Steganography and Steganalysis:
past, present, and future

Anderson Rocha
anderson.rocha@ic.unicamp.br

Siome Goldenstein
siome@ic.unicamp.br

Institute of Computing
University of Campinas (Unicamp)
CEP 13084-851, Campinas, SP - Brazil
Summary

- Steganography
  - LSB insertion/modification
  - FFTs and DCTs
- How to improve security
Summary

» Steganalysis
  • Aural
  • Structural
  • Statistical
Summary

- Freely available tools and software
- Open research topics
- Conclusions and remarks
Steganography
Hiding scenario

\[ + = \]
Steganography

- Computer Vision and Image Processing techniques
- Mostly based on replacing a noise component
Steganography
Steganography

- What are the problems of noise embedding?
Steganography

- What are the problems of noise embedding?
  - Compression
Steganography

- What are the problems of noise embedding?
  - Compression
  - Filtering
Steganography

- What are the problems of noise embedding?
  - Compression
  - Filtering
  - Conversions
Steganography

- What are the problems of noise embedding?
  - Compression
  - Filtering
  - Conversions
- MSB-based techniques
Steganography techniques

LSB insertion/modification

Extracting image's bit channels

Given a 24-bit typical color image, we want to extract its 8 bit channels
Steganography techniques

LSB insertion/modification
Steganography techniques

**FFTs and DCTs based**

1. Least significant coefficients
   - JSteg and Outguess
2. Block tweaking
3. Coefficient selection
4. Wavelets
Steganography techniques

**FFTs and DCTs based**

1. **Splitting.** Split up the image into 8x8 blocks.
2. **Transformation.** Transform each block via a DCT/FFT.
3. **Compression stage 1.** Use a quantizer to round the coefficients.
4. **Compression stage 2.** Use a Huffman encoding scheme or similar to further compress the streamlined coefficients.
5. **Decompressing.** Use inverse DCT/FFT to decompress.

**DCT and FFT general algorithm**
Steganography techniques

FFTs and DCTs

- JSteg
  - Sequentially replaces LSB of DCT/FFT coefficients
  - Does not use shared key
  - What is its main problem?
Steganography techniques

FFTs and DCTs

**JSteg general algorithm**

```plaintext
Require: message M, cover image I;
1: JSteg(M,I)
2:   while M != NULL do
3:     get next DCT coefficient from I
4:     if DCT != 0 and DCT != 1 then
5:       b = next bit from M
6:       replace DCT LSB with message bit b
7:       M = M - b
8:     end if
9:     Insert DCT into stego image S
10:    end while
11:   return S
12: end procedure
```
Steganography techniques

**FFT**s and **DCT**s

- **Outguess**
  - Improvement over JSteg
  - PRNG
  - Statistical profiling
Steganography techniques

**FFTs and DCTs**

**Outguess general algorithm**

**Require**: message $M$, cover image $I$, shared key $k$;

1: Outguess($M$,$I$,$k$)
2: Initialize PRNG with the shared key $k$
3: while $M$ != NULL do
4: get pseudo-random DCT coefficient from $I$
5: if DCT != 0 and DCT != 1 then
6: $b$ = next bit from $M$
7: replace DCT LSB with message bit $b$
8: $M = M - b$
9: end if
10: Insert DCT into stego image $S$
11: end while
12: return $S$
13: end procedure
Steganography techniques

**FFTs and DCTs**

2. Block tweaking

- DCT/FFT’s *quantizer* stage
- Keeps down distortions
- Vulnerable to *noise*
- Low-capacity embedding
Steganography techniques

**FFTs and DCTs**

- **Coefficient selection**
  - Selects $k$ largest DCT/FFT coefficients
  - Use a function $f$ that considers the required strength of the embedding process

\[
f(\gamma') = \gamma_i + \alpha b_i
\]

$b_i$ is the bit you want to embed in the coefficient
Steganography techniques

FFT's and DCT's

- Wavelets
  - DCT/FFT transformations are not effective at higher-compression levels
  - Possibility to embed in the high-frequency
  - Embedding in the quantization stage
Steganography techniques

How to improve security

- Kerckhoff’s Principle
- Destruction of the original
- Statistical profiling
Steganography techniques

How to improve security

- Structural profiling
- Split the information
- Compaction
Steganalysis
Steganalysis

- Detection of hidden messages
- Early approaches focused on detection
- Next step: recovery
Steganalysis

- Steganalysis attacks
  1. Aural
  2. Structural
  3. Statistical
Statistical Steganalysis

$\chi^2$ Analysis

- An $L$-bit color channel represent $2^L$ possible values
- Split in $2^{L-1}$ pairs differing in the LSBs only
- All possible patterns of neighboring bits for the LSBs

$$PoV : 0 \leftrightarrow 1, 2 \leftrightarrow 3, \ldots, 254 \leftrightarrow 255$$

Statistical Steganalysis

$\chi^2$ Analysis

- What if we use all available LSBs?
- Expected frequency vs observed one
- Expected frequency is not available
- In the original the EF is the arithmetical mean in each PoV
Statistical Steganalysis

$\chi^2$ Analysis

- The embedding affects only the LSBs
- Arithmetical mean remains the same in each PoV
- $\chi^2$ to detect hidden messages

$$\chi^2 = \sum_{i=1}^{\nu+1} \frac{(f_i^{obs} - f_i^{exp})^2}{f_i^{exp}}$$
Statistical Steganalysis

$\chi^2$ Analysis

- Probability of hiding

$$p_h = 1 - \int_0^{\chi^2} \frac{t^{(\nu-2)/2}e^{-t/2}}{2^{\nu/2}\Gamma(\nu/2)} dt$$
Statistical Steganalysis

$\chi^2$ Analysis

- Only detects sequential messages
- The **threshold** value for detection may be quite distinct for different images
- Low-order statistics
RS Analysis (RS)

- Analysis of the LSB loss-less embedding capacity
- The LSB plane is correlated with other bit planes
- Simulates artificial new embeddings

Let $I$ be the image with $W \times H$ pixels

Pixel values in $P = \{1 \ldots 255\}$

Divide $I$ in $G$ disjoint groups of $n$ adjacent pixels (e.g., $n = 4$)
Define a discriminant function to classify the $G$ groups

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$$
Statistical Steganalysis

RS Analysis (RS)

- **Flipping** invertible function
  
  \[ F_1 : 0 \leftrightarrow 1, 2 \leftrightarrow 3, \ldots, 254 \leftrightarrow 255 \]

- **Shifting** invertible function
  
  \[ F_{-1} : -1 \leftrightarrow 0, 1 \leftrightarrow 2, \ldots, 255 \leftrightarrow 256 \]

- **Identity** function
  
  \[ F_0(x) : x \forall x \in P \]
RS Analysis (RS)

- Define a mask $M = \{-1, 0, 1\}$
- The mask defines which function to apply
- The mask’s compliment is $-M$
Apply the functions over the groups for $M$ and -$M$ masks. Classify them as

- **Regular.** $G \in R_M \iff f(F_M(G)) > f(G)$
- **Singular.** $G \in S_M \iff f(F_M(G)) < f(G)$
- **Unusable.** $G \in U_M \iff f(F_M(G)) = f(G)$
It holds that

\[
\frac{R_M + S_M}{T} \leq 1 \quad \text{and} \quad \frac{R_{-M} + S_{-M}}{T} \leq 1,
\]

Statistical hypothesis

\[
R_M \approx R_{-M} \quad \text{and} \quad S_M \approx S_{-M}
\]
Gradient Energy Flipping Rate (GEFR)

- Gradient of an unidimensional signal
  \[ r(n) = I(n) - I(n - 1) \]
- The I(n)'s GE is
  \[ GE = \sum |I(n) - I(n - 1)|^2 = \sum r(n)^2 \]

Gradient Energy Flipping Rate (GEFR)

- After hiding a signal $S(n)$ in the original signal, $I(n)$ becomes $I'(n)$ and the gradient becomes

$$r(n) = I(n) - I(n - 1) = (I(n) + S(n)) - (I(n - 1) + S(n - 1)) = r(n) + S(n) - S(n - 1)$$
Gradient Energy Flipping Rate (GEFR)

- After any kind of embedding GE' becomes

\[ GE' = \sum |r(n) + \Delta(n)|^2 \]

where \( \Delta(n) = S(n) - S(n - 1) \)
Gradient Energy Flipping Rate (GEFR)

- To perform the detection, define a function to simulate new embeddings
Gradient Energy Flipping Rate (GEFR)

1. Find the test image’s \( GE \left( \frac{p/2}{W \times H} \right) \)
2. Apply F over the test image and calculate \( GE \left( \frac{W \times H - p/2}{W \times H} \right) \)
3. Find \( GE \left( \frac{W \times H}{2} \right) = \left[ EG \left( \frac{p/2}{W \times H} \right) + GE \left( \frac{W \times H - p/2}{W \times H} \right) \right] / 2 \)
4. \( GE(0) \) is based on \( GE \left( \frac{W \times H}{2} \right) = GE(0) + W \times H \)
5. Find the message’s estimated size \( p' = GE \left( \frac{p/2}{W \times H} \right) - GE(0) \)

GEFR general algorithm
High-order Statistical analysis

- Natural images have regularities
- They can be detected with high-order statistics
- Use QMF decomposition for multi-scale analysis

High-order Statistical analysis

Statistical Steganalysis

QMF decomposition
High-order Statistical analysis

- Let $V_i(x,y)$, $H_i(x,y)$, and $D_i(x,y)$ be the vertical, horizontal, and diagonal sub-bands for a given scale $i = \{1, \ldots, n\}$

- Statistical model composed by Mean, Variance, Skewness, and Kurtosis

- Basic coefficients distribution
High-order Statistical analysis

- Second set of statistics
  - Errors on an optimal linear predictor of coefficient magnitude
  - Spatial, orientation, and scale neighborhood
High-order Statistical analysis

- For instance: errors for all neighbors in the vertical sub-band at scale $i$

$$V_i(x, y) = w_1 V_i(x-1, y) + w_2 V_i(x+1, y) + w_3 V_i(x, y-1) +$$
$$w_4 V_i(x, y+1) + w_5 V_{i+1}(x/2, y/2) + w_6 D_i(x, y) + w_7 D_{i+1}(x/2, y/2)$$

- $w_k$ denotes scalar weighting values
High-order Statistical analysis

- **Quadratic minimization** of the error function
  
  \[ E(w) = [V - Qw]^2 \]

- \( V \) is a column vector of magnitude coefficients
- \( Q \) is the magnitude neighbors’ coefficients
Statistical Steganalysis

High-order Statistical analysis

- **Minimization** through differentiation wrt $w$

\[
\frac{dE(w)}{dw} = 2Q^T[V - Qw]
\]

- Calculate $w_k$ using the **linear predictor log error**

\[
\log_2(V) - \log_2(|Qw|)
\]
High-order Statistical analysis

- $12(n-1)$ basic statistics
- $12(n-1)$ error statistics
- $24(n-1)$ feature vector
Statistical Steganalysis

High-order Statistical analysis

- Supervised learning
- Training set of stego and clean images
- LDA and SVMs
Image Quality Metrics (IQMs)

- Often used for
  - Coding artifact evaluation
  - Performance prediction of vision algorithms
  - Quality loss due to sensor inadequacy

Image Quality Metrics (IQMs)

- IQMs
- Multivariate regression analysis (ANOVA)
- Exploits Steganographic schemes artifacts
Image Quality Metrics (IQMs)

- IQMs
  1. Mean absolute error
  2. Czekwnowski correlation
  3. Image fidelity
  4. HVS error
  5. etc
Image Quality Metrics (IQMs)

- Training set of stego and clean images
- ANOVA

\[
\begin{align*}
y_1 &= \beta_1 x_{11} + \beta_2 x_{12} + \ldots + \beta_q x_{1q} + \epsilon_1 \\
y_2 &= \beta_2 x_{21} + \beta_2 x_{22} + \ldots + \beta_q x_{2q} + \epsilon_2 \\
&\vdots \\
y_N &= \beta_n x_{n1} + \beta_2 x_{12} + \ldots + \beta_q x_{nq} + \epsilon_n,
\end{align*}
\]
Statistical Steganalysis

Progressive Randomization (PR)

- It captures the differences between image classes
- **Statistical artifacts** inserted during the hiding process

Progressive Randomization (PR)

- **Four stages**
  1. Randomization process
  2. Feature regions selection
  3. Statistical descriptors analysis
  4. Invariance
Progressive Randomization (PR)

- The idea behind PR
- Let $X$ be a Bernoulli RV
- Transformation $T(l, p)$

\[
L(px_i) \leftarrow b_i \forall px_i \in S
\]
Statistical Steganalysis

Progressive Randomization (PR)

**Require:** Input image $I$; Percentage $P = \{P_i, \ldots, P_n\}$;

1: **Randomization:** perform $n$ LSB pixel disturbances on $I$

\[
\{O_i\}_{i=0 \ldots n} = \{I, T(I, P_1), \ldots, T(I, P_n)\}
\]

2: **Region selection:** select $r$ feature regions of each image $i \in \{O_i\}_{i=0 \ldots n}$

\[
\{O_{ij}\}_{i=0 \ldots n, j=1 \ldots r} = \{O_{01}, \ldots, O_{nr}\}.
\]

3: **Statistical descriptors:** calculate $m$ descriptors for each region

\[
\{d_{ijk}\} = \{d_k(O_{ij})\}_{i=0 \ldots n, j=1 \ldots r, k=1 \ldots m}.
\]

4: **Invariance:** normalize the descriptors based on $I$

\[
F = \{f_e\}_{e=1 \ldots n \times r \times m} = \left\{ \frac{d_{ijk}}{d_{0jk}} \right\}_{i=0 \ldots n, j=1 \ldots r, k=1 \ldots m}.
\]

**Progressive Randomization algorithm**
Progressive Randomization (PR)

- Randomization stage
  - It simulates new embeddings
  - $n = 6$
  - $P = \{1\%, 5\%, 10\%, 25\%, 50\%, 75\%\}$ of the LSBs
Progressive Randomization (PR)

- Statistical descriptors stage
  - $\chi^2$
  - Ueli Maurer that measures randomness

Statistical Steganalysis
Progressive Randomization (PR)
Progressive Randomization (PR)

- **Invariance stage**
  - The variation rate is more interesting
  - Normalize all transformation’s result \((T_1...T_n)\) wrt. \(T_0\)
Progressive Randomization (PR)

Classification stage

- Training set of stego and clean images
- Supervised learning
- $|M| = 25\% \text{ (~13\% changed LSBs)} > 90\%$ accuracy (SVMs)
Progressive Randomization (PR)

Progressive Randomization
Normalizing the feature vector

We use six progressive randomization steps;
We normalize the descriptors' values of each stage by their value in the original image

\[ \text{Norm}(O_i) = \frac{d_i(O_i)}{d_i(I)}, \]

where \( O_i \) refers to the \( i \)-th output of the Progressive Randomization Stage.

We have a 96-dimensional feature space
Software and tools
Software and tools

- EzStego
- Stego Online
- Mandelsteg
- Stealth
Software and tools

- White Noise
- S-Tools
- Hide and Seek
- JSteg
- Outguess
Software and Tools
Camaleão

www.ic.unicamp.br/~rocha/sci/stego
Interesting research topics
Steganography

Open research topics

- Images are subjected to many operations
  - Translation, rotation, shear
  - Blurring, filtering, lossy compression
  - Printing, rescanning, conversion
Open research topics

- Designing of robust IH techniques
  - Robustness to geometrical attacks
  - Embeddings in regions with richness of details
Open research topics

- Good IQMs
- Public key systems
- Multiple embeddings with no interference
Open research topics

- Blind detection
- Very small embedding detection
- Adaptive techniques
- Hidden content recovery
Conclusions
Conclusions

- Steganography and Steganalysis overview
- IH embedding and detection techniques
- Open research topics
Conclusions

- Data hiding has passed its period of hype
- Public fear created by mainstream press reports
- Laws against IH techniques dissemination
Conclusions

- Nowadays...
  - Steganography and Steganalysis are mature disciplines
  - Applications
  - Research opportunities
Conclusions
Steg in real world
Questions?

*The thinker by Rodin*