STUDI *WATERMARKING* DAN *MAULICIOUS ATTACK* TANPA INFORMASI ALGORITMA YANG DILAKUKAN TERHADAP *IMAGE WATERMARKING*

Brahmasta Adipradana – NIM : 13503082

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if13082@students.if.itb.ac.idl

Abstrak

Makalah ini membahas tentang studi mengenai *watermarking* secara global dan serangan-serangan yang tergolong *Maulicious Attack* terhadap *image watermarking*. *Watermarking* merupakan sebuah teknik untuk menyisipkan informasi ke dalam sebuah *file* digital yang ditujukan untuk menjaga keaslian dari file tersebut. Penerapan *watermarking* ini dilakukan dalam banyak jenis file digital seperti gambar, video, dan audio.

Sebuah *watermarking* yang baik tentunya memiliki beberapa persyaratan. Syarat-syarat tersebut adalah . Imperceptibility, Key uniqueness, Noninvertibility, Image Dependency, dan Robustness. Salah satu yang menjadi perhatian lebih dalam makalah ini adalah robustness, yaitu bagaimana sebuah *watermarking* bisa bertahan dalam serangan-serangan yang dilakukan untuk membuka *watermark* yang disisipkan di sana.

Secara umum, ada empat jenis serangan yang digunakan di dalam menyerang *image watermarking*. Keempatnya adalah *Removal Attacks*, *Geometrical Attacks*, *Cryptographic Attacks*, dan *Protocol Attacks*. Selain itu , terdapat pengkategorian jenis serangan yang lain yaitu *Maulicious Attack dan Non Maulicious Attack. Maulicious attack* merupakan serangan terhadap *image watermarking* yang bertujuan agar *watermark* tidak dapat dideteksi atau dihilangkan dari gambar tersebut. Namun pada umumnya serangan dengan *Maulicious attack* ini dilakkukan dengan mengetahui algoritma yang digunakan untuk menyisipkan informasi. Saat ini beberapa metode telah diperkenalkan untuk melakukan serangan yang ingin menghilangkan *watermark* tanpa mengetahui informasi algoritma yang digunakan. Beberapa diantaranya yang akan dibahas di sini adalah *Copy Attack*, *Template Attack*, dan *Blind Estimation Without Priors*

Kata kunci*: watermarking, image watermarking*, serangan terhadap *watermarking, maulicious attack, copy attack, template attack, blind estimation.*

1. Pendahuluan

Produk digital pada saat ini merupakan sesuatu yang sangat populer dikarenakan perkembangan teknologi. Gambar, video, dan audio bisa ditransmisikan dari satu tempat ke tempat lain tanpa kehilangan banyak kualitasnya. Akibatnya muncul masalah mengenai penggunaan produk tersebut secara ilegal seperti perekaman, manipulasi, atau menggunakannya untuk kepentingan komersial. Hal ini dapat menyebabkan pencipta dari produk digital tersebut mengalami kerugian.

Untuk mengatasinya, berbagai teknik telah diterapkan untuk melakukan kontrol terhadap produk digital tersebut. Solusi-solusi teknis dibutuhkan untuk membuat lingkungan di mana hanya data digital yang sah yang dapat digunakan.

Cara terbaik untuk melakukan hal tersebut adalah dengan mamsukkan informasi privat atau publik ke dalam data digital tersebut, untuk menjamin bahwa file tersebut merupakan hak milik penciptanya. Informasi yang disisipkan tersebut akan terbawa terus sehingga ketika terjadi suatu pelanggaran *copyright*, pencipta bisa melakukan claim terhadap file digital tersebut.

Digital *watermarking* adalah teknik yang memungkinkan penyisipan itu terjadi. Penyisipan ini akan menolong untuk menyelesaikan masalah pendeteksian pemilik dari file illegal, memonitor oengguaan dari

penggunaan data yang memiliki copyrighm dan menganalisa penyebaran data di dalam jaringan.

Gambar 1 Sebelum diberi *watermark*

Gambar 2 Sesudah diberi *watermark*

Gambar 3 *Watermark* yang digunakan

Tujuan dari *watermarking* adalah untuk mengkomunikasikan informasi. Dalam hal ini banyak sekali isu yang diangkat. Salah satunya adalah robustness atau kekokohan dari penyimpanan informasi tersebut.

Kekokohan ini berkaitan dengan serangan terhadap *watermarking*. Jika sebuah penyisipan informasi dengan *watermarking* tersebut gampang dirusak, maka tujuan utama dalam *watermarking* akan sulit untuk tersampaikan. Untuk itu dibutuhkan sebuah metode penyisipan yang baik sehingga tidak memungkinkan utnuk dimanipulasi dan dirusak.

Ilmu mengenai serangan terhadap *watermarking* sendiri juga ikut berkembang seiringan dengan ilmu *watermarking* tersebut. Namun, ilmu ini kebanyakan digunakan untuk menganalisa kelemahan dari *watermarking*, bukan untuk merusak. Hasil analisis kelemahan tersebut akan berguna sebagai masukan dalam perkembangan *watermarking* itu sendiri. Selain itu, pengetahuan ini membantu kita untuk lebih mengerti mengenai apa yang bisa dilakukan oleh penyembunyian data dan apa yang tidak untuk proteksi kepemilikan, deteksi penyalahgunaan, dan kontrol terhadap akses. Semakin spesifik ilmu yang diperole mengenai proses tersebut, semakin baik kita bisa melakukan desain terhadap sistem yang bisa bertahan.

2. *Watermarking*

Watermarking sudah ada sejak 700 tahun yang lalu. Pada akhir abad 13, pabrik kertas di Fabriano, Italia, membuat kertas yang diberi *watermark* atau tanda-air dengan cara menekan bentuk cetakan gambar atau tulisan pada kertas yang baru setengah jadi. Ketika kertas dikeringkan terbentuklah suatu kertas yang ber-*watermark*. Kertas ini biasanya digunakan oleh seniman atau sastrawan untuk menulis karya mereka. Kertas yang sudah dibubuhi tanda-air tersebut sekalius dijadikan identifikasi bahwa karya seni di atasnya adalah milik mereka.

Ide *watermarking* pada data digital (sehingga disebut *digital watermarking*) dikembangkan di Jepang tahun 1990 dan di Swiss tahun 1993. *Digital watermarking* semakin berkembang seiring dengan semakin meluasnya penggunaan internet, objek digital seperti video, citra, dan suara yang dapat dengan mudah digandakan dan disebarluaskan.

Watermarking sendiri adalah sebuah teknik untuk menyisipkan informasi tertentu dalam sebuah data digital. Informasi yang tersimpan itu disebut dengan *watermark*. Saat ini, ilmu *watermarking*, terutama dalam hal *image watermarking*, merupakan salah satu bidang ilmu yang populer untuk authentikasi dan proteksi *copyright* [VOL00]. Hal ini disebabkan tingginya kebutuhan untuk perlindungan hak milik dari karya cipta.

Watermark, informasi yang dimasukkan ke dalam *file digital* dalam teknik ini dapat berupa teks, logo, data audio, hingga rangkaian bit yang tidak berarti. Penyisipan ini dilakukan sebisa mungkin agar tidak mengubah karya cipta itu sendiri, sehingga mata manusia tidak mampu mendeteksinya. Sebuah gambar yang memiliki *watermarking* yang tidak terlihat lebih baik daripada yang terlihat.

Bagaimana *watermark* ini bisa diketahui? Seperti halnya teknik kriptografi lainnya, proses penyisipan informasi tersebut menggunakan sebuah algoritma dan sebuah kunci untuk mengenkripsinya. Maka dari itu suatu klaim dari sebuah file digital hanya dapat dilakukan oleh

pemilik karya tersebut. Karena ia mengetahui bagaimana *watermark* itu disisipkan dan kuncinya.

Adapun jenis-jenis penggunaan *watermarking* yang umum dalam kehidupan sehari-hari adalah sebagai berikut:

a. Memberi label kepemilikan (ownership) pada file digital
   Informasi ownership yang disimpan tersebut dapat berupa identitas diri atau gambar yang menspesifikasikan pemilik.
b. Otentikasi atau *tamper proofing*
   Pemilik file digital menyisipkan *watermark* untuk membuktikan apakah file yang disimpan masih asli atau sudah berubah, Jika *watermark* yang diekstraksi tidak tepat sama dengan *watermark* asli maka dapat disimpulkan file tersebut sudah tidak otentik lagi
c. Fingerprinting
   Pemilik file digital mendistribusikan file yang sama ke beberapa distributor, setiap file yang terdistribusi memiliki identitas berupa *watermark*. Jika *watermark* tersebut. Penggandaan ilegal dapat diketahui dari dari adanya *watermark* yang sama. Pengganda ilegal tersebut juga dapat dideteksi dari file *watermark* yang ada di file tersebut. Operasi fingerprinting ini dapat juga dilakukan di perangkat keras, namun butuh kolaborasi dengan perangkat keras.
d. Aplikasi Medis
   Citra medis seperti foto sinar-X diberi *watermark* berupa ID pasien dengan maksud untuk memudahkan identifikasi pasien. Informasi lain yang dapat disisipkan adalah hasil diagnosis penyakit.

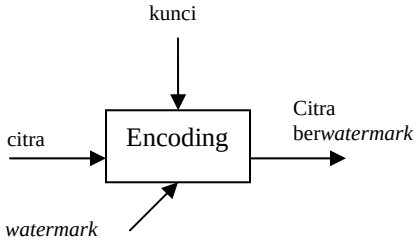
Bila dilihat *watermarking* memiliki kemiripan dengan steganografi. *Watermarking* merupakan aplikasi dari steganografi, namun ada perbedaan antara keduanya. Jika pada steganografi informasi rahasia disembunyikan di dalam media digital dimana media penampung tidak berarti apa-apa, maka pada *watermarking* justru media digital tersebut yang akan dilindungi kepemilikannya dengan pemberian label hak cipta (*watermark*).

Meskipun steganografi dan *watermarking* tidak sama, namun secara prinsip proses penyisipan informasi ke dalam data digital tidak jauh berbeda.

3 Proses Penyisipan dan Pembuktian *Watermark*

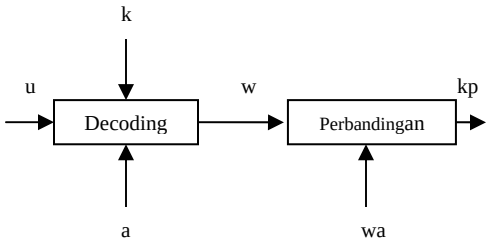
Seperti dijelaskan sebelumnya, terdapat sebuah mekanisme untuk melakukan *watermarking* ini. Secara umum, *watermarking* terdiri atas dua tahapan [RIN06], yaitu:
1. Penyisipan *watermark* (*watermark* embedding)
2. Ekstraksi *watermark* (*watermark* detection)

Kedua proses ini digambarkan pada gambar 3.1 di bawah, di mana diibaratkan terdapat sebuah encoder yang melakukan penyisipan *watermark* terhadap citra.



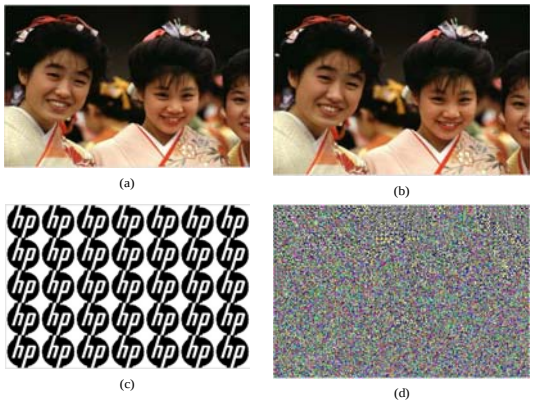
Bila sebuah *watermark* telah diberikan terhadap file digital, tanda tersebut akan terbawa terus bersama file.

Selanjutnya jika sebuah file digital telah digunakan dan melanggar aspek hukum, tentunya sang pemilik bisa melakukan klaim. Untuk membuktikannya, pemilik bisa melakukan verifikasi dari file digital ini. Proses ini dilakukan dengan melakukan decoding antara file asal dengan file yang disebut asli oleh pemiliknya. Secara umum proses ini digambarkan sebagai berikut:



Di mana:
k = kunci
u = Citra yang diuji
a = Citra asli
w = *watermark*

wa = *watermark* asli
kp = keputusan



Kunci yang salah akan memberikan *watermark* yang salah pula. Seperti contoh pada gambar di atas, terlihat bawa kunci yang benar pada gambar di kiri atas akan mengekstraksi gambar kiri bawah, yang menunjukkan logo perusahaan pemilik dari gambar tersebut. Namun jika kita memasukkan kunci yang salah, akan menghasilkan *watermark* yang tidak berarti.

4. Jenis-Jenis *Watermarking*

Ada beberapa jenis pengkategorian *watermarking* . Berdasarkan presepsi manusia, *watermarking* dapat dibedakan menjadi visible *watermarking* dan invisible *watermarking*. Pada invisible *watermarking*, *watermark* pada file digital tidak terlihat. Sementara untuk visible, *Watermark* tersebut terlihat dengan jelas. Contoh informasi yang diberikan pada visible *watermarking* adalah logo perusahaan, nama pencipta, dan lain sebagainya.

Bila dilihat pada tingkat kekokohan, *watermarking* dapat dibedakan menjadi secure *watermarking*, robust *watermarking*, dan fragile *watermarking*. Secure *watermarking* berarti *watermark* harus dapat bertahan terhadap *maulicious* dab non-*maulicious attack*. Sementara robust *watermarking* hanya bertahan terhadap non-*maulicious attack* dan fragile *watermarking* boleh dibilang tidak tahan terhadap serangan. Hilangnya *watermark* pada fragile *watermarking* menandakan bahwa karya cipta tersebut telah dirusak.

Penjelasan mengenai jenis-jenis serangan dapat dilihat pada bab Serangan-Serangan pada *Watermarking*.

5. Kriteria *Watermarking* yang Baik

Sebuah proses *watermarking* yang baik adalah yang memenuhi persyaratan sebagai berikut[RIN06]:

1. *Imperceptibility*
2. *Key uniqueness*
3. *Noninvertibility*
4. *Image Dependency*
5. *Robustness*

Imperceptibility adalah tidak dapat dipresepsinya keberadaan *watermark* oleh indra visual. *Key uniqueness* berarti kunci yang berbeda menghasilkan *watermark* yang berbeda, sehingga penggunaan kunci yang salah dapat menyebabkan hasil ekstraksi *watermark* yang salah pula. Noninvertibility adalah sulitnya menemukan *watermark* secara komputasi jika hanya diketahui file yang ber*watermark* saja. Image Dependency adalah satu kunci menghasilkan *watermark* tunggal tetapi. Robustness adalah *watermark* tetap kokoh terhadapa berbagai serangan yang dilakukan pada citra ber-*watermark*. Maksud serangan di sini adalah adanya manipulasi terhadap sebuah file digital yang ber-*watermark*.

6. Metode-Metode Image *Watermarking*

Penyisipan *watermark* dalam file gambar (*image*) dapat dilakukan dalam dua *domain*, yaitu spasial dan transform. Penyisipan spasial berarti langsung menyisipkan ke dalam citra. Sementara penyisipan *transform* berarti menyisipkan *watermark* ke dalam koefisien transformasi.

Setiap metode memiliki kelebihan dan kekurangannya masing-masing. Penyisipan spasial memiliki kelebihan dalam kecepatan proses *watermarking*. Namun sayangnya proses ini tidak kokoh terhadap serangan. Penyisipan *transform* memiliki kelebihan dalam *robustness*. Biasanya algoritma penyisipan *transform* tahan terhadap operasi geometri (penskalaan, rotasi, pergeseran). Kekokohan terhadap pemotongan (*cropping*) juga didapatkan jika *watermark* tersebar di seluruh file. Namun kelemahannya adalah untuk melakukannya tidak mudah.

Umumnya, ranah yang menjadi tempat penyisipan adalah frekuensi dan transformasi yang digunakan. Contoh dari metode ini adalah

DFT (Discrete Fourier Transform), DCT (Discrete Cosine Transform), dan DWT (Discrete Wavelet Transform).

Selain itu terdapat sebuah metode yang bernama LSB (Least Significant Bit) yang dapat digunakan pada proses *watermarking*. Penyisipan *watermark* dilakukan dengan mengganti bit-bit LSB dengan citra bit-bit *watermark*. Sayangnya metode ini tidak robust karena dapat dengan mudah dimodifikasi.

Metode lain adalah *spread spectrum watermarkng*. Pada awalnya citra ditransformasikan ke dalam ranah frekuensi, lalu bit *watermark* disisipkan pada koefisien transformasi. Secara umum metode ini lebih kokoh terhadap serangan seperti kompresi, cropping, dan penapisan lolos rendah.

7. Serangan-Serangan Terhadap Image *Watermarking*

Untuk membuktikan tingkat robustness dari sebuah algoritma *watermarking*, tentunya dilakuan sebuah ujicoba. Ujicoba dilakukan dengan menyerang file gambar yang telah disisipkan *watermark*.

Bila dilihat dari jenisnya, ada beberapa kategori serangan terhadap image *watermarking* [KUT00], yaitu:

1. *Removal Attacks*
2. *Geometrical Attacks*
3. *Cryptographic Attacks*
4. *Protocol Attacks*

Removal attacks adalah menghapus bersih *watermark* dari gambar. Pendekatan ini menganggap *watermark* yang dimasukkan ke dalam gambar adalah sebuah *noise*. Dengan perhitungan statistik teknik ini berusaha menebak bagaimana bentuk gambar aslinya. Salah satu algoritma yang efisien untuk penyerangan ini adalah yang diajukan oleh Langelaar. Mereka melakukan serangkaian operasi untuk image, termasuk median filteringm highpass filtering, dan non linear truncation. Selain itu, Voloshnovskiy mengajukan prediksi *watermark* spatial melalui filtering proses berdasarkan posteriori maksumum (MAP) demham remodulasi untuk membuat distribusi noise yang paling buruk untuk detektor *watermark.*

Berbeda dengan *Removal attacks*, *Geometrical attack* lebih memilih tidak untuk membuang *watermark* yang telah disesipkan tapi memberi distorsi terhadapnya dengan alterasi spatual atau temporal dari data stefo. Penyerangan ini selalu membuat detektor *watermark* gagal untuk melakukan sinkronisasi dengan *watermark* yang telah disisipkan. Serangan pertama yang berdasarkan metode ini adalah serangan terhadap gambar. Ada dua utiliti, yaitu Unsign dan Stirmark. Unsign memperkenalkan local pizel jittering dan sangat efisien dalam menyerang domain spatial dari skema *watermarking*. Sementara Stirmark memperkenalkan local *Geometrical* bending sebagai tambahan dari global *Geometrical* transformation.

Cryptographic attacks sangat mirip dengan serangan yang digunakan dalam ilmu kriptografi dan bisa saja berbeda dalam kenyataannya. Serangan in berupa serangan brute force yang berusaha menemukan rahasia melalui exhaustive search,

Protocol attack tidak melakukan penghancoran informasi yang disisipkan atau merusak deteksi terhadap informasi tersebut. Tujuan dari serangan ini adalah untuk menyerang konsep dari aplikasi *watermark*. *Protocol attack* yang pertama diperkenalkan oleh Craver . Mereka memperkenalkan konsep invertible *watermark* dan menunjukkan bahwa untuk tujuan proteksi copyright *watermark* seharusnya non-invertible. Kebutuhan ini dalam teknologi *watermarking* berarti bahwa seharunya tidak mungkin untuk mengambil *watermark* dari gambar yang tidak memiliki *watermark.*

Sementara pada referensi lain [RIN06], jenis-jenis serangan dapat dikategorikan sebagai

1. *Maulicious Attack*
2. Non-*Maulicious Attack*

Non-maulicius *attack* merupakan serangan yang normal terjadi saat file digital digunakan, seperti kompresi, cropping, penambahan derau, dan sebagainya. Jika sebuah file digital dikenakan serangan seperti itu dan *watermark* dapa diekstraksi, maka proses *watermarking* tersebut dikatakan robust.

Maulicious attack merupakan serangan yang tujuan utamanya adalah menghilangkan atau membuat *watermark* tidak dapat dideteksi. Biasanya penyerang dalam *maulicious attack* ini mengetahui algoritma dari *watermarking*

tersebut. Namun terdapat juga jenis serangan di mana penyerang tidak mengetahui algoritma dari *watermarking* tersebut. Pada karya tulis ini *maulicious attack* tanpa pengetahuan mengenai algoritma tersebut akan dibahas pendekatannya.

Meski setiap klasifikasi menunjukkan perbedaan yang jelas antar serangan-serangan tersebut, sangat sering terjadi penyerang *watermark* tidak hanya menggunakan satu jenis serangan yang ada. Tapi malah menggunakan kombinasi dari antaranya.

Adapun beberapa contoh dari seragan-serangan tersebut bila dikategorisasi berdasarkan jenis yang pertama diperlihatkan dalam tabel berikut

Removal Attacks	Geom. Attacks	Crypt. Attacks	Protocol Attacks
• Denoising	• Affine Transf.	• Brute Forth	• WM Inversion
• Remodulation	• Warping	• Collusion	• Copy Attack
• Lossy Comp.	• Jittering	• Averaging	
• Quantization	• Template Attack	• Oracle	

Dalam pembahasan makalah ini, ada beberapa teknik yang diterapkan untuk melakukan *watermarking*, yaitu:

Teknik-teknik ini merupakan penemuan yang berarti. Adapun detil dari teknik-teknik ini akan dijelaskan sebegai berikut.

7.1. *Watermarking Copy Attack*

7.1.1 Gambaran Umum

Watermarking Copy *Attack* merupakan salah satu teknik serangan terhadap *watermarking* yang diperkenalkan oleh M. Kutter, S. Voloshynovskiy, dan A. Herriage dalam sebuah *paper* yang diperkenalkan pada Januari 2000 di Photonics West SPIE convention.

Serangan ini merupakan serangan baru yang termasuk jenis dari *Protocol Attack*. Konsep dari serangan terdiri atas melakukan salinan dari *stego image* menuju *terget image* tanpa menggunakan informasi spesifik tentang teknologi yang digunakan untuk menguyimpan informasi. Ada tiga langkah utaman dari proses *watermarking* ini, yaitu lakukan prediksi dari *watermark* di stego data. Daripada langsung memprediksi *watermark* yang terletak pada gambar, teknik ini melakukan komputasi untuk prediksi melalui proses *denoising.* Dengan kata lain, prediksi dari *watermark* dikomputasi dengan mengambil perbedaan antara *stego image* dan versi *denoised* dari *stego image*. Untuk menampilkan *denoising*, dilakukan ML-estimates dan MAP-estimates terhadap *cover image* dan mengajukan bentuk tertutup dan solusi iteratif untuk kasus-kasus tertentu darii noise dan statistik *cover image*. Pada langkah kedua *watermark* yang terprediksi diproses. Tujuan dari proses adalah untuk memaksimalkan energi dari *watermark* untuk *constrainti dari* imperceptibility. Untuk mengadaptasi *watermark* ke gambar tujuan diajukan fungsi visibilitas. Pada langkah terakhir, proses prediksi dari *watermark* adalah *target image.*

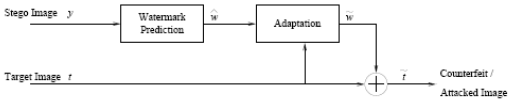
Keefektifan dari serangan telah diujikan dengan mengaplikasikannya ke dua tool *watermarking*. Hasil dari ujicoba tersebut akan diperlihatkan pada akhir dari bagian ini. Pada contoh tersebut diperlihatkan bahwa untuk kedua kakas tersebut mungkin untuk melakukan *copy* dari *watermark* dari *stego image* ke *target image*. Serangan ini dapat menyebabkan beberapa implikasi penting bergantung dengan aplikasi digital *watermarking*. Jika sebuah teknologi tidak bisa bertahan terhadap *copy attack*, seorang user bisa tidak yakin apakah *watermark* yang terdeteksi benar-benar data yang benar. Hal ini akan menjadi masalah dalam beberapa aplikasi.

Secara umum, teknik ini melakukan estimasi dari *watermark* yang diletakkan ke dalam sebuah media lalu melakukan pengisian *watermark* tersebut dalam media yang tidak ter*watermark*. Dalam skenario yang pertama, memungkinkan seorang pemilik gambar yang tidak autentik mendeklarasikan dirinya autentik, karena mengandung *watermark*. Sementara dalam skenario kedua, seorang penyerang bisa mengisi pasar dengan content yang memperbolehkan user untuk memanipulasi gambar, tapi dikarenakan *watermark*, terdapat limitasi. Hal ini sendiri kurang popular untuk distribusi luas.

Selankutnya akan dibahas bagaimana proses copy *attack* dari *watermark* ini dilakukan, sesuai pembahasan dari paper karya M. Kutter, S. Voloshynovskiy, dan A. Herriage.

Watermark copy *attack* termasuk dalam *Protocol attack*. Tujuan dari serangan ini adalah untuk melakukan copy sebuah *watermark* dari data stego ke target data tanpa mengetahui hal-hal mengenai teknologi *watermarking*nya.

Seperti dijelaskan sebelumnya, ide dari *watermark* copy *attack* muncul berdasarkan grup *Protocol attack*. Ini berarti tujuan dari serangan adalah tidak untuk menghancurkan *watermark* yang ada di dalam, tapi mengidentifikasi aplikasi yang digunakan oleh digital *watermark*s. Secara umum idenya adalah untuk melakukan copy sebuah *watermark* dari gambar satu ke gambar lain, dan hal ini tanpa informasi mengenai teknologi *watermarking* dan kunci rahasia.



Gambar 4 Gambaran umum proses watermark

Secara umum proses dari *watermark* dapat dilihat seperti di atas. Input dari sistem adalah stego image, yang mengandung *watermark* untuk disimpan, dan target image. Serangan ini terbagi atas tiga langkah utama. Di langkah pertama *watermark* dari stego image diprediksi, menghasilkan w. Prediksi ini kemudian diproses ke langkah berikutnya. Tujuan dari proses ini adalah untuk mengadaptasi *watermark* ke target image dengan tujuan untuk memaksimalkan energi dalam batasan untuk membuatnya dapat diteksi setelah penyisipan ke target image. Pada langkah terakhir, gambar yang telah terprediksi dan dan diproses diisi menjadi target image.

7.1.2. Prediksi *Watermark*

Memprediksi *watermark* yang ada di dalam gambar merupakan kunc utama serangan ini. Prediksi bisa dibagi menjadi dua cara, yaitu:
1. Prediksi langsung
2. Denoising.
Namun dalam pembahasan ini kita berfokus hanya pada teknik denoising.

Secara umum teknik *watermarking* bisa dimodelkan dalam persamaan matematis sebagai berikut:

$$y = x + w \quad (1)$$

Di mana y adalah stego image, x adalah gambar asli, dan w adalah *watermark*. Bila diumpamakan stego image adalah sebuah image yang noisy, maka *watermark* adalah noise dan kita bisa melaukan estimasi mengenai noise atau *watermark* w dengan mengambil perbedaan antara perkiraan x dari gambar asli dan stego image seperti berikut:

$$w' = y - z' \quad (2)$$

Pendekatan ini dapat digambarkan sebagai berikut:

Jika kita tidak memiliki informasi mengenai statistik dari stego image, kita bisa menggunakan teknik maximum likelihood (ML)-estimate dari *watermark*. Namun jika tidak, kita bisa menggunakan Maximum a posteriori Probability (MAP) estimate [KUT00], selanjutnya pembahasan mengenai dia teknik estimasi akan diberikan.

7.1.3 ML-Estimation

Seperti sudah dijelaskan sebelumnya, jika kita mengasumsikan tidak memiliki informasi statistik dari gambar dan statistik dari *watermark*, teknik ML-Estimation. Estimasi dirumuskan sebagai:

$$x' = \arg\max \{\ln p_w (y|x')\} \quad (3)$$

Di mana pw adalah probabilitasi kedalaman fungsi *watermark*.

ML-estimate memiliki dua solusi saat apakah *watermark* tergolong Gaussian atau Laplacian. Jika *watermark* merupakan distribusi Gaussian, maka ML-estimate diberikan sebagai local mean oleh y:

$$x' = \text{localmean}(y); \text{ Gaussian } watermark \quad (4)$$

Namun jika *watermark* memiliki Laplacian distribution, solusi dari ML-estimate diberikan oleh local median

$$x' = \text{localmedian}(y); \text{ Laplacian } watermark \quad (5)$$

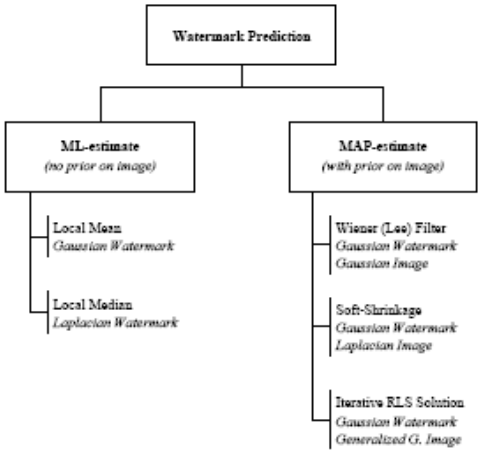
Melakukan komputasi ML-estimate dari cover image mengirangi perhitungan local mean atau local median. Untuk melakukan hal ini, beberapa pendekatan muncul dengan hanya menghitung rata-rata atau median di dalam sebuah gambar Walaubagaimanapun, jika kita mengasumsikan untuk bekerja dengan natural image, kita bisa melakukan komputasi untuk estimasi yang lebih akurat mengenai local mean atau median dengan hanya memperhitungkan pixel di *cross-shaped neighberhood*. Hal ini didasarkan fakta bahwa natural image menunjukkan korelasi yang tinggi dalam pengarahan horizontal dan vertikal.

5.1.4 MAP-Estimation

Jika kita punya informasi mengenai statistik dari cover image dan *watermark*, kita bisa menggunakan MAP Estimator.

Pembahasan mengenai MAP-Estimation tidak dicantumkan di karya tulis ini.

Namun, secara umum Kutt menggambarkan estimasi dalam bagan seperti berikut:



Gambar 5 Bagan umum estimasi yang dilakukan

Bagan tersebut menjelaskan bahwa untuk dalam melakukan prediksi ada dua alternatif, yaitu ML Estimate dan MAP-Estimate, dan setiap alteratif tersebut memiliki prinsip penyelesaiannya tergantung jenis *watermark* dan gambarnya. ML-estimate seperti telah dijelaskan sebelumnya memiliki Local Mean dan Local Median, sementara MAP-Estimate memiliki Wiener (Lee Filler), Soft Shrinkage, dan Iterative RLS Solution.

5.1.5 Copy Insertion

Setelah kita berhasil memprediksi *watermark* dari metode sebelumnya, hasil prediksi tersebut dapat dimasukkan ke dalam gambar tujuan. Namun pengisian ini membutuhkan sebuah proses agar tidak membuat hasil yang tidak baik pada gambar tujuan. Seperti dijelaskan sebelumnya, tujuan dari memproses adalah untuk mengadaptasikan salinan dari *watermark* tersebut ke gambar tujuan. Prosesnya sama dengan metode penyisipan biasa.

Pada papernya Kutter mencontohkan dengan metode Noise Visibility Function (NVF) yang diperkenalkan oleh Voloshynovskiy.

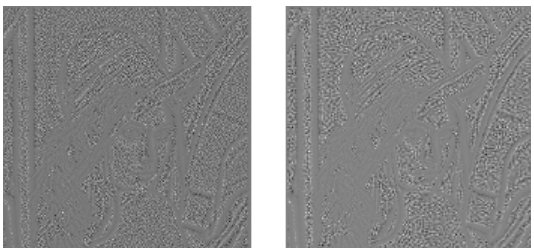
NVF mengkarakteristikkan tekstur lokal dari gambar atas 0 dan 1. Di mana 1 untuk area flat dan 0 untuk area yang bertekstur. Ini bisa digunakan untuk mendeskripsikan tekstur lokal untuk fenomena masking *watermark*, yakni di mana area yang paling banyak memiliki *watermark*, di situlah akan dilakukan paling banyak masking.

5.1.6 Hasil

Kutter melakukan ujicoba dari *watermark* copy *attack* pada dua software yang dapat menghasilkan *watermark*. Ia mereferensikannya menjadi software A dan B. Penyerangan dilakukan terhadap dua buah gambar grayscale yakni *lena* dan *cameraman*. Kedua gambar tersebut memiliki ukuran 256 x 256. Untuk melakukan prediksi, digunakan Wueber Fukter dengan ukuran window 5x5.

Adapun langkah-langkah yang dilakukan dari ujicoba tersebu adalah pertama *lena* diisi dengan sebuah tool *watermarking*, lalu *watermark* tersebut di-*copy* dari gambar lena dan dimasukkan ke dalam gambar *cameraman*. Hasil tersebut digambarkan pada gambar berikut:

Gambar 6 *Lena* sebagai *stegno image*



Gambar 7 *estimasi watermark yang didapatkan*

Gambar 8 gambar *photographer* sebagai *target image*

Gambar paling atas merupakan gambar *lena* yang telah disisipkan dengan *watermark* oleh software A (kiri) dan software B (kanan). Gambar pada baris kedua merupakan *watermark* hasil prediksi dari gambar di atasnya, dan gambar pada baris ketiga meupakan gambar *cameraman* setelah diisi *watermark*.

7.2 Serangan *Watermark* Berdasarkan *Blind Estimation Without Priors*

7.2.1 Gambaran Umum

Seperti halnya Copy *Attack*, Du Jiang memberikan sebuah pendekatan untuk melakukan serangan dalam *watermarking* tanpa mengetahui seperti apa algoritma *watermarking* yang digunakan. Dalam serangan ini, diasumsikan penyerangan menggunakan *blind estimation* yang diasumsikan bahwa *watermark* itu merupakan sesuatu yang ditambahkan (additive).

Serangan ini merupakan sebuah konsep.

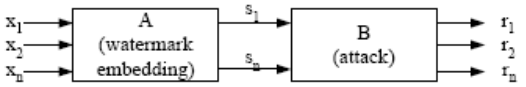
7.2.2 Rumusan Masalah

Seperti telah dijelaskan sebelumnya, *watermark* dapat dirumuskan seperti (1), di mana x adalah gambar asli, w adalah *watermark*, dan y adalah gambar yang telah disisipkan *watermark*.

Tujuan dari serangan adalah untuk membuang informasi *watermark*.

7.2.3 Pendekatan Umum

Estimasi dari sinyal yang didistorsi adalah permasalahan klasik yang data communication, radar signal processing, speech processingm dan image analysis. Di dalam Blind Signal Separation (BSS), tujuan yang ingin dicapai adalah untuk mengembalikan sinyal dari mixtures. Satu-satunya asumsi untuk metode BSS adalah kebebasannya dari sinyal aslinya. Biasanya sinyal diasumsikan untuk tergabung secara linear. Model data dan ide dasarnya digambarkan seperti di bawah:



Gambar 9 Model watermark berdasarkan sinyal

Sinyal yang berada di sebelah kiri (x1, x2,..,xn) merupakan meupakan sinyal sumber, di tengah (s1,s2,..sn) merupakan sinyal yang tercampur (dalam hal ini berupa gambar yang memiliki *watermark*), sinyal yang paling kanan (r1,r2,..,rn) nerupakan sinyal yang telah di-*recover* (dalam hal ini gambar yang telah diserang). Mixing yang dilakukan adalah statik, oleh karena itu dapat dimodelkan sebagai berikut:

$$s = Ax \quad (5)$$

Di mana A adalah mixing matriks yang tidak diketahui. Sementara untuk menghasilkan r:

$$r = Bx \quad (6)$$

BSS bisa diselesaikan dengan beberapa metode. Salahsatunya dengan ICA (Independent Component Analysis). ICA sangat dekat hubungannya dengan BSS. ICA merupakan teknik pemrosesan sinyal untuk merepresentasikan sekumpulan variabel random sebagai kombinasi dari komponen-komponen variabel independen. Secara umum komponen analisis independen dari vektor x bisa dibentuk dalam transformasi linear:

$$x = A\mathbf{G} \quad (7)$$

Jadi dengan metode tersebut, estimasi tehadap matriks A bisa dilakukan. Setelah matriks A diestimasi, matriks B bisa dikomputasi dengan menlakukan invers, dan selanjutnya menghasilkan komponen independen r dengan mudah.

Ide *Blind Estimation Watermark Attack* dapat dimasukkan ke dalam kategori removed *attack*. Sebab tujuan dari serangan ini adalah untuk memprediksi sinyal original dan *watermark* dari sinyal hasil *watermark*. *Watermark* dapat

dibuang dari gambar yang telah disisipi dalam beberapa kasus.

Namun secara eksak, pencarian nilai B tidak bisa dilakukan. Hal ini disebabkan kita tidak mengetahui isi dar imatriks A. Tapi untuk mendapatkan nilai A, kita bisa melakukan estimasi dengan estimator yang baik.

Contoh estimasi yang baik dicontohkan oleh Du Jiang dkk sebagai berikut: Berdasarkan persamaan (1), asumsikan bahwa gambar asli x adalah non Gaussian dan *watermark* w adalah Gaussioan. Jika ICA adalah transform, kita bisa melakukan transformasi ICA ke gambar yang telah disisipi *watermark*:

$$By = Bx + Bw = G + Bw \qquad (8)$$

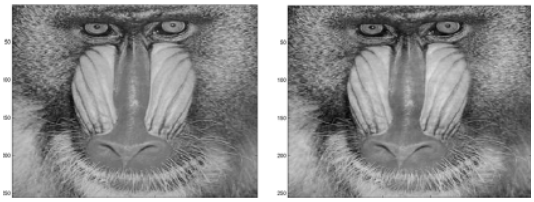
Di mana B adalah aproksimasi dari mising matriks ICA. Jika matriks B diambil secara tepat, maka komponen G = Bx menjadi sangat tidak Gaussian. , sementara Bw masih Gaussian dan putih. Dalam pemilihan matriks B, digunakan algoritma yang dideskripsikan Hyvarinen[HYV], lalu ML-estimate untuk image asli x telah dibuang.

Fungsi f memiliki bentuk karakteristik. Fungsi tersebut zero close terhadap original dan linear setelah mengambil nilai tergantung dari parameer Laplacian density dan Gaussian noise density[HYV]. *Watermark* yang dihasilkan cukup perbedaan antara s dan y'

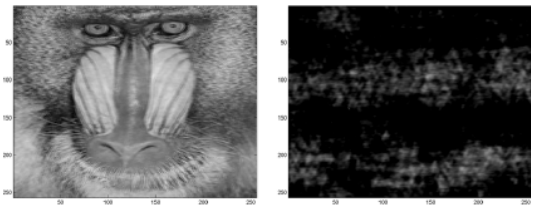
$$\hat{\mathbf{x}} = B^T \hat{\mathbf{G}} = B^T f(Bs) \qquad (9)$$

7.2.4 Hasil

Metode di atas diujikan terhadap sebuah grayscale image Baboon dengan ukuran 256x256. *Watermark* yang terenkripsi dmasukkan ke dalam 1000 higher magnitude coefficients dari gambar. Dengan menggunakan algoritma Hyvarinen[HYV], matriks B diestimasi dan mengembalikan transformasi ICA. Estimasi B membutuhkan vektor dari gambar asli dan gambar yang di*watermark*, yang pada kasus asli, tidak mungkin ada. Gambar berikut menunjukkan hasil eksperimen. Gambar pada baris pertama merupakan gambar yang tela diberi *watermarking* dengan algoritma spread spectrum. Baris kedua adalah gambar yang diserang, yang memiliki kualita gambar lebih daripada gambar yang ter-*watermark*, dan estimasi dari *watermark* yang disisipkan.



Gambar 10 Watermarking dengan metode normal



Gambar 11 Hasil estimasi watermark

7.3 *Watermark* Template *Attack*

7.3.1 Gambaran Umum

Watermark template *attack* adalah sebuah serangan untuk mengatasi template based *watermarking* systems. Serangan ini mengeksploitasi bagian-bagian dari algoritma yang spesifik terhadap ilmu *watermarking* dan merusak template yang digunakan seperti pola sinkronisasi, segingga detektor tidak bisa mengidentifikasi tipe dari *affine transformation* yang diterapkan. Sejak pendekatan template berusaha mereduksi jumlah informasi yang dimasukkan sehingga membuat template tidak terlihat, serangan ini menjadi sangat efisien. Serangan ini memiliki beberapa pengaruh berdasarkan rating yang bagus di dalam Stirmark benchmark untuk template based *watermarking* system. Jika sebuah teknologi tidak bisa bertahan dari template *attack*, maka user yang berperan sebagaai pemilik karya cipta harus menghadapi ancaman dan resiko.

Watermark Template *attack* merupakan serangan yang diperkenalkan oleh Herrigel, Voloshynovskiy dan Rytsar pada tahun 2001. Serangan ini tergolong atas *maulicious attack* karena tidak mengetahui sisi algoritmanya, dan *Geometrical attack*. Template *attack* mengerang pola sinkronisasi dalam data yang ter*watermark*, sehingga *watermark* tidak dapat dideteksi lagi.
Template adalah pola sinkronisasi untuk mengestimasi dan mengarahkan transform. Beberapa template sangat tergantung paada key.

Template yang diaplikasikan pada saat proses marking sangat spesifik terhadap teknologi *watermark*. Ada beberapa strategi untuk beberapa pembentukan template. Kita tidak butuh untuk mengetahui bagaimana sebuah template yang spesifik dalam sebuah domain dikonstruksi dari sisi prespektif counterfeiter, dikarenakan template yang diaplikasikan juga selalu menggenerasi beberapa peak di target domain. Beberapa pendekatan menggunakan template dalam 25 poin. Poin-poin tersebut adalah benuk yang terdistribusi di dalam domain DFT tanpa memasukkan frekuensi yang rendah. Frekuensi yang rendah tidak dimasukkan disebabkan karena mengandung *bulk* dari *spectral power* dan merepresentasikan *noise* saat decoding. Kekuatan dari template adalah adaptif.

Bagaimana mendeteksi sebuah template? Beberapa pendekatan dilakukan untuk mentransformasikan template matching problem menjadi point matching problem. Setelah problem ini berhasil, beberapa kandidat utama dari poin template akan diidentifikasi. Jika sebuah transformasi telah diaplikasikan, template yang teridentifikasi akan berbeda dengan yang original. Perubahan ini dieksploitasi untuk mengestimasi transformasi yang diaplikasikan. Kesesuaian transformasi inverse kemudian diaplikasikan untuk sinkronisasi dari *watermark.*

7.3.2 Langkah-Langkah dan Ujicoba

Skema dari serangan ini dapat dibagi menjadi dua fase sebagai berikut:

1. Fase 1
    - Baca gambar yang di*watermark* dan aplikasikan ke dalam median filter (Wiener filter juga bisa) untuk mengestimasi gambar asli
    - Substract gambar asli dari gambar yang telah ter*watermark* dan simpan hasilnya
2. Fase 2
    - Hitung discrete Fourier transform
    - Identifikasi peak maksimum
    - Modifikasi amplitudo dari maksimum peak dengan menggantikan amplitudo yang spesifik dengan amplitudo rata-rata dalam gambar
    - Hitung discrete Fourier transform setelah ampliturdo dari peak-peak telah dimodifikasi dan tuliskan hasilnya ke dalam file baru. File tersebut adalah gambarh yang telah diserang

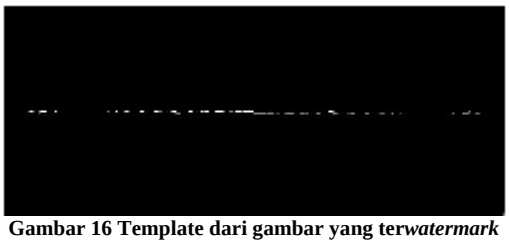
Serangan ini memanfaatkan hasil di dalam domain *watermarking*. Dikarenakan hanya ada sedikit skema *watermark* yang memiliki deteksi kuat untuk *watermark* 64 bit melawan transformasi seperti penskalaan dan rotasi. Beberapa ujicoba yang dilakukan terhadap serangan mungkin memecahkan proses deteksi menjadi beberapa teknologi *watermarking* jika tidak ada transformasi yang dilakukan. Gambar berikut merupakan salah satu contoh yang mengilustrasikan:

Gambar 12 Gambar asli

Gambar 13 Gambar yang telah diberi *watermark*

Gambar 14 Template dari gambar yang telah ter*watermark*

Gambar 15 Gambar yang telah ter*watermark* setelah template *attack*



Gambar 16 Template dari gambar yang ter*watermark* setelah template *attack*

Gambar 17 Gambar yang telah terotasi setelah serangan

Gambar 18 Capture dari pendeteksian yang gagal

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Template Removal Implementation
Attack test program for watermark technologies
Gray & color images.
Version 4.0, September 2000
Copyright (C) 2000, DCT, Zurich, Switzerland
All rights reserved
Legal relevant notice:
Authorized program execution only
for testing purposes.
Any other usage is not authorized
and prohibited.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
*** Watermarked:  (445 x 188), 83660 pixels  ***
Template Removal.....50%....(515).....100%
Output File Name -----> removed_wmdigi100usd.bmp
Output File Name -----> test11.jpg
```

Gambar 19 Screen Capture dari Program Template Attack

Pada gambar 1 terlihat gambar asli yang belum ter-*watermark*. Gambar berikutnya adalah gambar yang telah diberikan *watermark*. Gambar berikutnya menunjukkan template dari gambar yang ter*watermark*. Gambar 4 menunjukkan gambar yang sudah diserang dengan template *attack*. Gambar 5 menunjukkan tempate setelah diserang, Gambar 6 menunjukkan gambar yang telah diserang lalu dirotasi. Gambar ketujuh menunjukkan bahwa program Photoshop 5 yang digunakan oleh Herrige dkk untuk melakukan deteksi tidak dapat mendeteksi *watermark* itu lagi. Kualitasi dari gambar setelah serangan dapat ditingkatkan dengan menggunakan denoising dan teknik-teknik restorasi.

Herrige dkk telah melakukan beberapa ujicoba untuk menyerang gambar dan kebanyakan hasilnya berhasil. Ujicoba tersebut dilakukan terhadap beberapa sistem *watermarking*dari akademisi dan industri. Kadangkalah malah hasil dari serangan tidak butuh untuk ditransformasikan.

Beberapa contoh gambar lain yang berhasil dibuka dengan template *attack*:

Gambar 20 Contoh-contohgambar yang bisa dipecahkan dengan template attack

8. Evaluasi dari Serangan-Serangan

Dari hasil di atas dapat dilihat bahwa serangan-serangan yang tergolong *maulicious attack* bisa menghilangkan *watermark* dari gambar. Hal ini menunjukkan bahwa kerahasiaan algoritma dan kunci untuk melakukan penyisipan pesan sudah tidak terlalu penting lagi, karena *watermark*

biasa akan dapat dengan mudah dihilangkan dengan teknik-teknik di atas.

Dengan Copy *Attack*, gambar yang merupakan *watermark* dapat diambil dari sebuah gambar dan dapat digunakan untuk menjadi *watermark* di gambar lainnya.

Untuk menghilangkan *watermark* pada sebuah gambar, juga dapat dilakukan estimasi seperti dilakukan oleh *Blind Estimation*, atau dengan serangan geografis oleh *Template Attact*.

Serangan dari template attack sendiri memiliki beberapa pengaruh berdasarkan rating yang bagus di dalam Stirmark benchmark untuk template based *watermarking* system. Jika sebuah teknologi tidak bisa bertahan dari template *attack*, maka user yang berperan sebagaai pemilik karya cipta harus menghadapi ancaman dan resiko.Teknologi watermarking harus menjadikannya standar untuk menunjukkan sebuah algoritma pantas atau tidak dengan mengujinya melalui serangan ini.

9. Kesimpulan

Dari studi mengenai *watermarking* ini dapat diperoleh kesimpulan-kesimpulan sebagai berikut:

1. Watermarking merupakan proses penyisipan *watermark* ke dalam file digital untuk tujuan menjaga keaslian data tersebut
2. Sebuah watermark yang baik harus dapat memenuhi Imperceptibility, Key uniqueness, Noninvertibility, Image Dependency, dan Robustness
3. Serangan terhadap algoritma watermarking dapat digunakan untuk tujuan buruk, namun saat ini ilmunya terus dikembangkan untuk perkembangan ilmu watermarking yang baik pula. Dengan kata lain, keberadaan ilmu serangan-serangan ini akan menjadi evaluasi terhadap ilmu watermarking.
4. Secara umum, ada empat jenis serangan yang digunakan di dalam menyerang *image watermarking*. Keempatnya adalah *Removal Attacks*, *Geometrical Attacks*, *Cryptographic Attacks*, dan *Protocol Attacks*. Selain itu terdapat pengkategorian jenis serangan yang lain yaitu *Maulicious Attack dan Non Maulicious Attack*
5. *Watermark* Copy *Attack* merupakan serangan baru yang termasuk jenis dari *Protocol Attack*. Konsep dari serangan terdiri atas melakukan salinan dari *stego image* menuju *terget image* tanpa menggunakan informasi spesifik tentang teknologi yang digunakan untuk menyimpan informasi
6. *Blind Estimation Without Priors* merupakan sebuah teknik yang dapat diterapkan untuk melakukan estimasi *watermark* yang ada di dalam sebuah gambar
7. *Watermark* template *attack* adalah sebuah serangan untuk mengatasi template based *watermarking* systems. Serangan ini mengeksploitasi bagian-bagian dari algoritma yang spesifik terhadap ilmu *watermarking* dan merusak template yang digunakan .
8. *Watermark template attack* membuat teknologi watermarking harus mengacu pada standar untuk tidak bisa dipecahkan oleh metode ini. Jika sebuah algoritma watermarking masih bisa dipecahkan dengan ini, maka algoritma tersebut tidak aman.

10. Daftar Pustaka

Referensi Utama

1. Munir, Rinaldi. (2004). Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.
2. Alexander Herrigel, Sviatoslav Voloshynovskiy, and Yuriy Rytsar**,** The Watermark Template Attack, SPIE, 2001
3. M. Kutter, S. Voloshynovskiy and A. Herrigel, “Watermark copy attack,” In Ping Wah Wong and Edward J. Delp eds., IS&T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Content II, Vol. 3971 of SPIE Proceedings, San Jose, California USA, 23-28 January 2000. (Paper EI 3971-35)

4. Jiang Du, Choong-Hoon, Lee Heung-Kyu, Lee Youngho Suh, Watermark attack based on Blind Estimation Without Priors, 2002.

Referensi yang terkait dengan referensi utama

1. I. Cox, J. Kilian, T. Leighton, T. Shamoon, "Secure spread spectrum *watermarking* for multimedia", *IEEE Trans Image. Processing*, **6**, No 12, pp. 1673-1687, Dec. 1997.

2. G. Langelaar, R. Lagendijk, and J. Biemond, "Removing spatial spread spectrum *watermark*s by non-linear filtering", *in Proc. EUSIPCO98*, **4**, pp. 2281-2284, 1998.

3. J. Su and B. Girod, "Power spectrum condition for L2-efficient *watermarking*", *Submitted to IEEE Int Conf. Image Processing ICIP99*, October 1999.

4. M. Kutter, *Digital image watermarking: hiding information in images*, PhD thesis, EPFL, Lausanna, Switzerland, 1999.

5. J. Hernandez, F. Perez-Gonzalez, J. Rodriguez, "Performance analysis of a 2-D-multipulse amplitude modulation scheme for data hiding and *watermarking* of still images", *Proc. IEEE Journal on Selected Areas in Communications*, **16**, No 4, pp. 510-524, May 1998.

6. J.P. Linnartz, G. Depovere, T. Kalker, "On the design of a *watermarking* system: cosideration and rationals", 3rd *Workshop on Information Hiding*, Dresden, Germany, Sept. 29-Oct. 1, 1999, to appear in Lecture notes in Computer Science.

7. M. Kutter, "*Watermarking* resisting to translation, rotation, and scaling", *Proc. of SPIE, Multimedia systems and* applications, **3528**, pp. 523-531, San Jose, USA, November 1998.

8. S. Voloshynovskiy, F. Deguillaume, T. Pun, "Content adaptive *watermarking* based on a stochastic multiresolution image modeling", submitted EUSIPCO'2000.

9. Voloshynovskiy, A. Herrigel, N. Baumgartner, T. Pun, "A stochastic approach to content adaptive digital image *watermarking*", 3rd Workshop on Information Hiding, Dresden, Germany, Sept. 29-Oct. 1, 1999, to appear in Lecture notes in Computer Science.

10. J. Hernandez and F. Perez-Gonzalez, "Statistical analysis of *watermarking* schemes for copyright protection on images", *Proc. IEEE*, **87**, No 7, pp. 1142-1166, July 1999.

11. S. Kassam, *Signal detection in non-Gaussian noise*, Spriner-Verlag, New York, 1998.

12. S. Voloshynovskiy, "Robust Image Restoration Based on Concept of M-Estimation and Parametric Model of Image Spectrum", in *Proc. IEEE, IEE, EURASIP 5th International Workshop on Systems, Signals and Image Processing ``IWSSIP'98''*, pp. 123-126, Zagreb, Croatia, June 1998.

13. D. Donoho and I. Johnstone, "Ideal spatial adaptation via wavelet shrinkage", *Biometrika*, **81**, pp. 425-455, 1994.

14. P. Moulin and J. Liu, "Analysis of Multiresolution Image Denoising Schemes Using Generalized Gaussian and
15. Complexity Priors", *Proc. IEEE on Information Theory*, **45**, No3, pp. 909-919, April 1999.

16. N. Saito, "Simultaneous noise suppression and signal compression using a library of orthnormal bases and the MDL criterion", *in Wavelets in Geophysics*, E. Foufoula-Georgiou and P. Kumar, , pp. 299-324, Eds. New York: Academic, 1995.

17. B. Natarajan, "Filtering random noise from deterministic signals via data compression", *IEEE Trans. on Signal Processing*, **43**, No 10, pp. 2595-2605, November 1995.

18. J. Liu and P. Moulin, “Complexity-regularized image denoising”, in *Proc. IEEE Int. Conf. Image Processing ICIP97*, **2**, Santa Barbara, CA, pp. 370-373, 1997 S. Chang, B. Yu, and M. Vetterli, “Spatially adaptive wavelet thresholding with context modeling for image denosing”, *Submitted to IEEE Trans. on Image Processing*, 1998, (http://gabor.eecs.berkley.edu/~grchang/publications.html)

19. Markus G. Kuhn and Fabien A. P. Petitcolas. StirMark. <http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark/>, November 1997.

20. M. Kutter and F. Petitcolas. A fair benchmark for image watermarking systems. In Ping Wah Wong and Edward J. Delp, editors, Proceedings of the SPIE, Security and Watermarking of Multimedia Contents, volume 3657, pages 226{239, San Jose, CA, USA, January 1999. IS&T, The Society for Imaging Science and Technology and SPIE, The International Society for Optical Engineering, SPIE.

21. Sviatoslav Voloshynovskiy, Shelby Pereira, Alexander Herrigel, Nazanin Baumgärtner and Thierry Pun, Generalized watermark attack based on watermark estimation and perceptual remodulation, In Ping Wah Wong and Edward J. Delp eds., *IS&T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Content II*, Vol. 3971 of SPIE Proceedings, San Jose, California USA, 23-28January 2000. (Paper EI 3971-34)

22. Sviatoslav Voloshynovskiy, Alexander Herrigel, Frédéric Jordan, Nazanin Baumgärtner and Thierry Pun, A noise removal attack for watermarked images, In J. Dittmann, K. Nahrstedt and P. Wohlmacher eds., *Multimedia and Security Workshop*, Orlando, Florida, USA, 30-31 October 1999. (at the 7th ACM Multimedia Conference (Multimedia 99))

23. Svyatoslav Voloshynovskiy, Shelby Pereira, Victor Iquise and Thierry Pun, Attack modelling: Towards a second generation benchmark, *Signal Processing, Special Issue: Information Theoretic Issues in Digital Watermarking*, 2001.

24. V. Cappellini, M. Barni, F. Bartolini, Eds. Digimarc Corporation, US patent 5,822,436, Photographic Products And Methods Employing Embedded Information.

25. Shelby Pereira, Joseph J. K. O Ruanaidh, Frederic Deguillaume, and Thierry Pun.Template Based Recovery of Fourier- Based Watermarks Using Log-polar and Log-log Maps, IEEE Int. Conf. on Multimedia Computing and Systems, ICMCS‘99 Florence, Italy, June 1999.