Tugas Besar I IF4020 Kriptografi
Sem. I Tahun 2016/2017

Aplikasi Fragile Watermarking untuk Otentikasi Citra

Implementasikan sebuah metode *fragile watermarking* berdasarkan algoritma modifikasi LSB untuk mengontetikasi citra. Dengan metode *fragile watermarking* tersebut, kita dapat membuktikan apakah sebuah citra sudah diubah, kalau iya bagian mana dari citra tersebut yang mengalami perubahan (*tamper proofing*).

Metode *fragile watermarking* didasarkan pada makalah terlampir (makalah dosen Kripto pada sebuah konferensi internasional) dengan beberapa perubahan sbb: *watermark* tidak dienkripsi dengan sistem *chaos* seperti pada makalah, tetapi dienkripsi dengan Vigenere Cipher/Playfair cipher yang sudah anda buat pada Tucil 1. *Watermark* adalah sebuah citra biner yang dibuat sedemikian rupa sehingga berukuran sama dengan citra *host*. Metode tersebut dapat diterapkan untuk citra *grayscale* maupun citra berwarna.

Setelah citra diberi watermark, selanjutnya lakukan pengujian pada citar ber-*watermark* untuk mengetahui keotentikannya dengan *Photoshop*. Pengujian yhang dilakauan adalah bermacam-macam operasi image processing dan digital imaging seperti: pengubahan kontras/*brightness*, penambahan konten baru di dalam citra, penghapusan sebagain konten di dalam citra, *flipping*, *noising*, *sharpening*, dll.

Spesifkasi program:

1. Program menerima masukan berupa citra digital dengan format BMP atau PNG, nama file watermark, dan kunci steganografi.
2. 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.
3. Jangan menyisipkan kunci di dalam file citra.
4. Program dapat menyimpan *watermarked-image* (citra yang sudah disisipi pesan)..
5. Program dapat mengekstraksi watermark dari dalam citra dan menyimpannya sebagai file dengan nama lain (*save as*).
6. Program dapat menampilkan (*view*) citra asli dan citra ber-watermark, watermark asli,d an watermark hasil ekstraksi.
7. Program dapat menampilkan ukuran kualitas citra hasil watermarking 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.

8. 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 .... sebelum pukul 8.00 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 (fragile watermarking, 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 dan jenis file pesan. Ukur kapasitas penyimpanan dan *fidelity*-nya (PSNR)
        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.