

**Tugas Besar I IF3058 Kriptografi  
Sem. II Tahun 2012/2013**

**Penyembunyian Pesan di dalam Berkas Video dengan Metode  
Modifikasi LSB**

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](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 1. 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.

**Spesifikasi 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  $\text{Int}('S') + \text{Int}('T') + \text{Int}('E') + \text{Int}('G') + \text{Int}('A') + \text{Int}('N') + \text{Int}('O') = \dots$

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 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 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 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:

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

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. 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 1 Maret 2013 sebelum pukul 17.00 di Lab IRK). Terlambat menyerahkan tugas, nilai = 0.
3. Kakas pengembangan program bebas (Java, .NET, Delphi, Visual C, dll)
4. Program steganografi harus dibuat sendiri (namun untuk pustaka pengolahan video AVI dapat diambil dari kode yang sudah ada asalkan disebutkan sumbernya)
5. 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 .

6. Penilaian tugas dilakukan pada saat demo.
7. Beberapa sumber referensi yang perlu dibaca:
  - a. Steganography IV - Reading and Writing AVI files (<http://www.codeproject.com/Articles/5294/Steganography-IV-Reading-and-Writing-AVI-files> )
  - b. Steganography in video files ([http://brasil.cel.agh.edu.pl/~11ugbogusz/index.php?option=com\\_content&view=article&id=11&Itemid=16&lang=en](http://brasil.cel.agh.edu.pl/~11ugbogusz/index.php?option=com_content&view=article&id=11&Itemid=16&lang=en) )
  - c. Text Hiding in AVI Video ([www.iasj.net/iasj?func=fulltext&aId=28751](http://www.iasj.net/iasj?func=fulltext&aId=28751))
  - d. Stego Machine – Video Steganography using Modified LSB Algorithm ([www.waset.org/journals/waset/v50/v50-91.pdf](http://www.waset.org/journals/waset/v50/v50-91.pdf))