Tugas Besar I IF3058 Kriptografi
Sem. II Tahun 2013/2014

Penyembunyian Pesan di dalam Berkas Citra *Bitmap* atau Berkas Audio WAV atau Berkas Video AVI

Di dalam tugas besar ini, anda boleh memilih salah satu:

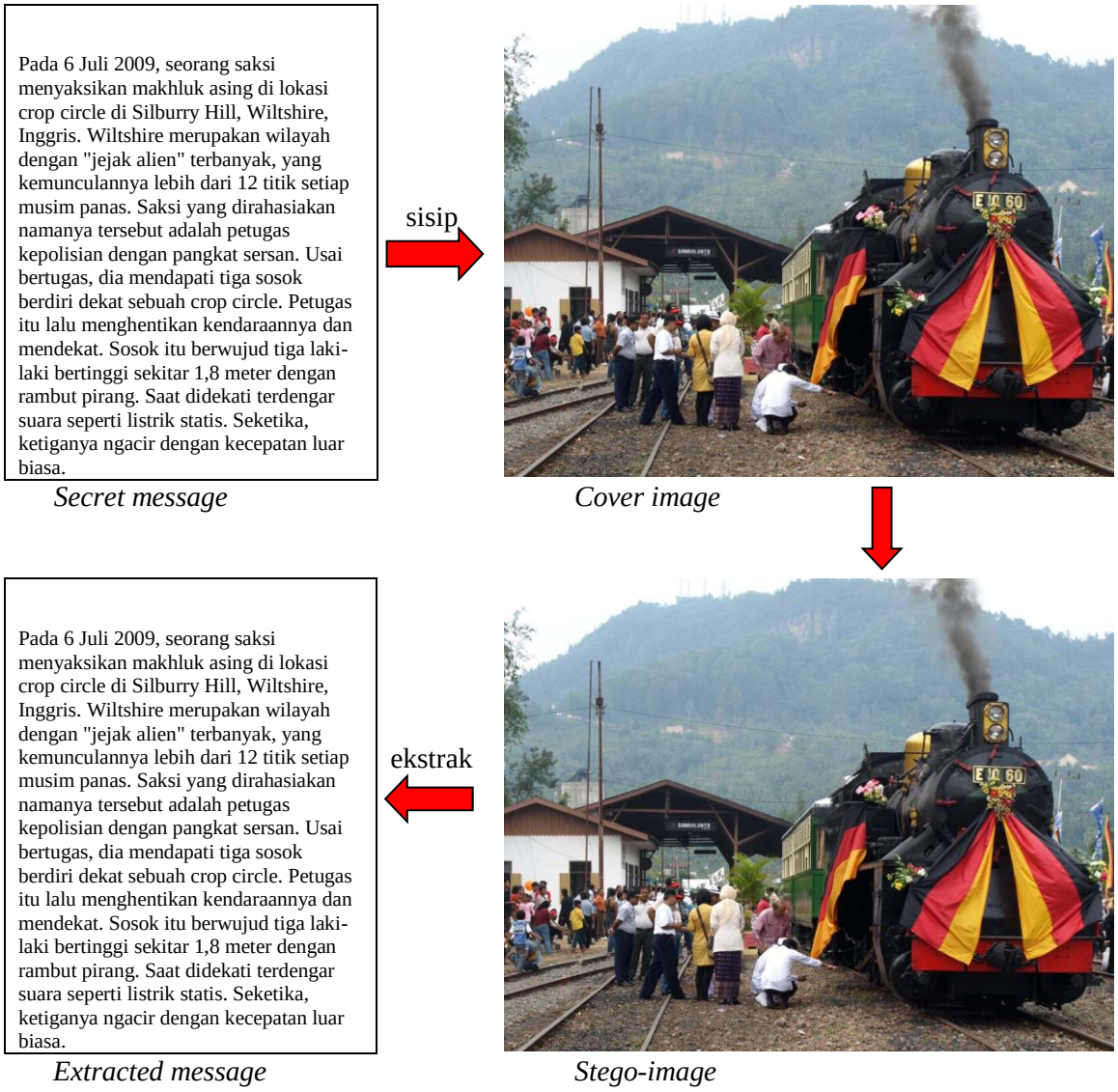
(a) Penyembunyian pesan di dalam berkas citra *bitmap*
(b) Penyembunyian pesan di dalam berkas citra audio WAV (bonus nilai 10)
(c) Penyembunyian pesan di dalam berkas video AVI (bonus nilai 20).

(a) Penyembunyian Pesan di dalam Berkas Citra Bitmap

Selain dengan enkripsi, kerahasisaan 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 (lihat Gambar 1).

Pada praktekmya, 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* (*extended* untuk alfabet 256 karakter) seperti yang pernah dikerjakan pada Tucil 1.

Selain itu, untuk meningkatkan kapasitas (*payload*) pesan yang dapat disisipkan, maka bit LSB yang dimodifikasi ada dua pilihan: 1 bit LSB, 2 bit LSB, atau 3 bit LSB.



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

Dalam tugas besar ini, anda diminta membuat program steganografi pada citra *bitmap* (*berwarna* atau *grayscale*). Format citra bitmap adalah format citra yang tidak terkompresi sehingga ukurannya lebih besar dibandingkan format yang terkompresi (misalnay JPEG). 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.

Spesifkasi 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, 2 bit, atau 3 bit)
3. Pengguna dapat memilih apakah pesan dienkripsi atau tidak dienkripsi sebeleum 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 Int('S') + Int('T') + Int('E') + Int('G') + Int('A') + Int('N') + Int('O') = ...
   Atau, hanya mengambil sebagian huruf dari STEGANO, misalnya karakter pada posisi ganjil saja, yaitu Int('S') + Int('E') + Int('A') + Int('O') = ..., 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 sediakala 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 a*s 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:

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

    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 $M \times N$:

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

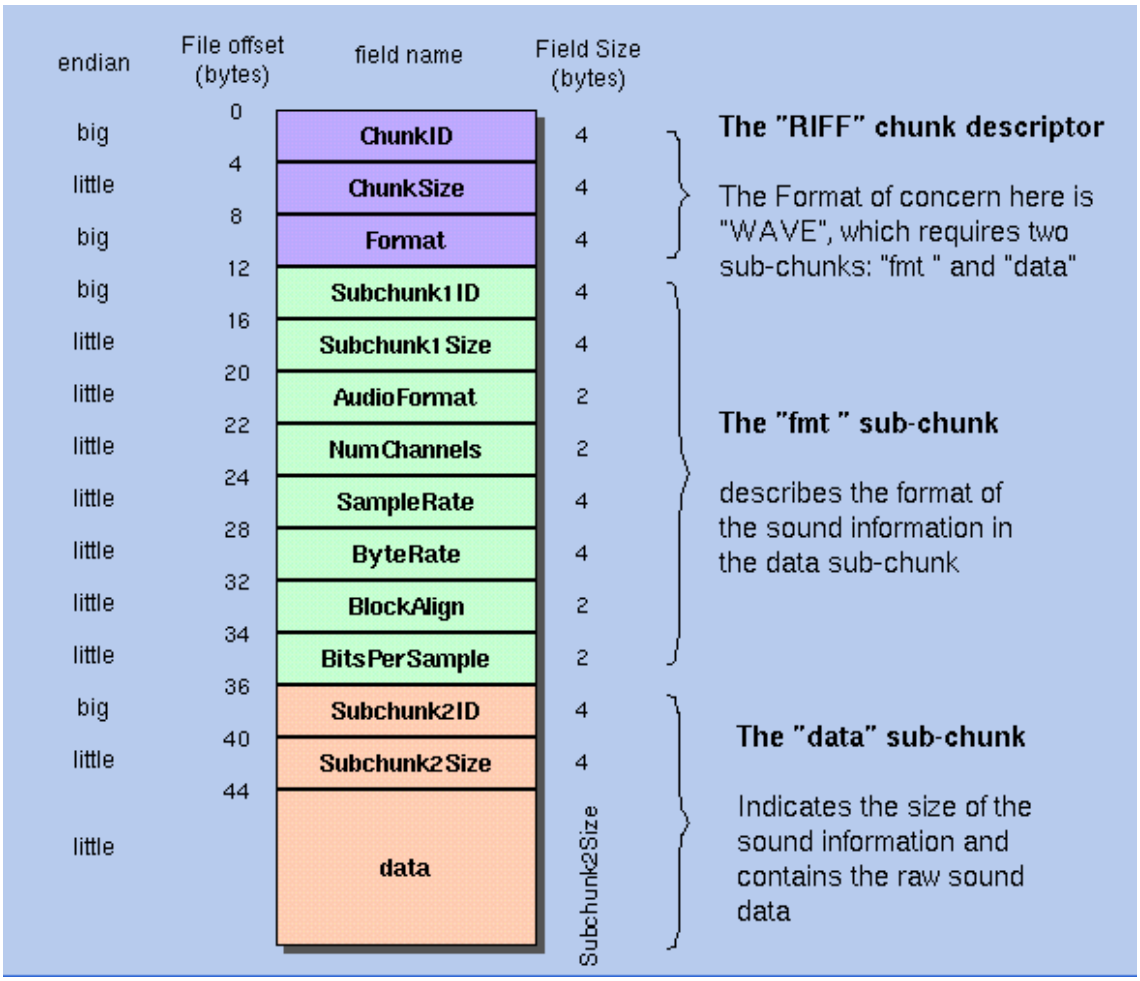
    Satuan *PSNR* adalah desibel (dB). *PSNR* menyatakan visibilitas derau di dalam citra. *PSNR* yang besar mengindikasikan nilai *rms* yang kecil; *rms* kecil berarti dua buah citra mempunyai sedikit perbedaan. Dari praktek pengolahan citra, citra dengan PSNR > 30 masih dapat dianggap kualitasnya bagus, tetapi jika PSNR < 30 dikatakan kualitas citra sudah terdegradasi secara signifikan.
12. Citra uji yang digunakan sedikitnya berupa citra homogen (misalnya gambar langit biru) dan citra heterogen (misalnya gambar bunga-bunga di taman).
13. Fitur-fitur lainnya dipersilakan dibuat.

(b) Penyembunyian Pesan di dalam Berkas Audio WAV

Seperti dikutip dari sini, https://ccrma.stanford.edu/courses/422/projects/WaveFormat/, format file WAVE atau WAV adalah bagian dari spesifikasi RIFF Microsoft untuk penyimpanan file multimedia. Sebuah file RIFF dimulai dengan sebuah header file diikuti dengan urutan dari potongan data. Sebuah file WAVE sering hanya file RIFF dengan sepotong tunggal "WAVE"

yang terdiri dari dua sub-potongan - sebuah "fmt" potongan menentukan format data dan "data" potongan yang berisi data audio yang sebenarnya. Umumnya data audio di dalam format WAV adalah dalam bentuk tidak terkompresi.

Format Data WAV:



Gambar 2. Format berkas WAV

Penyisipan pesan ke dalam berkas audio WAV dengan metode modifikasi LSB pada prinsipnya sama seperti pada citra, yaitu bit pesan disisipkan pada bit LSB dari *byte* audio. Perbedaannya adalah perubahan bit pada audio mempunyai efek lebih peka dibandingkan pada gambar. Perubahan bit LSB terasa merusak kualitas suara pada musik lembut, misalnya.

Spesifkasi program:
Spesifikasi program sama seperti pada citra *bitmap* dengan penambahan/perubahan sebagai berikut:

1. Program menyediakan fitur *player* sehingga berkas audio WAV dapat dimainkan.
2. Berkas WAV dapat diperoleh dengan *converter* dari MP3 ke WAV (cari *free software* nya di internet).
3. PSNR pada berkas audio dapat dihitung dengan rumus

$$PSNR = 10 \log_{10}\left(\frac{P_1^2}{P_1^2 + P_0^2 - 2P_1P_0}\right)$$

yang dalam hal ini P_0 dan P_1 adalah kekuatan sinyal berkas audio sebelum dan seduah penyembunyian pesan. Nilai minimal PSNR adalah 30 DB (jika kurang dari 30 DB berarti sinyal audionya mengalami kerusakan yang berarti).

(c) Penyembunyian Pesan di dalam Berkas Video AVI

Selain dengan enkripsi, keamanan pesan juga dapat menggunakan teknik steganografi. Pesan rahasia disimpan di dalam data multimedia seperti teks, citra, audio, dan video sedemikian sehingga keberadaan pesan tidak dapat dideteksi. Pada tugas besar kali ini pesan disembunyikan di dalam video digital dengan format AVI (*Audio Video Interleave*) (Baca ini: http://en.wikipedia.org/wiki/Audio_Video_Interleave). Video mempunyai kapasitas penyembunyian data yang lebih besar dibandingkan dengan citra tunggal, sebab video disusun oleh banyak *frame* (1 *frame* = 1 *image*). Video dengan format AVI adalah jenis format yang tidak dikompresi sehingga metode modifikasi LSB dapat langsung mengubah LSB setiap *pixel* pada setiap *frame*. Karena video digital disusun oleh *layer frame* dan *layer* audio, maka penyembunyian pesan biasanya dilakukan pada *layer frame* saja. Gambar 1 adalah sebuah *frame* video, gambar yang kiri adalah *frame* sebelum disisipi pesan, dan gambar yang kanan adalah frame yang sudah disisipi pesan.

Gambar 3. Kiri: *frame* yang belum disisipi pesan; kanan: *frame* yang sudah disisipi pesan

Untuk meningkatkan keamanan, maka penyisipan pesan di dalam setiap frame tidak dilakukan secara sekuensial *pada pixel-pixel*-nya, tetapi secara acak. Oleh karena itu, pembangkit bilangan acak dibutuhkan untuk membangkitkan posisi *pixel* di dalam setiap *frame*. Selain itu, karena ada banyak *frame* di dalam sebuah video, maka *frame* yang disisipi pesan pun tidak perlu disisipi diatur secara sekuensial, tetapi juga dapat dipilih secara acak. Pembangkit bilangan acak tergantung pada umpan (*seed*) yang diberikan oleh pengguna, dan umpan tersebut dianggap sebagai *stego-key*. Pada proses ekstraksi pesan, stego-key dibutuhkan kembali untuk membangkitkan bilangan acak yang sama.

Steganografi dapat dikombinasikan dengan kriptografi untuk membuat keamanan pesan menjadi berlapis. Sebelum disisipkan ke dalam video, pesan dienkripsi terlebih dahulu dengan sebuah algoritma enkripsi. Karena anda baru belajar algoritma kriptografi klasik, maka algoritma

enkripsi yang digunakan adalah *Vigenere Cipher* (*extended* untuk alfabet 256 karakter) seperti yang pernah dikerjakan pada Tucil 1. Pesan yang disisipkan adalah sembarang tipe *file* dengan ukuran yang tidak melebihi kapasitas penyisipan (*payload*). Kapasitas penyisipan dapat ditentukan sebelum proses penyisipan pesan.

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.

Spesifkasi program:

1. Program menerima masukan berupa video digital dengan format AVI (atau format tak terkompresi lainnya, jika ada), nama file pesan, dan kunci-stego.
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 Int('S') + Int('T') + Int('E') + Int('G') + Int('A') + Int('N') + Int('O') = ...
   Atau, hanya mengambil sebagian huruf dari STEGANO, misalnya karakter pada posisi ganjil saja, yaitu Int('S') + Int('E') + Int('A') + Int('O') = ..., atau terserah cara yang anda gunakan.
5. JANGAN menyisipkan kunci di dalam file video.
6. Program menolak menyisipkan pesan jika ukuran file pesan melebihi *payload*.
7. Program dapat menyimpan *stego-video* (video yang sudah disisipi pesan) dengan nama berbeda (*Save as*)
8. Program dapat mengekstraksi pesan utuh seperti sediakala 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 a*s 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 memainkan (*playback*) video asli dan stego-video melalui sebuah video *player* yang dipanggil dari dalam program.
11. Program dapat menampilkan ukuran kualitas video hasil steganografi dengan *PSNR* (*Peak Signal- to-Noise Ratio*). *PSNR* adalah metrik yang umum digunakan untuk mengukur kualitas sebuah citra. *PSNR* dihitung dengan rumus:

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

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. Oleh karena video terdiri dari banyak *frame*, maka PSNR video adalah rata-rata PSNR dari seluruh *frame* yang disisipkan pesan saja.

12. Fitur-fitur lainnya dipersilakan dibuat.

Prosedur Pengerjaan

1. Tugas dikerjakan secara berkelompok (1 kelompok @ 3 orang), dilarang *gabut*, dilarang menggunakan kode program orang lain. Cantumkan pembagian tugas dengan jelas antara anggota kelompok.
2. Waktu pengumpulan tugas: paling lambat 7 Maret 2014 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, audio, dll).
        ii. Perancangan dan Implementasi, termasuk : rancangan program.
        iii. Pengujian program dan analisis hasil. Uji program dengan bermacam-macam citra bitmap/berkas WAV 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.