Marcas d'Água Frágeis de Autenticação para Imagens em Tonalidade Contínua e Esteganografia para Imagens Binárias e Meio-Tom

Hae Yong Kim

Universidade de São Paulo

Escola Politécnica

1. Introdução

Esteganografia ("information hiding" ou "steganography"): estudo de como inserir dados escondidos na imagem.

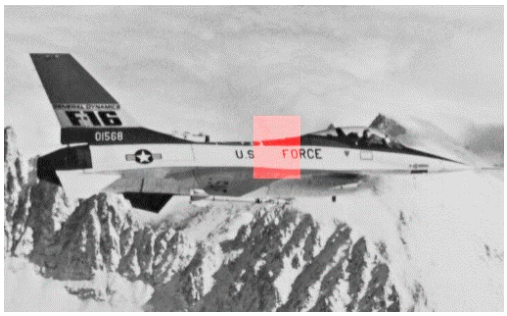
Marca d'água (MDA): dado visualmente imperceptível inserida numa imagem, normalmente para atestar propriedade ou autenticar imagem.

Tipos de MDA:

1. Robustas: MDA difícil de ser removida. Usada para atestar copyright.
2. Frágeis (de autenticação): MDA facilmente removível. Identifica o produtor da imagem e detecta alterações na imagem.
3. Semi-frágeis: MDA de autenticação resistente a algumas operações: compressão com perdas, ajuste de brilho/contraste, etc.

Tipos de MDA de autenticação:

1. Sem chave: Similar ao código de autenticação de mensagem ("check-sum"). Detecta alterações não-intencionais.
2. Chave secreta: A mesma chave é usada na inserção e na detecção.
3. Chave pública/privada: Usa chave privada para inserção e chave pública para verificação.



MDA de autenticação para imagens binárias não está muito desenvolvida.

Unidades deste tutorial:

1. (Seção 2) - conceitos básicos: assinatura digital e meio-tom.
2. (seção 3) - MDA de autenticação para imagens contone (tonalidade contínua, i.e., níveis de cinza e colorida).
3. (seção 4) - esteganografia para imagens binárias e halftone.

Este tutorial não tem a pretensão de esgotar o assunto.

2. Conceitos Preliminares

2.1 Assinatura digital (AD)

Criptografia de chave secreta ou simétrica: requer que o transmissor e o receptor da mensagem possuam a mesma chave.

Criptografia de chave pública ou cifra assimétrica: chave pública para criptografar e chave privada para decriptografar.

Conhecendo a chave privada, é fácil calcular a chave pública. O contrário é extremamente difícil computacionalmente.

- **Alice quer enviar uma mensagem secreta para Bob.**
- **Alice usa chave pública do Bob para criptografar a mensagem.**
- **Bob usa a sua chave privada para decriptografar.**

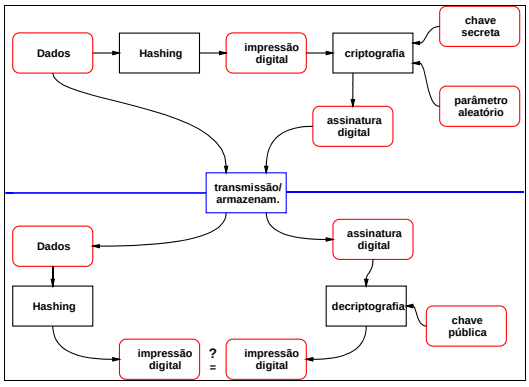
Implementando esse processo "ao contrário", obtém-se AD:

- **Alice quer demonstrar que foi ela que enviou a mensagem.**
- **Alice usa a sua chave privada para criptografar uma mensagem.**
- **Bob usa a chave pública da Alice para decriptografar a mensagem.**
- **Somente Alice poderia ter enviado essa mensagem.**

Detalhando AD:

- **Alice quer "assinar eletronicamente" um documento (isto é, demonstrar que ela gerou o documento e ninguém o alterou).**
- **Alice calcula a "impressão digital" do documento através de uma função hashing *H*. (Uma função hashing calcula representantes curtos para mensagens arbitrariamente longas).**
- **Alice criptografa a "impressão digital" do documento usando a sua chave privada, obtendo AD.**
- **Alice envia AD e o documento para Bob.**
- **Bob decriptografa AD usando a chave pública da Alice.**

- **Bob calcula novamente a "impressão digital" do documento, usando a função hashing.**
- **Se as duas impressões digitais são iguais, a assinatura está autenticada. Isto é, o documento foi assinado pela Alice e ninguém alterou o documento.**



AD:

- ◊ **Determinística**
- ◊ **Não-determinística**

Uma AD determinística pode tornar-se não-determinística acrescentando dados aleatórios na mensagem.

2.2 Meio-tom por difusão de erro

Meio-tom converte uma imagem em níveis de cinza *G* numa imagem binária *B* de modo que *B* parece *G* quando visto de certa distância.

Dada $G: \mathbb{Z}^2 \rightarrow [0,1]$, meio-tom deve gerar $B: \mathbb{Z}^2 \rightarrow \{0,1\}$ tal que para todos (i, j):

$$B(i,j) \cong G(i,j).$$

Meio-tom por difusão de erro:

Para todos os pixels (i, j) da imagem *G* {
Se $(G(i, j) < 0{,}5)$ $B(i, j) \leftarrow 0$; senão $B(i, j) \leftarrow 1$;
erro $\leftarrow G(i, j) - B(i, j)$;
$G(i, j+1) \leftarrow G(i, j+1) + \text{erro} \times \text{alpha}$;
$G(i+1, j-1) \leftarrow G(i+1, j-1) + \text{erro} \times \text{beta}$;
$G(i+1, j) \leftarrow G(i+1, j) + \text{erro} \times \text{gamma}$;
$G(i+1, j+1) \leftarrow G(i+1, j+1) + \text{erro} \times \text{delta}$;
}

$\left[\frac{1}{16}\right] \times \begin{bmatrix} \bullet & 7 \\ 3 & 5 & 1 \end{bmatrix}$ **Floyd e Steinberg**	$\begin{bmatrix} \bullet & \alpha \\ \beta & \gamma & \delta \end{bmatrix}$ **Floyd e Steinberg**
$\left[\frac{1}{48}\right] \times \begin{bmatrix} & & \bullet & 7 & 5 \\ 3 & 5 & 7 & 5 & 3 \\ 1 & 3 & 5 & 3 & 1 \end{bmatrix}$ **Jarvis, Judice e Ninke**	$\left[\frac{1}{42}\right] \times \begin{bmatrix} & & \bullet & 8 & 4 \\ 2 & 4 & 8 & 4 & 2 \\ 1 & 2 & 4 & 2 & 1 \end{bmatrix}$ **Stucki**

Matrizes de pesos populares

Imagem em níveis de cinza *G*

Imagem halftone *B*

3. MDA de Autenticação para Imagens Contone

3.1 Câmera digital confiável de Friedman

Câmera confiável [Friedman, 1993] inspirou os primeiros trabalhos de MDA de autenticação.

Câmera produz dois arquivos:

1. **Imagem digital;**
2. **AD da imagem.**

3.2 Melhoramentos da câmera de Friedman

3.2.1 Embutir AD na imagem - um arquivo

- ◊ No cabeçalho ou rodapé da imagem.
- ◊ Na própria imagem (MDA).

3.2.2 Permitir localizar a região alterada.

- ◊ Localizar a região alterada permite descobrir as intenções do falsificador.

3.3 MDA de Yeung-Mintzer

[Yeung, 1997]: uma das primeiras técnicas de MDA de autenticação:

- ◊ MDA inserida pixel a pixel.
- ◊ 1 bit de autenticação por pixel: 50% de chance de alteração de 1 pixel passar despercebida.
- ◊ Esquema com chave secreta.
- ◊ Demonstrou-se que é inseguro: é possível inserir MDA em qualquer imagem dispondo de uma imagem com MDA válida.

3.3.1 Inserção de MDA de Yeung-Mintzer

- ◊ I: imagem hospedeira em níveis de cinza;
- ◊ B: logo binário (mesmo tamanho que I);
- ◊ $k : \{0...255\} \rightarrow \{0,1\}$: look-up-table (LUT) - funciona como chave secreta;
- ◊ I': imagem marcada.

Processamento dos pixels em ordem "raster".

Inserção de MDA no primeiro pixel (1, 1):

- ◊ Calcula-se $k(I(1,1))$;
- ◊ Se $k(I(1,1)) = B(1,1)$, nada a fazer;
- ◊ Se $k(I(1,1)) \neq B(1,1)$, $I(1,1)$ é alterado para um nível de cinza próximo $I'(1,1)$ para obter $k(I'(1,1)) = B(1,1)$.
- ◊ O erro $I(1,1) - I'(1,1)$ é espalhado para os pixels vizinhos (semelhante à difusão de erro).
- ◊ Este processo se repete até processar a imagem toda.

Para inserir MDA numa imagem colorida I:

k_R, k_G e k_B: LUTs dos planos de cor R, G e B.
Para inserir MDA num pixel (i, j), calcula:

$$k_R(\bar{I}_R(i,j)) \otimes k_G(\bar{I}_G(i,j)) \otimes k_B(\bar{I}_B(i,j))$$

$\bar{I}_R(i,j)$, $\bar{I}_G(i,j)$ e $\bar{I}_B(i,j)$: valor do pixel (i, j) obtido difundindo o erro cometido ao aproximar I pela I'.

- Se a expressão = $b(i,j)$, nada a fazer.
- Se for diferente, $\bar{I}_R(i,j)$, $\bar{I}_G(i,j)$ e/ou $\bar{I}_B(i,j)$ devem ser alterados para valores próximos $I'_R(i,j)$, $I'_G(i,j)$ e $I'_B(i,j)$.

3.3.2 Extração de MDA de Yeung-Mintzer

I': imagem com MDA;
k: LUT usada como chave secreta;
B: logo binário inserido;
C: Imagem binária de checagem a ser extraída.

Para I' em níveis de cinza, calcule:

$$C(i,j) \leftarrow k(I'(i,j)).$$

Para imagem I' colorida, calcule:

$$C(i,j) \leftarrow k_R(I'_R(i,j)) \otimes k_G(I'_G(i,j)) \otimes k_B(I'_B(i,j)).$$

- Se C=B, I' não foi alterada.
- Caso contrário, houve alteração onde C e B forem diferentes.

3.3.3 Ataque de falsificação

[Holliman, 2000]: counterfeiting attack.

Dispondo de algumas imagens marcadas com chave secreta k, é possível colocar MDA em qualquer imagem (sem conhecer k).

É possível descobrir a chave k a partir de algumas imagens marcadas com k.

Vamos explicar caso em níveis de cinza.

Mallory quer inserir MDA em J sem conhecer k. Mallory conhece B e tem acesso a uma imagem I' marcada usando k.

Mallory subdivide os pixels de I' em S_0 (pixels com valor 0 em B) e S_1. Mallory provavelmente conhece todas as entradas de LUT k (só tem 256 entradas e uma imagem tem da ordem de 10^6 pixels).

Conhecendo k, pode marcar qualquer imagem.
Conhecendo k parcialmente, também é possível marcar qualquer imagem.

3.4 MDA de Wong

[Wong, 1997] propôs MDA baseada em criptografia simétrica (chave secreta).

[Wong, 1998] propôs MDA baseada em criptografia assimétrica (chave pública). Primeira MDA de chave pública.

- MDA de Wong: divide imagem em blocos e assina cada bloco independentemente.
- Permite localizar bloco alterado.
- MDA inserida nos bits menos significativos (LSB - least significant bit).
- Vários bits de autenticação por bloco: impossível uma alteração passar despercebida.
- Esquema com chave-secreta ou chave-pública.
- O esquema original é inseguro. Mas melhoramentos posteriores tornam-no seguro.

3.4.1 Inserção de MDA de Wong

1. I: imagem em níveis de cinza $N \times M$ a ser marcada. Particione I em n blocos I_t de 8×8 pixels.
2. B: logo binário (mesmo tamanho que I). Para cada I_t, existe um bloco B_t.
3. I_t^*: I_t com LSBs zerados. Calcule a impressão digital $H_t = H(M,N,I_t^*)$ (M e N para detectar cortes (cropping)).
4. $\hat{H}_t = H_t \otimes B_t$.
5. Criptografe $\hat{H}_t$ com chave privada, gerando AD S_t do bloco t.
6. Insira S_t nos LSBs de I_t^*, obtendo I_t'.

3.4.2 Extração de MDA de Wong

1. I': imagem $N \times M$ marcada. Particione I' em blocos 8×8 I_t'.
2. I_t^*: I_t' com LSBs zerados. Calcule a impressão digital $H_t = H(M,N,I_t^*)$.
3. Retire os LSBs de I_t' e decriptografe-o (usando chave pública), obtendo D_t.
4. $C_t = H_t \otimes D_t$.
5. Se C_t = B_t, MDA verificada. Caso contrário, I' foi alterada no bloco t.

3.5 Ataques à MDA de Wong

RSA de 64 bits é completamente inseguro.

Um esquema seguro de autenticação tem que detectar *qualquer* alteração.

[Wong,1998] generaliza o esquema para imagens coloridas aplicando o método nos 3 planos de cores. Não detecta permutação dos planos.

3.5.1 Ataques recortar-e-colar e falsificação

Se copiar um bloco, MDA de Wong não detecta (ataque recortar-e-colar).

Se repetir ataque recortar-e-colar, uma imagem inteira falsificada pode ser construída (araque de falsificação de Holliman-Memon).

- **Mallory quer marcar uma imagem J.**
- **Mallory tem um bando de dados de imagens com MDA de Wong. Imagem-logo B é conhecido publicamente.**
- **Mallory particiona J em blocos J_t.**
- **Para cada bloco J_t (onde se deve inserir B_t) Mallory procura no banco de dados o bloco D'_t parecido com J_t.**
- **Insere D'_t no lugar de J_t.**
- **Repete o processo para todos os blocos de J.**

3.5.2 Ataques de aniversário simples

Hashing com m possíveis valores $\Rightarrow$ se houver $\sqrt{m}$ blocos, há mais de 50% de chance de colisão (achar 2 blocos com impressões digitais iguais).

Função de hashing de 64 bits $\Rightarrow$ se houver 2^{32} (4 bilhões) blocos assinados, consegue substituir um bloco por outro.

BD com um milhão de imagens 640×480, particionado em blocos 8×8: 4 bilhões de blocos.

Para substituir bloco I_t pelo bloco forjado J_t:

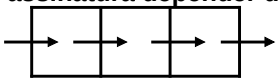
1) Dispor de BD com $\sqrt{m}$ assinaturas válidas.
2) Gerar $\sqrt{m}$ variantes visuais de J_t (altera o segundo bit menos significativo).

Provavelmente haverá colisão de J_t com um bloco B_t do BD.

Substitui I_t pelo J_t com AD pega de B_t.

3.5.3 Hash block chaining versão 1

Idéia: Fazer assinatura depender do contexto.



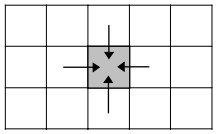
Uma dependência por bloco para aumentar resolução de localização.

HBC1:

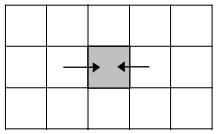
$$H_t \equiv H(M, N, I_t^*, I_{(t-1) \bmod n}^*, t)$$

Obs1: Se B_t for alterado, B_t e $B_{(t+1) \bmod N}$ serão relatados como inválidos.
Obs2: A capacidade de localização é perdida pela inserção e/ou remoção.

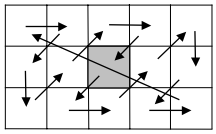
Recortar-e-colar é impossível em HBC1.
Ataque de aniversário simples é impossível.



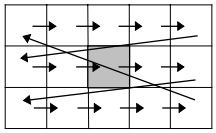
a) 4 deps. por bloco



b) 2 deps. por bloco



c) 1 dep. por bloco (zig-zag)

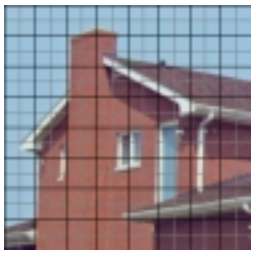


d) 1 dep. por bloco (raster)

a) Imagem original

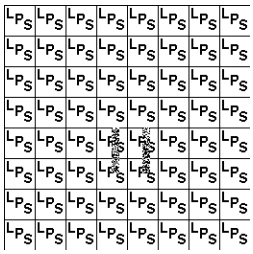
b) Imagem-logotipo 32×32

c) Imagem marcada com HBC2



d) Blocos 16×16 e 32×32

e) Ataque recortar e colar



f) Delimitação das alterações

3.5.4 Ataques de transplante

X' e $\overline{X}'$ duas imagens marcadas com HBC1.

$$\cdots \rightarrow X'_A \rightarrow X'_D \rightarrow X'_B \rightarrow X'_C \rightarrow \cdots$$
$$\cdots \rightarrow \overline{X}'_A \rightarrow \overline{X}'_E \rightarrow \overline{X}'_B \rightarrow \overline{X}'_C \rightarrow \cdots$$

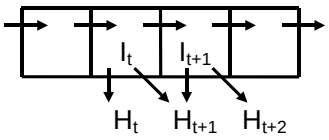
HBC1 não detecta:

$$\cdots \rightarrow X'_A \rightarrow \overline{X}'_E \rightarrow \overline{X}'_B \rightarrow X'_C \rightarrow \cdots$$
$$\cdots \rightarrow \overline{X}'_A \rightarrow X'_D \rightarrow X'_B \rightarrow \overline{X}'_C \rightarrow \cdots$$

Documentos têm muitos blocos brancos.
Não adiante aumentar dependências.

3.5.5 Ataques de aniversário melhorado

Forjar simultaneamente I_t e I_{t+1} (substituí-los por J'_t e J'_{t+1}).



3.5.6 Hash block chaining versão 2

Usa assinatura digital não-determinística.
Faz impressão digital depender da AD:

$$H_t \equiv H(M,N,I_t^*,I_{(t-1) \bmod n}^*,t,S_{t-1})$$

HBC2 resiste a todos os ataques.
Detecta qualquer alteração:

- **Bloco modificado ou rearranjado: localiza região de alteração (2 blocos).**
- **Bloco apagado ou inserido: indica alteração (mas não localiza).**
- **Se uma grande região é copiada, indica as bordas.**

3.5.8 MDA Wong-Memon

Necessita um identificador $\mathtt{I}_I$ único para cada imagem I:

$$H_t \equiv H(\mathtt{I}_I,M,N,I_t^*,t)$$

$\mathtt{I}_I$ deve ser armazenado de alguma forma: pouco prático.

4. Esteganografia para Imagens Binárias e Halftone

- **Como estender MDA de autenticação para imagens binárias?**
- **Imagem binária/halftone não possuem LSB. Como embutir informação?**
- **Se embutir informação, altera a impressão digital.**
- **As pesquisas MDA binária/halftone estão num estágio anterior à MDA de autenticação.**

Veremos alguns exemplos.

4.1 Imagem visível com sobreposição

Esconde uma imagem binária em duas imagens meio-tom. Se duas imagens forem sobrepostas, aparece a imagem binária escondida.

Primeiras idéias usando algoritmo de meio-tom ordered dithering patenteadas [Knox, 1998; Wang, 1998].

[Fu, 2001]: versão para meio-tom pela difusão de erro.
[Pei, 2003]: melhora o algoritmo de Fu e Au.

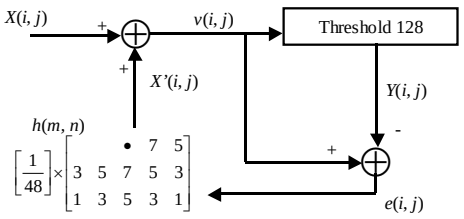


Fig. 3: Difusão de erro padrão.

$$\begin{cases} v(i,j) = X(i,j) + X'(i,j), \\ X'(i,j) = \sum_{m=0}^{2}\sum_{n=-2}^{2} e(i+m, j+n)h(m,n) \\ e(i,j) = v(i,j) - Y(i,j) + N_B, \\ Y(i,j) = \begin{cases} 0, & \text{se } v(i,j) < 128 \\ 255, & \text{se } v(i,j) \geq 128 \end{cases} \end{cases} \quad (4.1)$$

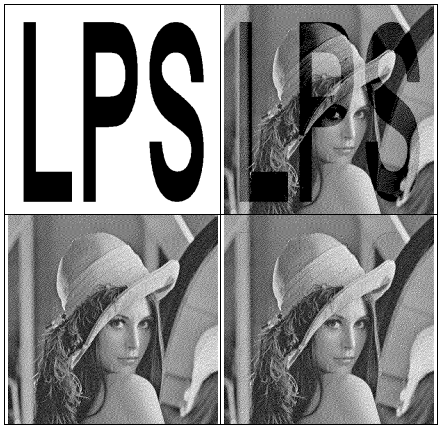
Y_1: meio-tom de X obtida difusão de erro normal.
Y_2: meio-tom contendo logo B.

$B(i, j)$ preto e $Y_1(i, j)$ branco: $Y_2(i, j)$ é calculado usando (4.2) em vez da (4.1):

$$\begin{cases} v(i,j) = X(i,j) + X'(i,j) - N_B \\ e(i,j) = v(i,j) - Y(i,j) + N_B \end{cases} \quad (4.2)$$

$B(i, j)$ preto e $Y_1(i, j)$ preto: utiliza-se (4.1).
$B(i, j)$ branco e $Y_1(i, j)$ preto: utiliza-se (4.2).
$B(i, j)$ branco e $Y_1(i, j)$ branco: utiliza-se (4.2) invertendo os sinais de N_B.

O valor da N_B em (4.2) controla a qualidade de Y_2 e do B decodificado.
Se N_B aumentar, Y_2 piora e B torna-se mais nítido.



Generalizações:

1. **Aumentar o número de meio-tons para 3 ou mais. Aumenta nitidez de B.**
2. **Y_1 e Y_2 são meio-tons de imagens em níveis de cinza diferentes. Diminui nitidez de B. Foi sugerida uma técnica para enfatizar bordas.**
3. **Autodecodificação de B (B escondido numa única imagem). A imagem aparece quando sobrepõe duas cópias de Y deslocadas.**

4.2 Modificação de pixels individuais

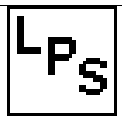
[Fu, 2000; Fu, 2002]: DHST (data hiding by self toggling). Embute dados em meio-tom ponto disperso.

Inserção:
- **Gerador de número pseudo-aleatório (com semente conhecida) gera um conjunto de posições na imagem.**
- **Um bit é embutido em cada posição forçando-o a ser ou preto ou branco.**

Leitura:
- **Gerador pseudo-aleatório gera novamente o mesmo conjunto de posições na imagem.**
- **Basta ler os valores dos pixels nessas posições.**

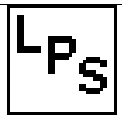
Se DHST for utilizado em imagem binária, aparecerá ruído sal-e-pimenta visível.

Imagem original



Logo a ser inserido

1024 bits embutida (DHST)



Logo extraído

Melhoramentos:

- DHPT: Data hiding by pair toggling. Mudança de um pixel acompanhada pela mudança complementar de um dos 8 vizinhos (para manter o nível de cinza médio constante).
- DHPST: Data hiding by smart pair toggling. Entre os vizinhos possíveis de serem mudados, escolhe aquele que gera padrão visual agradável.

4.3 Modificação de características de bloco

A imagem binária ou halftone é dividida em blocos e insere bits por bloco mudando a característica.

4.3.1 Número de pixels pretos por bloco

[Wu, 2000]:

- Divide imagem binária em blocos 8×8.
- Bloco com pixels pretos par → bit 0.
- Bloco com pixels pretos ímpar → bit 1.
- Os pixels têm "score" visual. Altera pixel com score mais baixo.

Pode-se escolher blocos de tamanhos diferentes, inserir mais de um bit por bloco, etc.

Problema:

- Algum bloco pode ser inteiramente branco, inteiramente preto, etc.

Solução:

- Fazer "shuffling" (permutação aleatória) dos pixels.

Problema:

- Não dá para usar para MDA de autenticação.

4.3.2 Alterar matriz de pesos da difusão de erro

[Pei,2003; Hel-Or, 2001]:
É possível descobrir a matriz de pesos usada na difusão de erro analisando Transformada de Fourier do bloco.

Associa cada matriz de pesos a um bit. Em cada bloco, usa uma matriz de pesos diferente.

5. Conclusão

Neste tutorial vimos:

1. **Visão panorâmica das pesquisas sobre MDA e esteganografia;**
2. **Dois conceitos básicos: assinatura digital e meio-tom por difusão de erro;**
3. **Principais técnicas de MDA de autenticação para imagens contone; ataques; mecanismos de defesa;**
4. **Algumas técnicas para embutir dados em imagens binárias e meio-tom.**

Estudamos algumas técnicas. Não temos a pretensão de esgotar o assunto.

Referências que não aparecem em RITA:

[Wu, 2000] M. Wu, E. Tang, and B. Liu, "Data Hiding in Digital Binary Image," *IEEE Int. Conf. on Multimedia and Expo*, 2000, New York, USA.