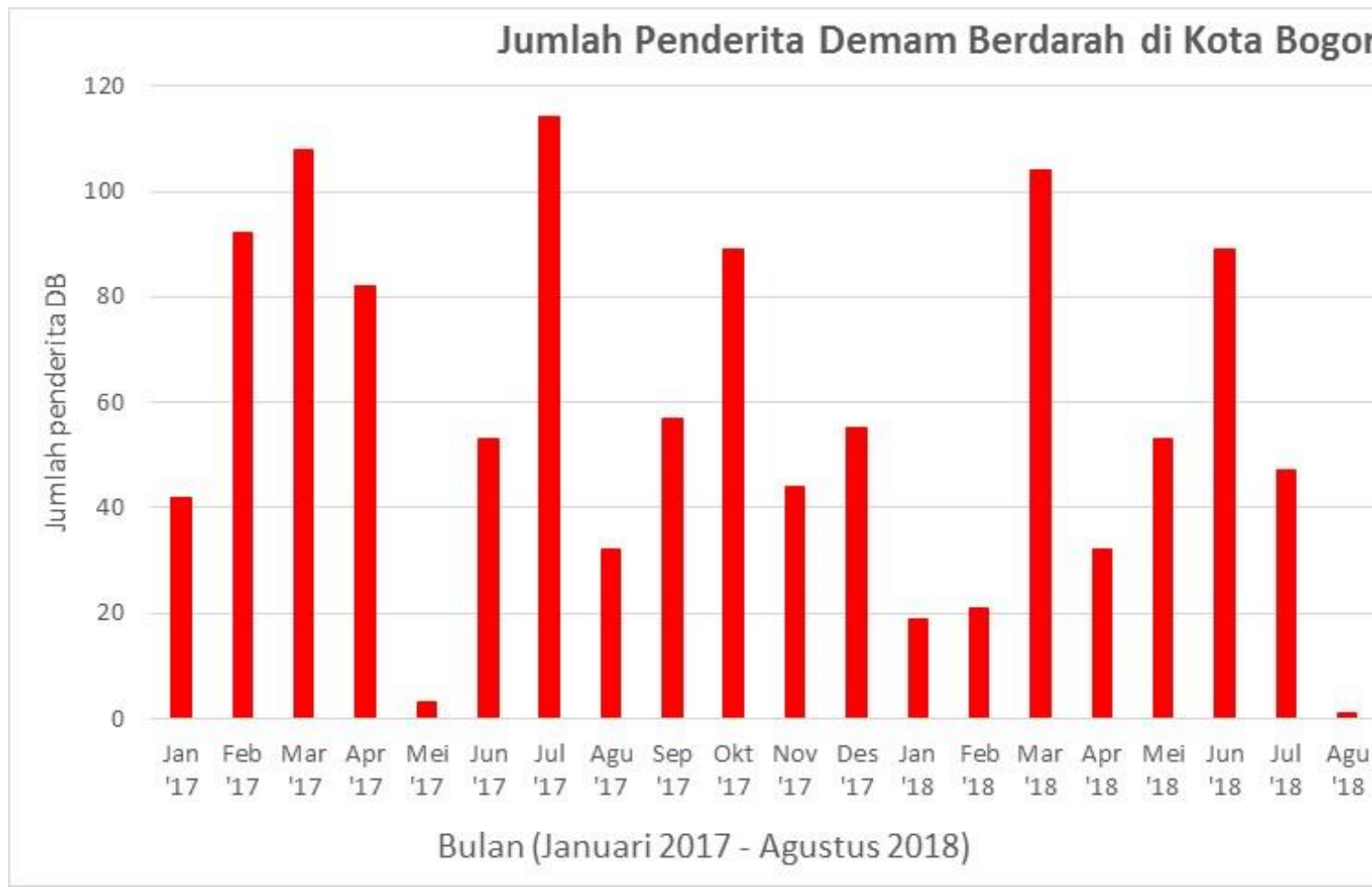
- Selanjutnya, kelompokkan menjadi kelompok-kelompok 7 bit:

0101010 1011100 1101100 1010010 0000011 0110101 1110010
0100000 0111001 1011001 0101100 0110111 0010011 0010101
1101000 0100000 0110101 1011001 0101111 001

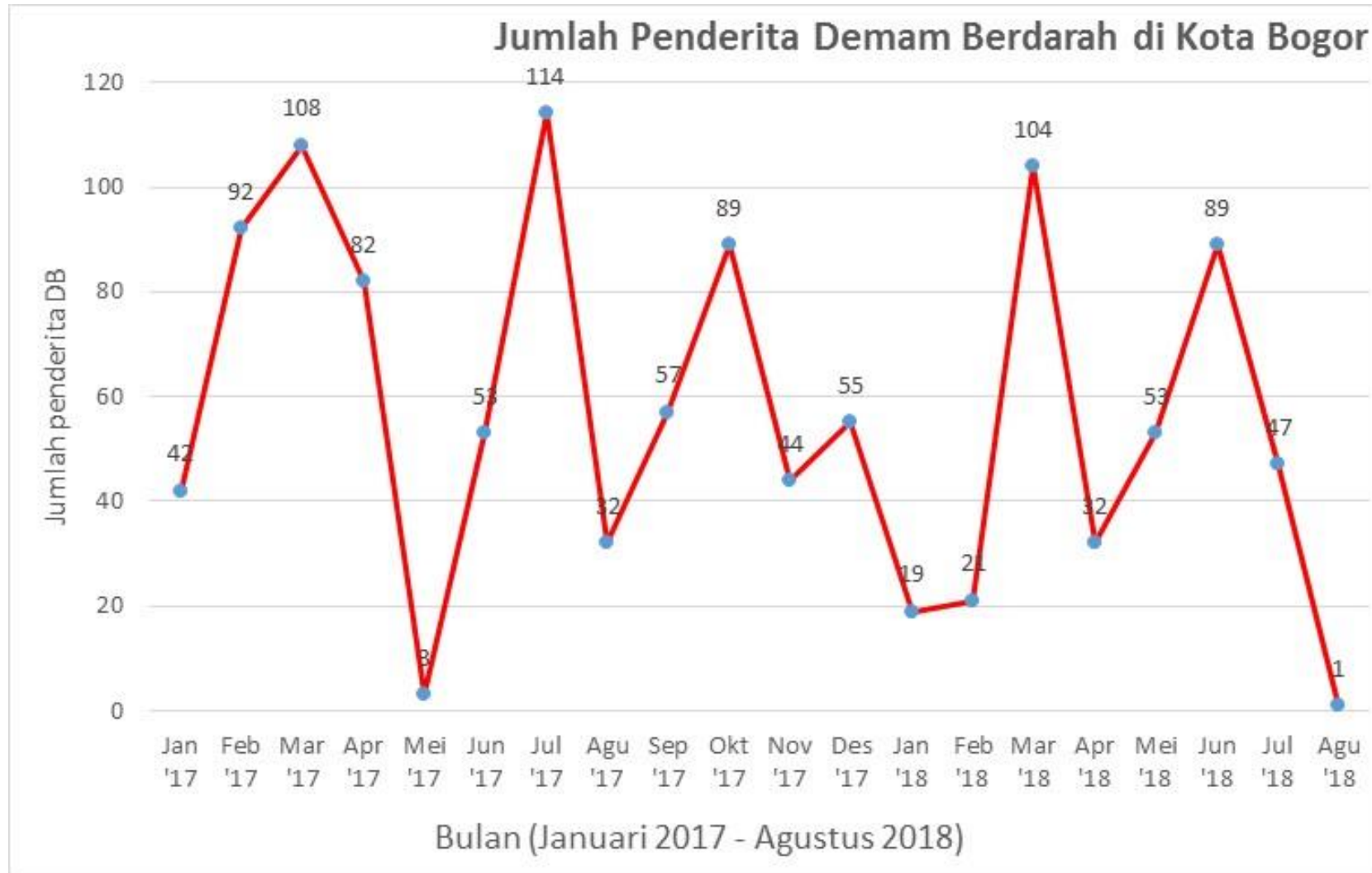
- Konversi kelompok-kelompok 7-bit di atas menjadi nilai desimal

42 92 108 82 3 53 114 32 57 89 44 55 19 21 104 32 53 89 47 1.

- Buatlah grafik dengan *Microsoft Excell* dengan menggunakan nilai-nilai desimal di atas
- Misalnya nilai-nilai tersebut menyatakan jumlah penderita demam berdarah selama 20 bulan di Kota Bogor (Januari 2017 – Agustus 2018).



Alternatif grafik lainnya:



Untuk membaca pesan rahasia, penerima membaca nilai-nilai pada grafik,

42 92 108 82 3 53 114 32 57 89 44 55 19 21 104 32 53 89 47 1

lalu mengubahnya ke dalam biner,

00101010 01011100 01101100 01010010 00000011 00110101 01110010
00100000 00111001 01011001 00101100 00110111 00010011 00010101
01101000 00100000 00110101 01011001 00101111 0001

Untuk setiap kelompok biner ambil 7-bit dari belakang (7-bit LSB),

0101010 1011100 1101100 1010010 0000011 0110101 1110010
0100000 0111001 1011001 0101100 0110111 0010011 0010101
1101000 0100000 0110101 1011001 0101111 001

Gabungkan semua kelompok bit menjadi satu,

0101010101110011011001010010000001101101011110010
01000000111001101100101011000110111001001100101011
1010000100000011010110110010101111001

Kemudian kelompokkan menjadi kelompok-kelompok 8-bit,

01010101 01110011 01100101 00100000 01101101 01111001
00100000 01110011 01100101 01100011 01110010 01100101
01110100 00100000 01101011 01100101 01111001

Kodekan setiap delapan bit tersebut menjadi karakter ASCII

Use my secret key

Pesan rahasia berhasil dibaca kembali!

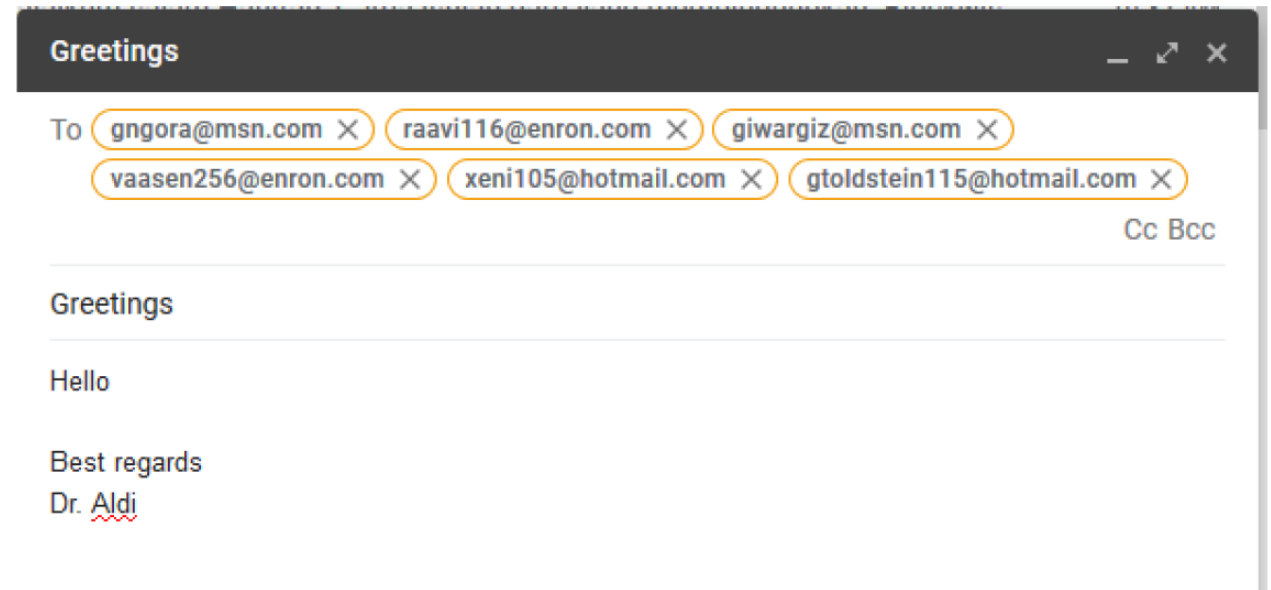
Teknik NoStega lainnya

3. Nc3	Nf6
4. exd5	exd5
5. Nf3	Bd6
6. Bd3	O-O
7. O-O	h6
8. Re1	Nc6
9. Nb5	Bb4
10. c3	Ba5
11. Na3	Bg4
12. Nc2	Qd7
13. b4	Bb6
14. h3	Bh5
15. Ne3	Rfe8
16. b5	Ne7
17. g4	Bg6
18. Ne5	Qc8
19. a4	c6
20. bxc6	bxc6
21. Ba3	Ne4
22. Qc2	Ng5
23. Bxe7	Rxe7
24. Bxg6	fxg6
25. Qxg6	Nxh3+
26. Kh2	Nf4
27. Qf5	Ne6
28. Ng2	Qc7



The Chessmaster recommends: Queen to d3

Chess-stega



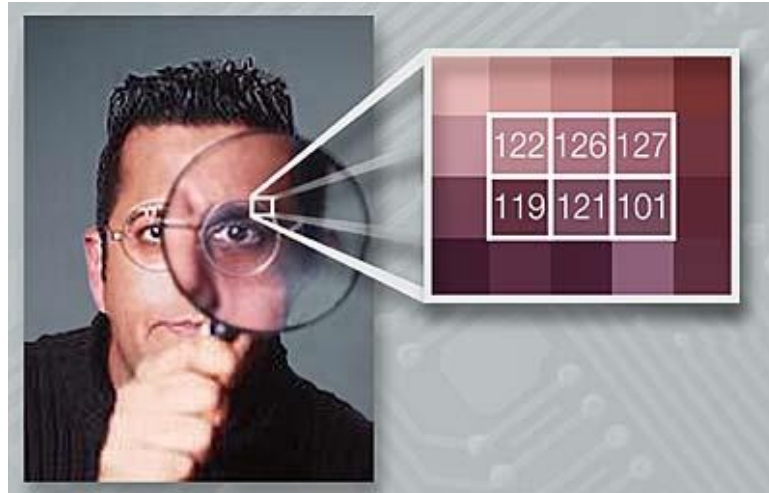
Head-stega

Pengantar Steganalisis

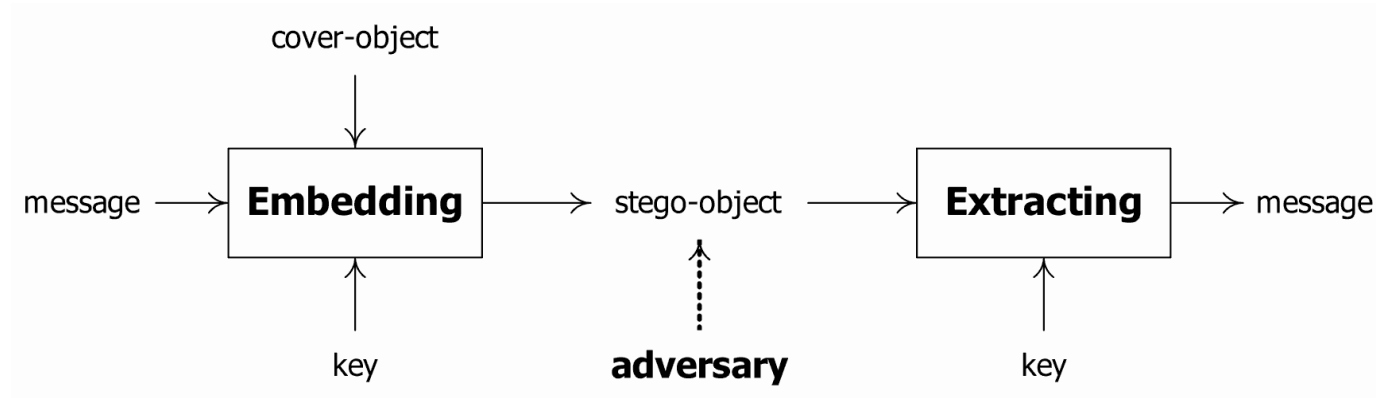
(Opsional)

Steganalysis

- Tujuan: menentukan apakah sebuah media *suspect* mengandung pesan tersembunyi



- Steganografi



- Steganalisis



*) Keterangan: 1 jika ada pesan tersembunyi, 0 jika tidak

Fakta: Gambar-gambar bertebaran di internet
(*website, social media, social networking*)

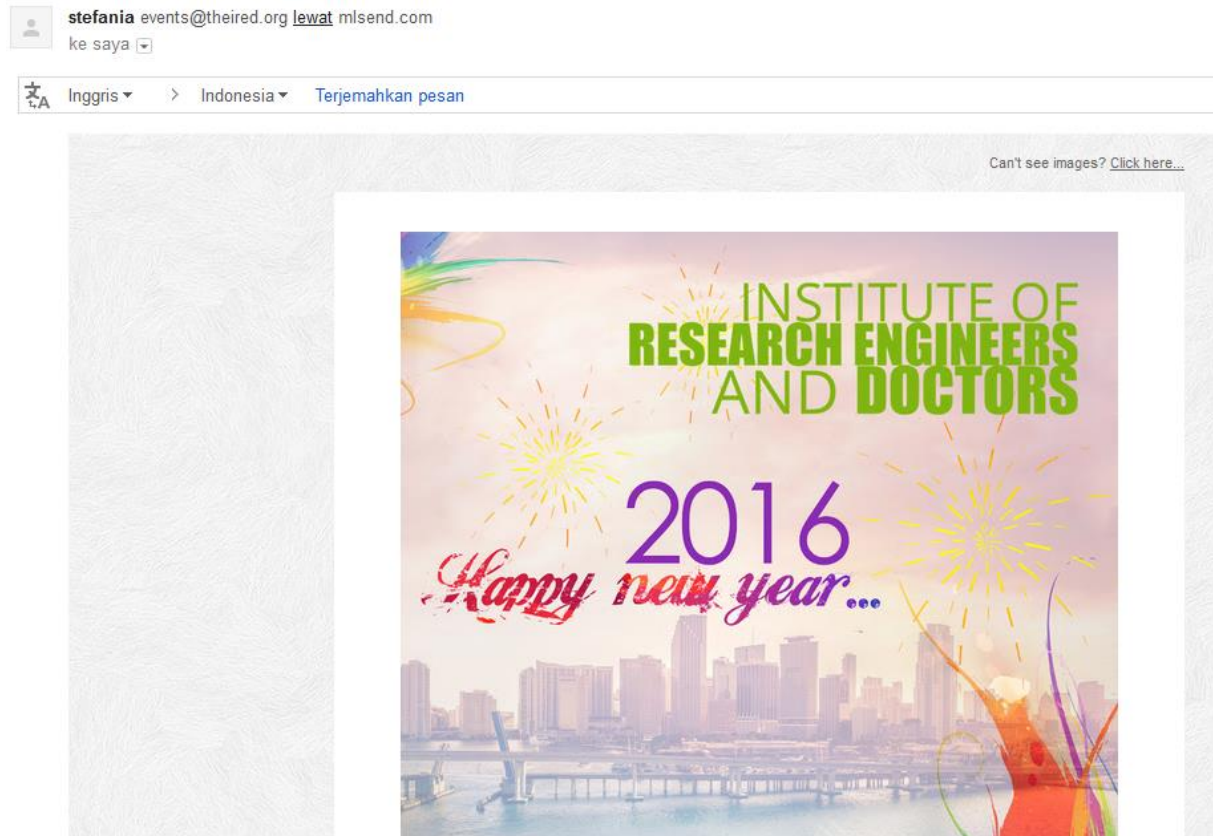


Namun, dibalik sebuah gambar dapat tersembunyi informasi rahasia

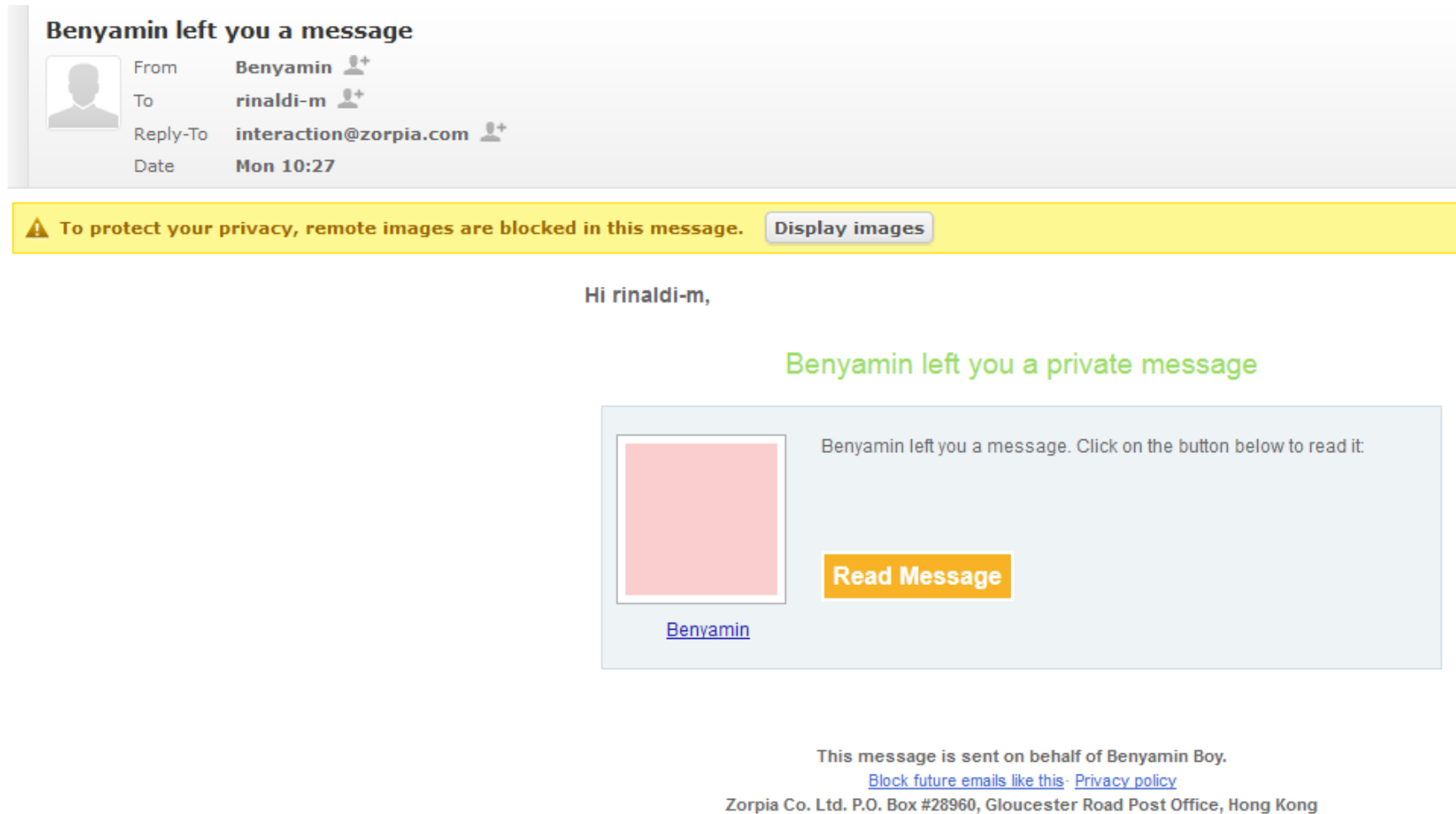


Informasi rahasia tersebut dapat berupa pesan biasa, pesan kejahatan, program jahat, bahkan virus komputer!

Pernah terima surel (*e-mail*) dari orang tak dikenal dan mengandung *file attachmet* berupa gambar seperti di bawah ini?



HATI-HATI!!!!!!!!!!



HATI-HATI! Jangan langsung klik jika anda tidak yakin!

Ingat kembali stegosploit!!!

How to Hack a Computer Using Just An Image

Monday, June 01, 2015 Swati Khandelwal

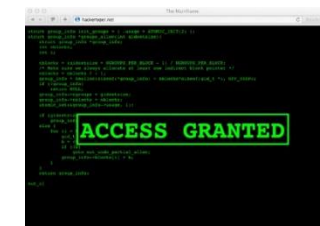
512 8.5K 12.8K 923 84 19.2K



Next time when someone sends you a photo of a cute cat or a hot chick than be careful before you [CLICK](#) on the image to view — it might hack your machine.

Yes, the normal looking images could hack your computers — thanks to a technique discovered by security researcher *Saumil Shah* from India.

Dubbed "*Stegosplit*," the technique lets hackers hide malicious code inside the pixels of an image, hiding a malware exploit in plain sight to infect target victims.



Just look at the image and you are HACKED!

<http://thehackernews.com/2015/06/Stegosplit-malware.html>

Rinaldi Munir/414031 Kriptografi dan Koding

- Steganalisis diperlukan di dalam *forensic image analysis*
- ***Forensic Image Analysis*** is the application of image science and domain expertise to interpret the content of an image and/or the image itself in legal matters.
- Subdisiplin dari *Forensic Image Analysis*:
 - (1) *Photogrammetry*
 - (2) *Photographic Comparison*
 - (3) *Content Analysis*
 - (4) *Image Authentication*

- Salah satu pekerjaan di dalam *content analysis* adalah mendeteksi apakah ada pesan tersembunyi di dalam sebuah gambar.
- Contoh sebuah skenario: Mr. Abdul, seorang investigator forensik, diminta Lab Forensik Polri untuk menginvestigasi sebuah *cybercrime* berupa foto. Sebagai investigator forensik yang ahli, dia menganalisis foto untuk menemukan pesan tersembunyi di dalamnya dengan kakas steganalisis.



- Tujuan utama steganalisis adalah untuk membedakan apakah sebuah media mengandung pesan rahasia atau tidak.
- Steganalisis dianggap berhasil jika ia dapat menentukan apakah sebuah media mengandung pesan tersembunyi dengan peluang lebih tinggi daripada menerka secara acak.
- Selain tujuan utama di atas, terdapat beberapa tujuan minor steganalisis:
 - menentukan panjang pesan
 - menentukan tipe algoritma penyisipan
 - kunci yang digunakan

Visual Attack

- Memanfaatkan indera penglihatan → inspeksi kerusakan pada gambar akibat penyisipan
- Ide dasar :



Metode Enhanced LSB

BLUE	GREEN	RED
1010010 <u>1</u>	1001110 <u>0</u>	1110011 <u>1</u>

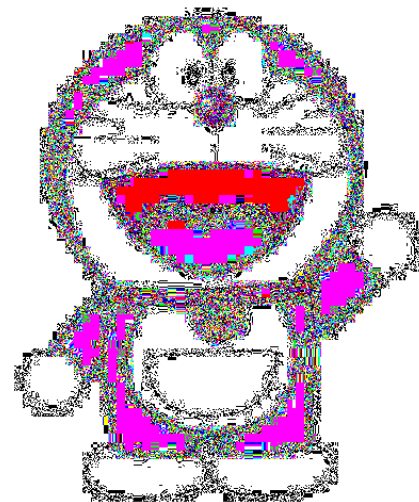


BLUE	GREEN	RED
<u>11111111</u>	<u>00000000</u>	<u>11111111</u>





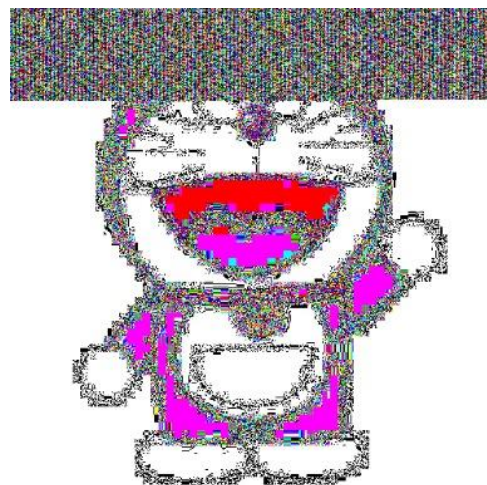
(a) Citra orisinal



(b) Citra hasil *enhanced LSB*



(c) Citra stego



(d) Citra hasil *enhanced LSB*

Teknik Steganalisis: *Visual Attack*

Artefak mencurigakan



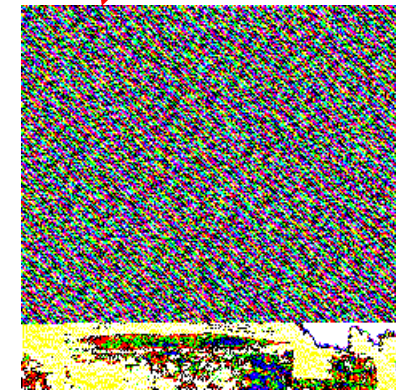
Gambar asli



Hasil penapisan
(asli)



Terdeteksi ada pesan



Terdeteksi ada pesan



Terdeteksi ada pesan



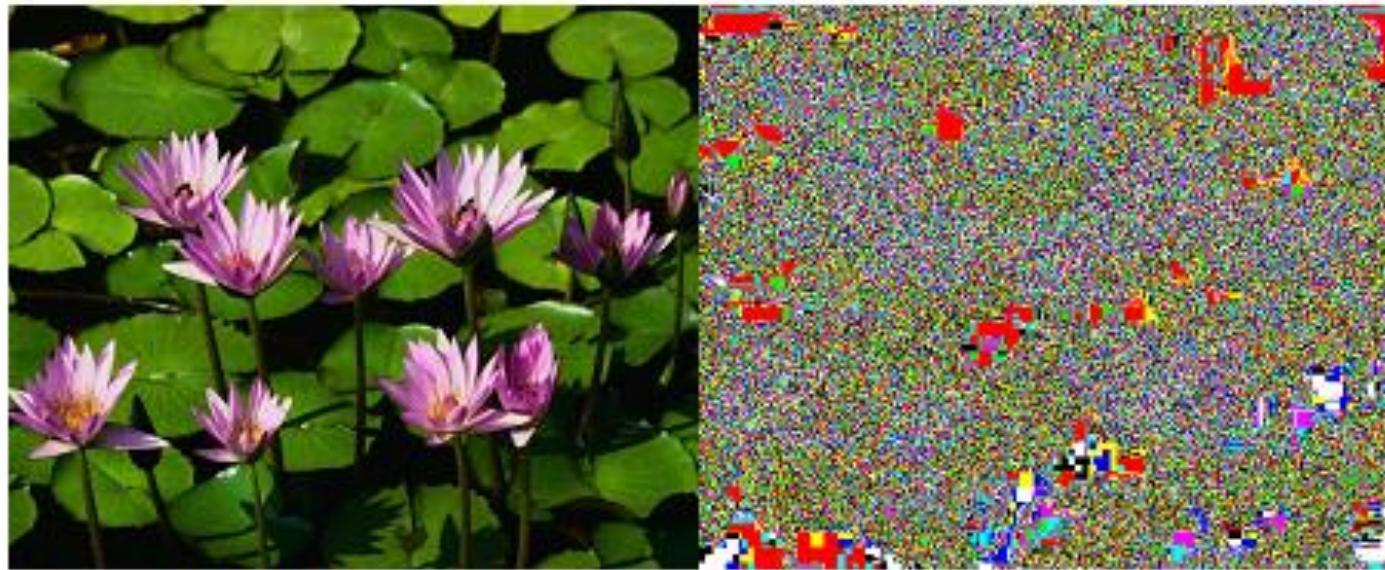
Terdeteksi ada pesan

Metode *enhanced-LSB* bagus untuk citra dengan kontras tinggi, yaitu citra yang memiliki warna latar yang jelas atau memiliki perbedaan warna yang kontras antara latar dengan gambar utama



Gambar III-1 Gambar yang mengandung pesan rahasia dan hasil *enhanced LSB*-nya [PAU07]

Untuk citra dengan kontras rendah (seperti citra hasil fotografi), metode *enhanced LSB* seringkali menyulitkan steganalisis. Karena steganalisis akan kesulitan membedakan antara gambar yang seharusnya muncul dengan pesan rahasia.



Gambar III-3 Gambar dengan kontras rendah dan hasil *enhanced LSB*-nya