Program Studi Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

Tugas Besar I IF4020 Kriptografi
Sem. I Tahun 2020/2021

Aplikasi Steganografi pada Berkas Citra, Audio, dan Video dengan Metode LSB dan Metode BPCS

Pada tugas besar ini anda diminta membuat gabungan program steganografi dan kriptografi untuk menyembunyikan pesan di dalam berkas citra, audio, dan video tak-terkompresi. Metode steganografi yang digunakan untuk berkas citra adalah LSB dan BPCS, sedangkan untuk berkas audio dan video menggunakan metode LSB saja. Cipher kriptografi yang diguankan adalah Vigenere Cipher extended (256 karakater) yang sudah dibuat pada Tucil 1. Spesifikasi tugas dijelaskan di bawah ini.

I. Penyembunyian pesan di dalam berkas citra

Dalam tugas besar ini, anda diminta membuat program steganografi pada citra digital dengan metode LSB dan BPCS. Format citra yang digunakan adalah BMP (bitmap) dan PNG (Portable Network Graphics). Format BMP tidak terkompresi, sedangkan format PNG terkompresi dengan metode kompresi *lossless*.

Pada praktekmya, sebelum disisipkan, pesan dapat 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. Kunci Vigenere Cipher menjadi kunci stego. Pesan yang disisipkan adalah sembarang *file* dengan ukuran yang tidak melebihi kapasitas penyisipan (*payload*). Kapasitas penyisipan dihitung sebelum proses penyisipan.

Pesan dapat disisipkan secarak acak pada setiap blok 8 x 8, sehingga pembangkitan bilangan acak menjadi kunci stego 2

Spesifkasi program:

1. Program menerima masukan berupa citra digital dengan format BMP atau PNG, nama file pesan, dan kunci stego (opsional, jika pengguna memilih untuk mengenkripsi pesan dan/atau jika memilih penyisipan secara acak).
2. Metode steganografi yang digunakan adalah metode LSB dan BPCS.
3. Untuk metode BPCS, selain masukan di atas, parameter *threshold* pada metode BPCS juga menjadi salah satu masukan (default = 0.3).
4. Pengguna dapat memilih apakah pesan dienkripsi atau tidak dienkripsi sebelum disisipkan.
5. Jika menggunakan metode LSB, maka ada pilihan apakah pesan disisipkan secara sekuensial pada pixel-pixel citra atau acak.
6. Jika menggunakan metide BPCS, pengguna dapat memilih apakah pesan disisipkan secara sekuensial pada blok-blok 8 x 8 atau pada blok-blok acak.

Struktur menu kira-kira sebagai berikut (anda boleh membuat struktur menu yang lain, tidak harus sama dengan di bawah ini):

A. Penyisipan pesan
   (x) Tanpa enkripsi ( ) Dengan enkripsi (ket: check box)
   1. Metode LSB
      1.1 Pixel-pixel Sekuensial
      1.2 Pixel-pixel acak

   2. Metode BPCS
      2.1 Blok-blok Sekuensial
      2.2 Blok-blok acak
B. Ekstraksi pesan
   1. Metode LSB
   2. Metode BPCS

Pada waktu ekstraksi pesan, pengguna tidak memilih lagi apakah sekuensial atau acak. Program harus dapat menentukan apakah pada waktu penyisipan pesan dilakuka secara acak atau secara sekuensial. Cara yang paling mudah adalah menyisipkan kode tertentu pada pixel-pixel awal pada frame pertama yang mengindikasikan pilihan sekuensial atau acak pada waktu penyisipan (misalnya kode 11, 12, 21, 22) atau denga cara yang lain.

7. 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.
8. Jangan menyisipkan kunci di dalam file citra.
9. Program menolak menyisipkan pesan jika ukuran file pesan melebihi *payload*.
10. Program dapat menyimpan *stego-image* (citra yang sudah disisipi pesan)..
11. Program dapat mengekstraksi pesan utuh seperti sediakala dan menyimpannya sebagai file dengan nama lain (*save as*).
12. 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.
13. Program dapat menampilkan (*view*) citra asli dan citra stego dalam dua jendela berbeda.
14. 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:

$$PSNR = 20 \times log_{10}\left(\frac{255}{rms}\right) \qquad \text{(II.13)}$$

yang dalam hal ini 255 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). Dari praktek pengolahan citra, citra dengan PSNR > 30 masih dapat dianggap kualitasnya bagus, tetapi jika PSNR < 30 dikatakan kualitas citra sudah terdegradasi secara signifikan.

15. Citra uji yang digunakan sedikitnya berupa citra homogen (misalnya gambar langit biru, salju, laut, dsb), citra heterogen (misalnya gambar bunga-bunga di taman), citra *grayscale*, dan citra berwarna.
16. Fitur-fitur lainnya dipersilakan dibuat.

II. Penyembunyian pesan di dalam berkas video

AVI adalah salah satu format video digital. AVI adalah singkatan dari *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 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 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. 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.

Spesifkasi program:

1. Program menerima masukan berupa video digital dengan format AVI, nama file pesan, dan kunci-stego.
2. Metode steganografi yang digunakan adalah metode LSB.
3. Pengguna dapat memilih apakah pesan dienkripsi atau tidak dienkripsi sebelum disisipkan.
4. Pengguna dapat memilih apakah pesan disisipkan secara sekuensial pada *frame-frame* video (frame 1, 2, 3, ds) atau frame dipilih secara acak (frame 17, 21, 10, 9, 11, dst). Selain itu, untuk setiap frame yang dipilih, pengguna juga dapat memilih apakah disisipkan secara sekuensial atau secara acak

   Struktur menu kira-kira sebagai berikut:

   A. Penyisipan pesan
      (x) Tanpa enkripsi ( ) Dengan enkripsi (ket: check box)
      1. Frame sekuensial
         1.1 Pixel-pixel Sekuensial
         1.2 Pixel-pixel acak
      2. Frame acak
         2.1 Pixel-pixel Sekuensial
         2.2 Pixel-pixel acak
   B. Ekstraksi pesan

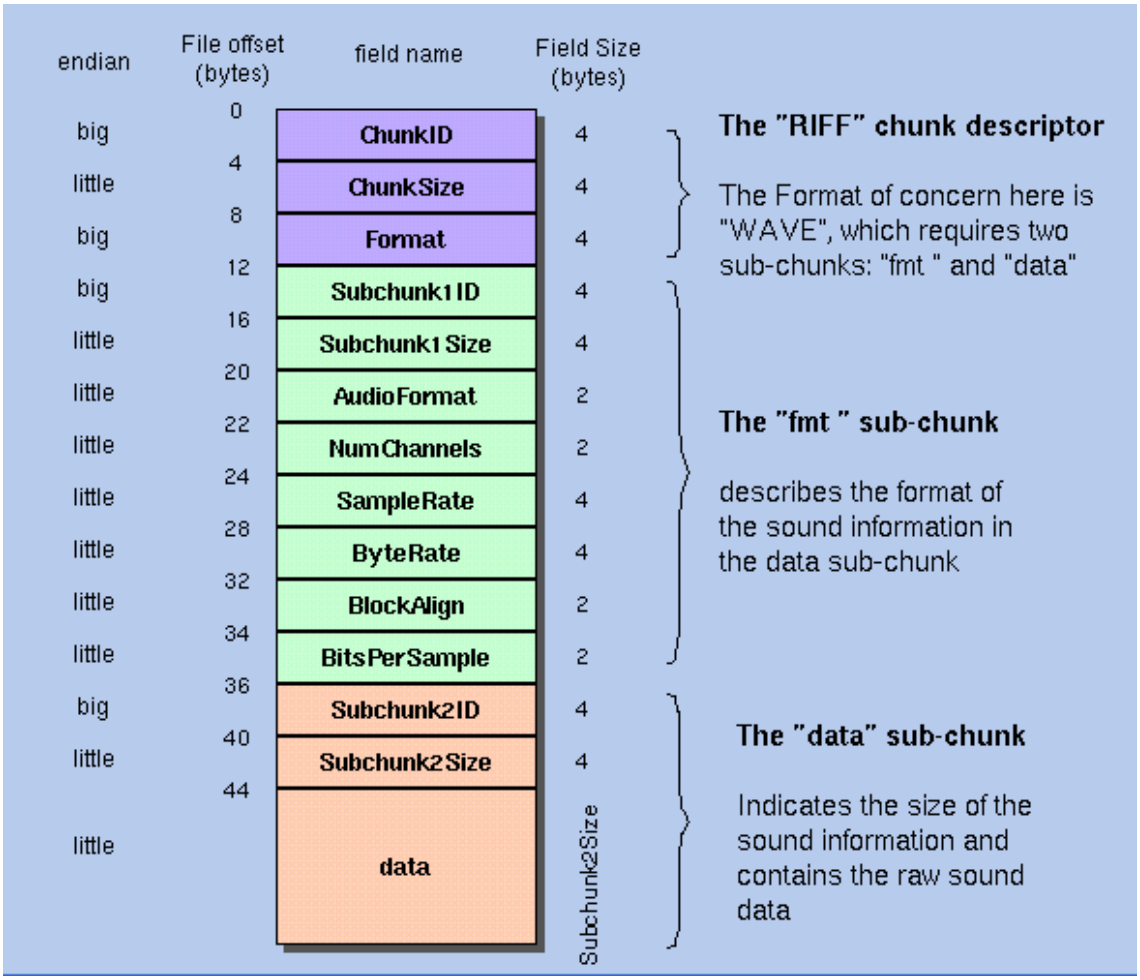
   Pada waktu ekstraksi pesan, pengguna tidak memilih lagi apakah sekuensial atau acak. Program harus dapat menentukan apakah pada waktu penyisipan pesan dilakuka secara acak atau secara sekuensial. Cara yang paling mudah adalah menyisipkan kode tertentu pada pixel-pixel awal pada frame pertama yang mengindikasikan pilihan sekuensial atau acak pada waktu penyisipan (misalnya kode 11, 12, 21, 22) atau denga cara yang lain.
5. 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.
6. JANGAN menyisipkan kunci di dalam file video.
7. Program menolak menyisipkan pesan jika ukuran file pesan melebihi *payload*.
8. Program dapat menyimpan *stego-video* (video yang sudah disisipi pesan) dengan nama berbeda (*Save as*)

9. Program dapat mengekstraksi pesan utuh seperti sediakala dan menyimpannya sebagai *file* dengan nama lain (*save as*).
10. 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.
11. Program dapat memainkan (*playback*) video asli dan stego-video melalui sebuah video *player* yang dipanggil dari dalam program (gunakan API).
12. Program dapat menampilkan ukuran kualitas video hasil steganografi dengan *PSNR*. Oleh karena video terdiri dari banyak *frame*, maka PSNR video adalah rata-rata PSNR dari seluruh *frame* yang disisipkan pesan saja.
13. Fitur-fitur lainnya dipersilakan dibuat.

III. Penyembunyian pesan di dalam berkas audio

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:



Penyisipan pesan ke dalam berkas audio WAV dapat menggunakan salah satu metode berikut: metode LSB. Metode LSB pada audio prinsipnya sama seperti metode LSB 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 pada prinsipnya sama seperti pada citra dan video:

1. Program menerima masukan berupa file audio digital dengan format WAV, nama file pesan, dan kunci-stego.
2. Metode steganografi yang digunakan adalah metode LSB (1 bit).
3. Pengguna dapat memilih apakah pesan dienkripsi atau tidak dienkripsi sebelum disisipkan.
4. Pengguna dapat memilih apakah pesan disisipkan secara sekuensial di dalam audio atau secara acak.
5. Pada waktu ekstraksi pesan, pengguna tidak memilih lagi apakah sekuensial atau acak. Program harus dapat menentukan apakah pada waktu penyisipan pesan dilakukan secara acak atau secara sekuensial. Cara yang paling mudah adalah menyisipkan kode tertentu pada awal audio .
6. Pengguna memasukkan sebuah kata kunci (maksimal 25 karakter) yang berfungsi dua: sebagai kunci enkripsi pada *Vigenere Cipher* dan sebagai kunci (*seed*) pembangkitan bilangan acak.
7. Contoh: Kunci = ’STEGANO’, kunci ini langsung dijadikan sebagai kunci enkripsi.

8. 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') = ...
9. 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.
10. JANGAN menyisipkan kunci di dalam file audio.
11. Program menolak menyisipkan pesan jika ukuran file pesan melebihi *payload*.
12. Program dapat menyimpan *stego-audio* (video yang sudah disisipi pesan) dengan nama berbeda (*Save as*)
13. Program dapat mengekstraksi pesan utuh seperti sediakala dan menyimpannya sebagai *file* dengan nama lain (*save as*).
14. 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.
15. Program dapat memainkan (*playback*) berkas audio asli dan stego-video melalui sebuah video *player* yang dipanggil dari dalam program (gunakan API).
16. Berkas WAV dapat diperoleh dengan *converter* dari MP3 ke WAV (cari *free software* nya di internet) atau dari sumber lain.
17. Berkas audio dapat berupa mono (1 band) atau stereo (2 band)
18. 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 sesudah penyembunyian pesan. Nilai minimal PSNR adalah 30 DB (jika kurang dari 30 DB berarti sinyal audionya mengalami kerusakan yang berarti).

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 5 Oktober 2020 sebelum jam kuliah di Google Drive (akan diberitahukan kemudian oleh asisten/dosen).
3. Bahasa pemrograman yang digunakan bebas (Java, C, C++, Python, Go, dll)
4. Program steganografi harus dibuat sendiri (namun untuk pustaka pengolahan video AVI dan audio WAV dapat diambil dari kode yang sudah ada asalkan disebutkan sumbernya).
   Program Vigenere Cipher tidak perlu dibuat lagi, gunakan program dari Tucil 1.
5. Yang diserahkan pada saat pengumpulan antara lain:
   a. Kode program (*source code*).
   b. Laporan yang memiliki sistematika sebagai berikut :
      i. Teori singkat (steganografi, metode LSB, BPCS, image, audio, video, dll).
      ii. Perancangan dan Implementasi, termasuk.
      iii. Pengujian program dan analisis hasil. Uji program dengan bermacam-macam citra, audio dan video dan jenis berbagai jenis dan ukuran file pesan.
      iv. Kesimpulan dari hasil implementasi.
      v. Tampilkan foto anda bertiga di *cover* laporan sebagai pengganti logo gajah.

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)
    c. Text Hiding in AVI Video (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*)*
    f. Information Hiding Using Audio Steganography – A Survey (http://aircconline.com/ijma/V3N3/3311ijma08.pdf).