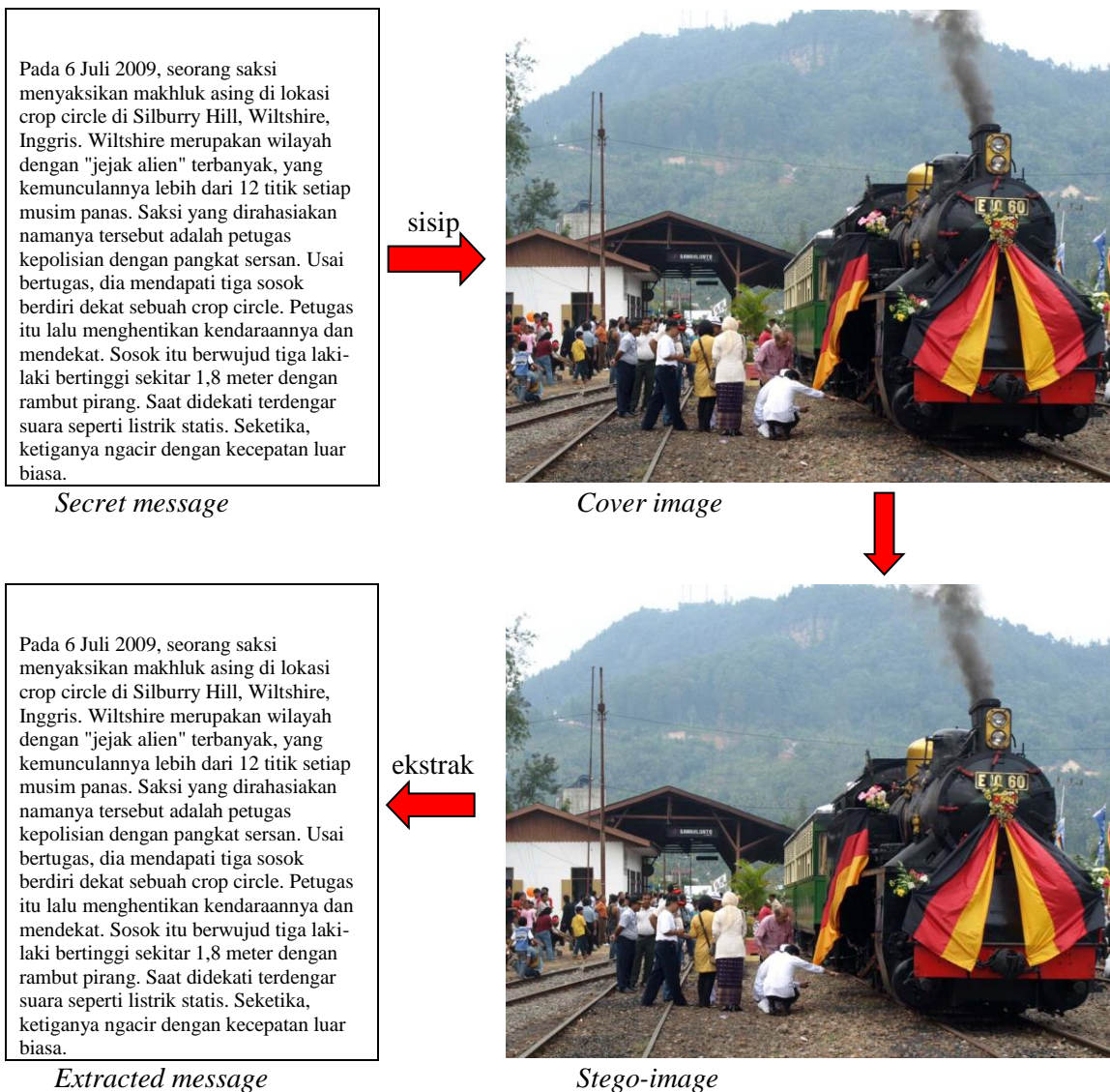


Tugas Besar I IF3058 Kriptografi
Sem. II Tahun 2010/2011

Penyembunyian Pesan di dalam Citra *Bitmap*

Selain dengan enkripsi, kerahasiaan pesan juga dapat diimplementasikan dengan steganografi. Pesan rahasia disimpan di dalam media digital seperti citra sedemikian sehingga keberadaan tidak dapat dideteksi. Penyembunyian pesan di dalam citra dilakukan sedemikian sehingga tidak merusak kualitas citra (Gambar 1). Algoritma steganografi sederhana pada citra digital adalah dengan algoritma modifikasi LSB. Nilai bit LSB pada *pixel-pixel* citra diganti dengan bit-bit pesan. Untuk meningkatkan keamanan, maka penyisipan pesan ke dalam *pixel-pixel* citra tidak dilakukan secara sekuensial, tetapi secara acak. Oleh karena itu, pembangkit bilangan acak dibutuhkan untuk membangkitkan posisi *pixel*. Pembangkit bilangan acak ini tergantung pada kunci (yang akan menjadi *seed* atau nilai awal untuk memulai pembangkitan). Pada proses ekstraksi pesan, kunci ini dibutuhkan kembali untuk membangkitkan bilangan acak yang sama.



Gambar 1. Penyisipan dan ekstraksi pesan rahasia pada citra *bitmap*

Pada prakteknya, sebelum disisipkan, pesan dienkripsi terlebih dahulu dengan sebuah algoritma enkripsi. Karena anda baru belajar algoritma kriptografi klasik, maka algoritma enkripsi yang digunakan adalah *Vigenere Cipher* seperti yang pernah dikerjakan pada Tupil 1.

Selain itu, untuk meningkatkan kapasitas (*payload*) pesan yang dapat disisipkan, maka bit LSB yang dimodifikasi ada dua pilihan: 1 bit LSB atau 2 bit LSB.

Dalam tugas besar ini, anda diminta membuat program steganografi pada citra *bitmap* (*berwarna* atau *grayscale*). Anda harus memahami format *file* citra *bitmap* agar tahu cara memanipulasi bit LSB-nya. Pesan yang disisipkan adalah sembarang *file* dengan ukuran yang tidak melebihi kapasitas penyisipan (*payload*). Kapasitas penyisipan dihitung sebelum proses penyisipan.

Spesifikasi program:

1. Program menerima masukan berupa citra digital dengan format bitmap (.BMP), nama file pesan, dan kunci steganografi.
2. Pengguna dapat memilih ukuran bit LSB yang digunakan (1 bit atau 2 bit)
3. Pengguna dapat memilih apakah pesan dienkripsi atau tidak dienkripsi sebelum disisipkan.
4. Pengguna memasukkan sebuah kata kunci (maksimal 25 karakter) yang berfungsi dua: sebagai kunci enkripsi pada *Vigenere Cipher* dan sebagai kunci (*seed*) pembangkitan bilangan acak.
 Contoh: Kunci = 'STEGANO', kunci ini langsung dijadikan sebagai kunci enkripsi.
 Untuk *seed* berupa bilangan acak (yang umumnya berupa integer/real), maka nilai-nilai integer dari string 'STEGANO' dijumlahkan, yaitu $\text{Int}('S') + \text{Int}('T') + \text{Int}('E') + \text{Int}('G') + \text{Int}('A') + \text{Int}('N') + \text{Int}('O') = \dots$
 Atau, hanya mengambil sebagian huruf dari STEGANO, misalnya karakter pada posisi ganjil saja, yaitu $\text{Int}('S') + \text{Int}('E') + \text{Int}('A') + \text{Int}('O') = \dots$, atau terserah cara yang anda gunakan.
5. Jangan menyisipkan kunci di dalam file citra.
6. Program menolak menyisipkan pesan jika ukuran file pesan melebihi *payload*.
7. Program dapat menyimpan *stego-image* (citra yang sudah disisipi pesan)..
8. Program dapat mengekstraksi pesan utuh seperti sedia kala dan menyimpannya sebagai file dengan nama lain (*save as*).
9. Agar format file hasil ekstraksi diketahui, maka properti file seperti ekstensi (.exe, .doc, .pdf, dll), sebaiknya juga disimpan (atau nama file asli juga disimpan. agar diketahui formatnya, sehingga ketika di-*save as* yang muncul adalah nama file asli tersebut, lalu pengguna dapat menggantinya dengan nama lain). Penyimpanan nama file (dan properti lainnya) tentu akan mengurangi kapasitas pesan yang dapat disimpan.
10. Program dapat menampilkan (*view*) citra asli dan citra stegano dalam dua jendela berbeda.
11. Program dapat menampilkan ukuran kualitas citra hasil steganografi dengan *PSNR* (*Peak Signal- to-Noise Ratio*). *PSNR* adalah metrik yang umum digunakan untuk mengukur kualitas citra. *PSNR* dihitung dengan rumus:

$$PSNR = 20 \times \log_{10} \left(\frac{256}{rms} \right) \tag{II.13}$$

yang dalam hal ini 256 adalah nilai sinyal terbesar (pada citra dengan 256 derajat keabuan), dan *rms* (*root mean square*) adalah akar pangkat dua dari kuadrat selisih dua buah citra *I* dan \hat{I} yang berukuran $M \times N$:

$$rms = \sqrt{\frac{1}{MN} \sum_{i=1}^N \sum_{j=1}^M (I_{ij} - \hat{I}_{ij})^2}$$

Satuan *PSNR* adalah desibel (dB). *PSNR* menyatakan visibilitas derau di dalam citra. *PSNR* yang besar mengindikasikan nilai *rms* yang kecil; *rms* kecil berarti dua buah citra mempunyai sedikit

perbedaan. Dari praktek pengolahan citra, citra dengan PSNR > 30 masih dapat dianggap kualitasnya bagus, tetapi jika PSNR < 30 dikatakan kualitas citra sudah terdegradasi secara signifikan.

12. Fitur-fitur lainnya dipersilakan dibuat.

Prosedur Pengerjaan

1. Tugas dikerjakan secara berkelompok (1 kelompok @ 3 orang), dilarang *gabut*. Cantumkan pembagian tugas dengan jelas antara anggota kelompok.
2. Waktu pengumpulan tugas: paling lambat 25 Februari 2011 sebelum pukul 17.00 di Lab IRK). Terlambat menyerahkan tugas, nilai = 0.
3. Kakas pengembangan program bebas (Java, .NET, Delphi, Visual C, dll)
4. Yang diserahkan pada saat pengumpulan antara lain:
 - a. Disket atau CD yang berisi program sumber (*source code*), arsip siap eksekusi (*executable file*) (termasuk semua *.dll* jika ada), dan arsip-arsip uji (citra, file pesan).
 - b. Laporan yang memiliki sistematika sebagai berikut :
 - i. Teori singkat (steganografi, metode modifikasi LSB, citra bitmap, dll).
 - ii. Perancangan dan Implementasi, termasuk : rancangan program.
 - iii. Pengujian program dan analisis hasil. Uji program dengan bermacam-macam citra bitmap dan jenis file pesan.
 - iv. Kesimpulan dari hasil implementasi.
 - v. Tampilkan foto anda bertiga di *cover* laporan sebagai pengganti logo gajah.

Laporan dikumpulkan dalam bentuk *hard copy* dan *soft copy* dengan format *.pdf .

4. Penilaian tugas dilakukan pada saat demo.