Steganography and Steganalysis of JPEG Images

Ph.D. Proposal
Mahendra Kumar
CISE Department

Chair: Dr. Richard Newman
Co-Chair: Dr. Jonathan Liu
Outline

- Intro to Steganography and Steganalysis.
- JPEG Steganography
- Steganalysis Techniques
- J2 – Topological approach to Steganography
- J3 – Histogram neutral JPEG Steganography
- Contribution
- Future Work
  - Steganography using second order statistics restoration.
  - Steganalysis using second order statistics estimation.
What is Steganography?

- Hide data inside a cover medium
- Existence of any communication is undetectable.
- Has an edge over cryptography, does not attract any public attention.
- **Cover medium**: the medium without any message embedded.
- **Stego medium**: medium with message embedded.

![Diagram of Steganography process]

- **Secret Message**
- **Shared Secret Key**
- **Redundant Data**
- **Cover Image**
- **Stego Image**
## Watermarking vs. Steganography

<table>
<thead>
<tr>
<th>Steganography</th>
<th>Watermarking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal is stealthiness</td>
<td>Goal is robustness</td>
</tr>
<tr>
<td>Existence of message is unknown</td>
<td>Sometimes existence of message is known</td>
</tr>
<tr>
<td>Higher data capacity</td>
<td>Lower data capacity</td>
</tr>
<tr>
<td>One to one communication hiding</td>
<td>One to many communication hiding</td>
</tr>
<tr>
<td>Eavesdropper cannot detect presence of data</td>
<td>Eavesdropper cannot detect or remove data</td>
</tr>
<tr>
<td>Secret communication between two agents. Private data in medical imaging, anonymous communication.</td>
<td>Tracking copyright, fingerprinting, access control information for DRM.</td>
</tr>
</tbody>
</table>
JPEG Compression

- Most popular image format used.
Since compression is lossy, data embedding in spatial domain will result in too much noise.

Solution: Hide data before the entropy coding stage.
**JPEG Steganography - LSB Embedding**

- Hide data by changing the LSB of JPEG coefficient.
- Most common technique.

```
<table>
<thead>
<tr>
<th></th>
<th>-26</th>
<th>-3</th>
<th>-6</th>
<th>2</th>
<th>3</th>
<th>2</th>
<th>-1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>-4</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-3</td>
<td>-2</td>
<td>1</td>
<td>5</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-4</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Bit Pos: 1 2 3
Message: 1 0 1
```
Popular Algorithms - JSteg

Before

After

**cover coeff.**

**Bit value to embed**

**Stego coeff.**

**Stego message bit value**
Popular Algorithms – F5

- Uses password driven permutation using pseudo random generator.
- Matrix encoding to reduce number of changes.
  - embed $K$ bits by changing one of $2^K - 1$ places.
- Resistant against Chi-square attack
- Absolute value of coefficient is decreased by 1 to change a bit.
  - Changes a lot of 1s and -1s to zeros.
- Ignores zero coefficient.
Popular Algorithms – Outguess

- Changes LSB to embed data.
- Part of coefficients reserved for restoration of changed coefficients.
- Cover histogram same as stego histogram.
- Skips 1s and 0s.
- Uses a error threshold to determine the amount of change tolerated.
- Might not completely restore the histogram.
- Performs poorly with large number of coefficients.
Steganalysis

- Hide and seek game. Aims to detect the presence of data in a medium (image).
- Primary goal is to focus more on detecting statistical anomalies.
  - Most Steganography algorithms avoid visual distortions.
- Modification to a typical cover image will lead to statistical distortion of some kind.
  - Global histogram, blockiness, inter/intra block dependencies.
Steganalysis- Types

- **Specific Steganalysis**
  - Designed to attack one particular algorithm
  - Steganalyst is aware of the embedding method.
  - Compare the statistical trend of the stego images for that algorithms with natural JPEG images.
  - Examples- JSteg, F5.
Steganalysis- Types

- **Universal Steganalysis**
  - Also know as blind steganalysis.
  - More powerful and modern approach
  - Does not depend on knowing the particular embedding algorithm.
  - Based on finding first and second order statistics- also called features.
  - Uses a pattern classifier to train the cover and stego images from their features.
    - Predict unknown images by extracting its features.
    - Stego files from different algorithms have to be trained and classified before being used for detection.
  - Markov model based approach (intra block correlation), individual mode histograms, inter block correlation, combined inter/intra block correlation
Detecting JSteg

- Uses chi-square attack to detect typical histogram change.
- Can be categorized as first order statistical attack.
Detecting F5

- Proposed by Fridrich et al.
- Decompress the given stego image to spatial domain.
- Crop the image by 4 rows and 4 columns.
- Recompress the cropped image.
- The cropped image is an estimation of the cover image.
- Calibrate the cropped image to remove artifacts.
- Compare the statistics of cropped image with the stego image.
  - Blockiness, global histogram, individual histograms

JPEG → BMP → Cropped BMP → Cropped JPEG

Statistics → Calibrated statistics → Statistics

Compare
Detecting Outguess

- Since first order statistics are restored.
  - Cannot be detected using chi-square technique.
- Use second order statistics.
- Use a pattern recognition classifier.
- We will come back to this detection technique later.
Pattern Recognition Classifier

- Takes an unknown variable and predicts which class the variable belongs to.
- Has to be trained with a given data set from different classes.
- Support vector machine (SVM) is the most common pattern classifier.
  - Based on the training set, its builds a prediction model.
  - Usually 50% for training and 50% for testing.
SVM Classifier

Non-liner classifier

Linear classifier

SVM tries to find a hyper-plane which separates the two classes by a maximum distance.
Steganalysis Using Markov Model

- Detects intra-block dependency anomalies.
- Calculate the difference matrices.

\[
F_h(u,v) = F(u,v) - F(u+1,v) \\
F_v(u,v) = F(u,v) - F(u,v+1) \\
F_d(u,v) = F(u,v) - F(u+1,v+1) \\
F_m(u,v) = F(u+1,v) - F(u,v+1)
\]
Steganalysis Using Markov Model

- Calculate the transition probability matrices (TPM).
- Use the TPM as features for SVM classifier.

---

**Image:**

- A grid showing horizontal transitions and their counts.
- A bar chart illustrating the counts of horizontal transitions.
Transition Probability Matrix

Transition Probabilities:

- From state 0: 3/10 to state 1, 4/10 to state 2
- From state 1: 5/13 to state 0, 3/10 to state 2
- From state 2: 5/7 to state 1, 2/7 to state 0

Transition Matrix:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>
Detection rate of various algorithms using Markov based features.

<table>
<thead>
<tr>
<th></th>
<th>bpc</th>
<th>TN</th>
<th>TP</th>
<th>AR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outguess</td>
<td>0.05</td>
<td>87.6</td>
<td>90.1</td>
<td>88.9</td>
</tr>
<tr>
<td>Outguess</td>
<td>0.1</td>
<td>94.6</td>
<td>96.5</td>
<td>95.5</td>
</tr>
<tr>
<td>Outguess</td>
<td>0.2</td>
<td>97.2</td>
<td>98.3</td>
<td>97.8</td>
</tr>
<tr>
<td>F5</td>
<td>0.05</td>
<td>58.6</td>
<td>57.0</td>
<td>57.8</td>
</tr>
<tr>
<td>F5</td>
<td>0.1</td>
<td>68.1</td>
<td>70.2</td>
<td>69.1</td>
</tr>
<tr>
<td>F5</td>
<td>0.2</td>
<td>85.8</td>
<td>88.3</td>
<td>87.0</td>
</tr>
<tr>
<td>F5</td>
<td>0.4</td>
<td>95.9</td>
<td>97.6</td>
<td>96.8</td>
</tr>
<tr>
<td>MB1</td>
<td>0.05</td>
<td>79.4</td>
<td>82.0</td>
<td>80.7</td>
</tr>
<tr>
<td>MB1</td>
<td>0.1</td>
<td>91.2</td>
<td>93.3</td>
<td>92.3</td>
</tr>
<tr>
<td>MB1</td>
<td>0.2</td>
<td>96.7</td>
<td>97.8</td>
<td>97.3</td>
</tr>
<tr>
<td>MB1</td>
<td>0.4</td>
<td>98.8</td>
<td>99.4</td>
<td>99.1</td>
</tr>
</tbody>
</table>
J2- A Topological Approach To JPEG Steganography

- Makes changes to JPEG coefficient in frequency domain.
- Embeds data in spatial domain.
- Threshold to determine which blocks are usable.
- Hash the spatial data bytes to find if it matches the message bits.
- Embeds k number of bits per block.

![Image](image-url)

- Convert to spatial matrix:
  
  $\begin{bmatrix}
  -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\
  0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\
  -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\
  -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\
  1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 
  \end{bmatrix}$

- Hash it with key, K:
  
  $\begin{bmatrix}
  60 & 63 & 55 & 58 & 70 & 61 & 58 & 80 \\
  58 & 56 & 56 & 83 & 108 & 88 & 63 & 71 \\
  60 & 52 & 62 & 113 & 150 & 116 & 70 & 67 \\
  66 & 56 & 68 & 122 & 156 & 116 & 69 & 72 \\
  69 & 62 & 65 & 100 & 120 & 86 & 59 & 76 \\
  68 & 68 & 61 & 68 & 78 & 60 & 53 & 78 \\
  74 & 82 & 67 & 54 & 63 & 64 & 65 & 83 \\
  83 & 96 & 77 & 56 & 70 & 83 & 83 & 89 
  \end{bmatrix}$

- Compare the LSBs with Message bits.
J2- Embedding And Extraction Algorithms

**Embedding**

1. 63 quantized DCT coefficients in block, F
2. Is number of zeros in F < Thr
   - Yes: i = random number between 1 and 63
   - No: F(i) = F(i) ± 1
3. Convert block F to spatial domain, S
4. Hash S with Key K, H(S|K)
5. Does n LSBs of H(S|K) = next n bits of Message M?
   - No: Go to next random DCT block in order of visitation
   - Yes: Go to next random DCT block in order of visitation

**Extraction**

1. Quantized DCT block, F
2. Is number of zeros in F < Thr
   - Yes: Convert block F to spatial domain, S
   - No: Go to next random DCT block in order of visitation
3. Hash S with Key K, H(S|K)
4. n LSBs of H(S|K) = next n bits of Message M
5. Go to next random DCT block in order of visitation
Randomly changing a coefficient by +/- 1 can be expected to remove many more zeros than it adds.
Hence number of 1s and -1s will increase in number and zeros will decrease.
J3- High Payload Histogram Neutral JPEG Steganography

- Completely restores the histogram to its original values.
- Optimizes the use of coefficients to maximize capacity.
- Coefficients are always changed in pairs.
  - $(2x, 2x+1)$ form a pair
  - $2x$ will always increase to $2x+1$ if needed to change.
  - $2x+1$ will always decrease to $2x$ if needed to change.
  - 1 is changed to -1 and vice versa. (to maximize capacity)
- Uses stop points to determine when to stop encoding
J3 Continued

- Algorithm keeps track of changes made.
- If just enough coefficients remain to restore the histogram for that coefficient, it stops encoding that pair.
- The index of that position is stored as stop point for that pair.
- J3 uses header data to store stop points and other information.
- Matrix encoding is used to minimize the changes.

<table>
<thead>
<tr>
<th>4 Bits</th>
<th>20 Bits</th>
<th>5 Bits</th>
<th>5 Bits</th>
<th>((N_{SP}N_{BSP})) Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value of (n) for Matrix encoding, (H_n)</td>
<td>Data Length in Bytes, (M_L)</td>
<td>No. of bits required to store a single stop point, (N_{bSP})</td>
<td>No. of stop points, (N_{SP})</td>
<td>Stop point array, (SP(-2n, -2n-1) \ldots SP(-2, -3), SP(-1, 1), SP(2, 3) \ldots SP(2n, 2n+1))</td>
</tr>
</tbody>
</table>
J3 - Example

- Hist(2) = 500, Hist(3) = 200

- During embedding, assume the following:
  - Changed(2->3) = 100, Changed(2->2)= 100
  - Changed(3->2) = 50, Changed(3->3)= 100
  - Remaining(2)= 500 - (100+100) = 300
  - Remaining(3)= 200 – (50 +100) = 50
  - 100 2s have been changed to 3, only 50 3s have been changed to 2. Hence, 50 more 3s and 50 less 2s.
  - Hence imbalance in 2 = -50
  - Imbalance in 3 = +50
  - We cannot encode any more data in pair (2,3).
  - Only enough 3s remain to convert back to 2.
1. Header data bits are embedded at the end of embed process, since all stop points are not known in the beginning.
2. Coefficients for the header bits are reserved in the beginning.
1. Header data bits are always extracted in the beginning.
2. Stop points are extracted and stored.
3. If an index reaches a value of stop point, that pair of coefficient is not decoded after that.
J3 - Theoretical Stop-point Estimation

Estimated Stop Point = \[ y_{2x,2x+1} = \frac{Hist(2x+1)}{p_{m,1} \cdot (p_{c,2x} + p_{c,2x+1})} \]

Where,

\[ k_{total} = \sum_{x=2}^{coeff \_limit} Hist(x) \]

\[ p_{c,2x+1} = \frac{Hist(2x+1)}{k_{total}} \]

\[ p_{c,2x} = \frac{Hist(2x)}{k_{total}} \]
J3- Theoretical Capacity Estimation

Estimated Capacity =

\[ C_{est} = \min \left\{ C_{est,1}, C_{est,2} \right\} \]

Where,

\[ C_{total,1} = \frac{\sum_{x=1}^{\text{coeff limit}} \left( \text{Hist}(2x) + \text{Hist}(-2x) \right) + \text{Hist}(-1)}{p_{m,0}} \quad \text{Bits} \]

\[ C_{total,2} = \frac{\sum_{x=1}^{\text{coeff limit}} \left( \text{Hist}(2x+1) + \text{Hist}(-2x-1) \right) + \text{Hist}(1)}{p_{m,1}} \quad \text{Bits} \]
J3: Lena Image Statistics

cover

stego
J3: Histogram Of Lena Image
J3: Estimated Capacity Vs Actual Capacity
J3: Estimated Stop-point Vs Actual Stop-point

\[
\gamma_{2x,2x+1} = \frac{Hist(2x+1)}{p_{m,1} \cdot (p_{c,2x} + p_{c,2x+1})}
\]
J3: Embedding Efficiency (Bpp)
J3: Embedding Efficiency (Bpnz)
J3: Embedding Efficiency (Bepcc)
J3: Capacity Comparison With Other Algorithms
J3: Steganalysis Performance

- SVM classifier with RBF (Radial basis function) kernel was used.
- 274 merged Markov and DCT features were used as data for each image.
- 1000 JPEG images for training and testing.
- All the images were embedded with random data using J3, F5, Outguess and Steghide algorithms.
  - Hence we have 5000 images.
  - 1000 cover, 1000 outguess, 1000 J3 and so on.
- 70% images were used for training and rest 30% for testing.
  - i.e. 700 cover and 700 stego images from each algorithm.
- Training and testing sets were randomized 100 times.
J3: Binary Classification

Prediction accuracy

Prediction accuracy
J3: Binary Classification

- Cover with one of the algorithms were used for training and prediction.

| Embedding Algorithm capacity | Al- (100%) | Classifiers as (%) | | Embedding Algorithm capacity | Al- (100%) | Classifiers as (%) |
|-----------------------------|------------|-------------------| |-----------------------------|------------|-------------------|
| Cover                       | 98.50      | 1.50              | | Cover                       | 99.86      | 0.14              |
| J3                          | 1.71       | **98.29**         | | F5                          | 0.11       | **99.89**         |
| Classified as (%)           |            |                   | | Classified as (%)           |            |                   |

<table>
<thead>
<tr>
<th>Embedding Algorithm capacity</th>
<th>Classifiers as (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover</td>
<td>99.61</td>
</tr>
<tr>
<td>Outguess</td>
<td><strong>99.89</strong></td>
</tr>
<tr>
<td>Classified as (%)</td>
<td></td>
</tr>
<tr>
<td>Classified as (%)</td>
<td></td>
</tr>
</tbody>
</table>

100% message length
## J3: Binary Classification

### Classified as (%)

<table>
<thead>
<tr>
<th>Embedding Algorithm (50% capacity)</th>
<th>Cover</th>
<th>J3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover</td>
<td>97.23</td>
<td>2.77</td>
</tr>
<tr>
<td>J3</td>
<td>3.71</td>
<td><strong>96.29</strong></td>
</tr>
</tbody>
</table>

### Classified as (%)

<table>
<thead>
<tr>
<th>Embedding Algorithm (50% capacity)</th>
<th>Cover</th>
<th>Outguess</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover</td>
<td>99.39</td>
<td>0.61</td>
</tr>
<tr>
<td>Outguess</td>
<td>0.36</td>
<td><strong>99.64</strong></td>
</tr>
</tbody>
</table>

### Classified as (%)

<table>
<thead>
<tr>
<th>Embedding Algorithm (25% capacity)</th>
<th>Cover</th>
<th>J3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover</td>
<td>84.75</td>
<td>15.25</td>
</tr>
<tr>
<td>J3</td>
<td>15.36</td>
<td><strong>84.64</strong></td>
</tr>
</tbody>
</table>

### Classified as (%)

<table>
<thead>
<tr>
<th>Embedding Algorithm (25% capacity)</th>
<th>Cover</th>
<th>Outguess</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover</td>
<td>98.24</td>
<td>1.76</td>
</tr>
<tr>
<td>Outguess</td>
<td>1.40</td>
<td><strong>98.60</strong></td>
</tr>
</tbody>
</table>

### Classified as (%)

<table>
<thead>
<tr>
<th>Embedding Algorithm (50% capacity)</th>
<th>Cover</th>
<th>F5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover</td>
<td>99.35</td>
<td>0.65</td>
</tr>
<tr>
<td>F5</td>
<td>0.54</td>
<td><strong>99.46</strong></td>
</tr>
</tbody>
</table>

### Classified as (%)

<table>
<thead>
<tr>
<th>Embedding Algorithm (50% capacity)</th>
<th>Cover</th>
<th>Steghide</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover</td>
<td>99.82</td>
<td>0.18</td>
</tr>
<tr>
<td>Steghide</td>
<td>0.11</td>
<td><strong>99.89</strong></td>
</tr>
</tbody>
</table>

### Classified as (%)

<table>
<thead>
<tr>
<th>Embedding Algorithm (25% capacity)</th>
<th>Cover</th>
<th>F5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover</td>
<td>95.54</td>
<td>4.46</td>
</tr>
<tr>
<td>F5</td>
<td>4.24</td>
<td><strong>95.76</strong></td>
</tr>
</tbody>
</table>

### Classified as (%)

<table>
<thead>
<tr>
<th>Embedding Algorithm (25% capacity)</th>
<th>Cover</th>
<th>Steghide</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover</td>
<td>97.41</td>
<td>2.59</td>
</tr>
<tr>
<td>Steghide</td>
<td>18.27</td>
<td><strong>81.73</strong></td>
</tr>
</tbody>
</table>
J3: Multi-Classification

Training Set

Randomize

Cover
J3
F5
Outguess
Steghide

Randomize

Testing Set

Prediction accuracy
J3: Multi-Classification

- Images from different algorithms were used together for training and classification.

<table>
<thead>
<tr>
<th>Embedding Algorithm (capacity)</th>
<th>Al- (100%)</th>
<th>A -</th>
<th>F5</th>
<th>OutGuess</th>
<th>Steghide</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover</td>
<td>97.93</td>
<td>1.79</td>
<td>0.04</td>
<td>0.14</td>
<td>0.11</td>
</tr>
<tr>
<td>J3</td>
<td>2.02</td>
<td><strong>97.55</strong></td>
<td>0.18</td>
<td>0.14</td>
<td>0.11</td>
</tr>
<tr>
<td>F5</td>
<td>0</td>
<td>0.05</td>
<td>99.64</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>OutGuess</td>
<td>0.02</td>
<td>0.18</td>
<td>0.09</td>
<td>98.96</td>
<td>0.75</td>
</tr>
<tr>
<td>Steghide</td>
<td>0.16</td>
<td>0.25</td>
<td>0.16</td>
<td>0.48</td>
<td>98.95</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Embedding Algorithm (50% capacity)</th>
<th>Cover</th>
<th>J3</th>
<th>F5</th>
<th>OutGuess</th>
<th>Steghide</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover</td>
<td>97.01</td>
<td>2.23</td>
<td>0.45</td>
<td>0.20</td>
<td>0.11</td>
</tr>
<tr>
<td>J3</td>
<td>3.09</td>
<td><strong>96.01</strong></td>
<td>0.63</td>
<td>0.22</td>
<td>0.05</td>
</tr>
<tr>
<td>F5</td>
<td>0.43</td>
<td>0.21</td>
<td>99.32</td>
<td>0.03</td>
<td>0.01</td>
</tr>
<tr>
<td>OutGuess</td>
<td>0.27</td>
<td>0.20</td>
<td>0.36</td>
<td>98.31</td>
<td>0.86</td>
</tr>
<tr>
<td>Steghide</td>
<td>0.13</td>
<td>0.16</td>
<td>0.26</td>
<td>0.64</td>
<td>98.80</td>
</tr>
</tbody>
</table>
### J3: Multi-classification

#### Classified as (%)

<table>
<thead>
<tr>
<th>Embedding Algorithm (25% capacity)</th>
<th>Cover</th>
<th>J3</th>
<th>F5</th>
<th>OutGuess</th>
<th>Steghide</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover</td>
<td>81.16</td>
<td>14.22</td>
<td>2.42</td>
<td>0.83</td>
<td>1.46</td>
</tr>
<tr>
<td>J3</td>
<td>15.37</td>
<td>80.48</td>
<td>2.55</td>
<td>0.69</td>
<td>0.91</td>
</tr>
<tr>
<td>F5</td>
<td>4.14</td>
<td>1.85</td>
<td>93.11</td>
<td>0.25</td>
<td>0.65</td>
</tr>
<tr>
<td>OutGuess</td>
<td>1.13</td>
<td>0.63</td>
<td>0.92</td>
<td>93.59</td>
<td>3.73</td>
</tr>
<tr>
<td>Steghide</td>
<td>15.39</td>
<td>3.42</td>
<td>2.23</td>
<td>3.75</td>
<td>75.21</td>
</tr>
</tbody>
</table>

#### Classified as (%)

<table>
<thead>
<tr>
<th>Embedding Algorithm (Equal capacity)</th>
<th>Cover</th>
<th>J3</th>
<th>F5</th>
<th>OutGuess</th>
<th>Steghide</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover</td>
<td>96.43</td>
<td>1.58</td>
<td>0.99</td>
<td>0.04</td>
<td>0.96</td>
</tr>
<tr>
<td>J3</td>
<td>3.56</td>
<td>94.71</td>
<td>1.27</td>
<td>0.03</td>
<td>0.44</td>
</tr>
<tr>
<td>F5</td>
<td>1.28</td>
<td>0.30</td>
<td>98.25</td>
<td>0.00</td>
<td>0.17</td>
</tr>
<tr>
<td>OutGuess</td>
<td>0.00</td>
<td>0.00</td>
<td>0.06</td>
<td>99.83</td>
<td>0.12</td>
</tr>
<tr>
<td>Steghide</td>
<td>1.54</td>
<td>0.05</td>
<td>0.66</td>
<td>0.26</td>
<td>97.49</td>
</tr>
</tbody>
</table>
J3: Conclusion

- Performance of J3 in terms of capacity is better than Outguess and Steghide.
- J5 has more capacity than F5 when the image size is large but F5 performs poorly with steganalysis.
- When equal message is embedded, J3 has 4% less detection rate than other algorithms.
- 3% lower detection rate compared to other algorithm with 50% message length.
- Embedding efficiency of 0.65 bits per non zero coefficient.
- Overall, J3 is a better candidate than other algorithms in terms of capacity and stealthiness.
Contribution

- J2: a novel technique to embed data in spatial domain by changing coefficients in frequency domain.
- J3: High capacity with complete histogram restoration.
  - Performs better than other existing algorithms in terms of capacity and stealthiness.
- J4: Restoration of second order statistics which has not been done and analyzed before. (Future work- Feb 2011)
- Steganalysis algorithm using second order statistics by estimation of cover image. (Future Work- March 2011)
- Modification of J2 to provide first order compensation and analyzing its performance. (Future Work – May 2011)
Future Work

- Steganography by restoring second order statistics.
  - Most steganalysis methods use second order statistics. These include inter/intra block correlations.
  - J4 aims to restore second order statistics. The embedding process keeps track of all the dependency changes made. A part of the coefficients will be preserved for restoration of these dependencies.
Future Work

- Restoring intra-block dependencies.
  - Keeps track of all the horizontal and vertical transitions. The transitions are stored in bins.
  - One coefficient change will lead to multiple dependency changes.
  - Find a set of coefficient which would restore all those dependencies.
Future Work

- Restoring inter-block statistics
  - Coefficients at the same position in neighboring blocks are correlated.
  - Change to any coefficient would disrupt these correlations.
Future Work

- Steganalysis using cover image estimation
  - Crop the given image by n rows and n columns.
  - Calculate the second order statistics of the cropped image.
  - Calculate the second order statistics using the cropped image.
  - Perform calibration for any bias for the statistics of cropped image.
  - Compare the second order statistics of the given image with the cropped image.
  - If statistics are not close enough, the image is a stego image.
  - Advantage: we do not need any training and testing sets.
Publications


Other Areas


Thank You

- Sincerely thankful to all my committee members:
  - Dr. Richard Newman (Chair)
  - Dr. Jonathan Liu (Co. Chair)
  - Dr. Jose Fortes
  - Dr. Randy Chow
  - Dr. Liuqing Yang
Chi-Square Attack

- Find the difference between the theoretical expected frequency in the steganogram with observed frequency for a pair of values (2,3) (4,5).

Theoretical expected frequency: \( y_i^* = \frac{n_{2i} + n_{2i+1}}{2} \)

Observed frequency: \( y_i = n_{2i} \)

\[ \chi^2 = \sum_{i=1}^{\nu+1} \frac{(y_i - y_i^*)^2}{y_i^*}, \quad p = 1 - \int_0^{\chi^2} \frac{t^{(\nu-2)/2}e^{-t/2}}{2^{\nu/2}\Gamma(\nu/2)} \, dt. \]

When two distributions are equal (p = 1), the image is a stego image embedded with JSteg.
Matrix Encoding

- Form of Hamming error correction coding.
- Advantage: less number of changes to embed more bits
- Disadvantage: low data rate

\[(d_{max}, n, k): \text{a code word with } n \text{ places will be changed in not more than } d_{max} \text{ places to embed } k \text{ bits. For } (1,n,k), n= 2^k-1\]

Hash Function for the code:

\[f(a) = \bigoplus_{i=1}^{n} a_i \cdot i\]

The position of the bit to replace:

\[s = x \oplus f(a)\]

- \(x\) is the bit array to embed.
- \(a_i\) is the LSB of the \(i\)th coefficient.

LSB of 3 coefficients: 1 0 1  X = Bits to embed: 0 1

\[F(a) = \begin{array}{c}0 1 \text{ XOR} \\ 0 0 \text{ XOR} \\ 1 1 \text{ XOR} \end{array} = \begin{array}{c}1 0 \text{ XOR} \\ 1 1 \text{ XOR} \end{array} = 1 0\]

S = 01 XOR 10 = 11 = 3

Changed coefficient bits = 1 0 0
Matrix Encoding

- Increases embedding efficiency, i.e. number of bits embedded per change.

Embedding Efficiency = $W(k) = \frac{R(k)}{D(k)}$

Change Density = $R(k) = \frac{k}{n}$

Embedding Rate = $D(k) = \frac{1}{n + 1} = \frac{1}{2^k}$

<table>
<thead>
<tr>
<th>$k$</th>
<th>$n$</th>
<th>change density</th>
<th>embedding rate</th>
<th>embedding efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>50.00 %</td>
<td>100.00 %</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>25.00 %</td>
<td>66.67 %</td>
<td>2.67</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>12.50 %</td>
<td>42.86 %</td>
<td>3.43</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>6.25 %</td>
<td>26.67 %</td>
<td>4.27</td>
</tr>
<tr>
<td>5</td>
<td>31</td>
<td>3.12 %</td>
<td>16.13 %</td>
<td>5.16</td>
</tr>
<tr>
<td>6</td>
<td>63</td>
<td>1.56 %</td>
<td>9.52 %</td>
<td>6.09</td>
</tr>
<tr>
<td>7</td>
<td>127</td>
<td>0.78 %</td>
<td>5.51 %</td>
<td>7.06</td>
</tr>
<tr>
<td>8</td>
<td>255</td>
<td>0.39 %</td>
<td>3.14 %</td>
<td>8.03</td>
</tr>
<tr>
<td>9</td>
<td>511</td>
<td>0.20 %</td>
<td>1.76 %</td>
<td>9.02</td>
</tr>
</tbody>
</table>
Markov Process Based Steganalysis

- Calculate the horizontal, vertical, diagonal difference matrices.
Markov Process Based Steganalysis

JPEG 2-D Array

Main Diagonal Difference Array

JPEG 2-D Array

Minor Diagonal Difference Array
Markov Process Based Steganalysis

• If a value in the difference matrix is outside the range \([-T, T]\), change it to \(-T\) or \(T\) depending on if it is positive or negative.
• Calculate the transition probability matrices for all four difference matrices.
• In this case, \(T = 4\). Hence we have a TPM of \((2T+1) \times (2T+1) = 81\)
• Total features = \(81 \times 4 = 324\).

\[
M_h(i, j) = \frac{\sum_{u=1}^{S_u-2} \sum_{v=1}^{S_v} \delta(F_h(u, v) = i, F_h(u + 1, v) = j)}{\sum_{u=1}^{S_u-1} \sum_{v=1}^{S_v} \delta(F_h(u, v) = i)}
\]

\[
M_v(i, j) = \frac{\sum_{u=1}^{S_u} \sum_{v=1}^{S_v-2} \delta(F_v(u, v) = i, F_v(u, v + 1) = j)}{\sum_{u=1}^{S_u} \sum_{v=1}^{S_v-1} \delta(F_v(u, v) = i)}
\]

\[
M_d(i, j) = \frac{\sum_{u=1}^{S_u-2} \sum_{v=1}^{S_v-2} \delta(F_d(u, v) = i, F_d(u + 1, v + 1) = j)}{\sum_{u=1}^{S_u-1} \sum_{v=1}^{S_v-1} \delta(F_d(u, v) = i)}
\]

\[
M_m(i, j) = \frac{\sum_{u=1}^{S_u-2} \sum_{v=1}^{S_v-2} \delta(F_m(u + 1, v) = i, F_m(u, v + 1) = j)}{\sum_{u=1}^{S_u-1} \sum_{v=1}^{S_v-1} \delta(F_m(u, v) = i)}
\]
J3- Stop Point Estimation

\[
p_{m,0} = \frac{\sum M_0}{\sum M_0 + \sum M_1}
\]

\[
p_{m,1} = \frac{\sum M_1}{\sum M_0 + \sum M_1}
\]

\[
k_{total} = \frac{\sum_{x=2}^{coeff\_limit} Hist(x)}{}
\]

\[
p_{c,2x+1} = \frac{Hist(2x+1)}{k_{total}}
\]

\[
p_{c,2x} = \frac{Hist(2x)}{k_{total}}
\]

\[
Pr(2x + 1 \rightarrow 2x) = p_{m,0} \cdot p_{c,2x+1}
\]

\[
Pr(2x \rightarrow 2x + 1) = p_{m,1} \cdot p_{c,2x}
\]

\[
Pr(2x + 1 \rightarrow 2x + 1) = p_{m,1} \cdot p_{c,2x+1}
\]

\[
Pr(2x \rightarrow 2x) = p_{m,0} \cdot p_{c,2x}
\]
Let $\gamma_{2x,2x+1}$ = Total number of eligible coefficients visited so far at any instant.

Let $TC_{Ex}(x \rightarrow y)$ be the expected number of coefficients with value $x$ changed to $y$ to embed a data bit.

$$TC_{Ex}(2x + 1 \rightarrow 2x) = \gamma_{2x,2x+1} \cdot Pr(2x + 1 \rightarrow 2x)$$

$$TC_{Ex}(2x + 1 \rightarrow 2x + 1) = \gamma_{2x,2x+1} \cdot Pr(2x + 1 \rightarrow 2x + 1)$$

$$TC_{Ex}(2x \rightarrow 2x + 1) = \gamma_{2x,2x+1} \cdot Pr(2x \rightarrow 2x + 1)$$

$$TC_{Ex}(2x \rightarrow 2x) = \gamma_{2x,2x+1} \cdot Pr(2x \rightarrow 2x)$$

$$TR_{Ex}(2x + 1) = Hist(2x + 1) - \left[ TC_{Ex}(2x + 1 \rightarrow 2x) + TC_{Ex}(2x + 1 \rightarrow 2x + 1) \right]$$

$$TR_{Ex}(2x) = Hist(2x) - \left[ TC_{Ex}(2x \rightarrow 2x + 1) + TC_{Ex}(2x \rightarrow 2x) \right]$$
\[ \Delta_{Ex}(2x + 1) = TC_{Ex}(2x \rightarrow 2x + 1) - TC_{Ex}(2x + 1 \rightarrow 2x), \]

\[ TC_{Ex}(2x \rightarrow 2x + 1) \geq TC_{Ex}(2x + 1 \rightarrow 2x) \]

\[ \Delta_{Ex}(2x) = TC_{Ex}(2x + 1 \rightarrow 2x) - TC_{Ex}(2x \rightarrow 2x + 1), \]

\[ TC_{Ex}(2x + 1 \rightarrow 2x) \geq TC_{Ex}(2x \rightarrow 2x + 1) \]

The stop condition is:

\[ TR_{Ex}(x) = \Delta_{Ex}(x) \]

\[ \gamma_{2x,2x+1} = \frac{Hist(2x + 1)}{p_{m,1} \cdot (p_{c,2x} + p_{c,2x+1})} \]
The estimated embedding capacity, $C_{est}$, for coefficient pair $(2x, 2x+1)$ is:

$$C_{est}(2x, 2x+1) = TC_{Ex}(2x \rightarrow 2x+1) + TC_{Ex}(2x \rightarrow 2x) + TC_{Ex}(2x+1 \rightarrow 2x) + TC_{Ex}(2x+1 \rightarrow 2x+1) \text{ Bits}$$

$$C_{est}(2x, 2x+1) = \gamma_{2x,2x+1} \cdot \left( p_{c,2x} + p_{c,2x+1} \right) \text{ Bits}$$

$C_{est}^{total}$ = Negative Coefficient Capacity + $(-1, 1)$ Capacity + Positive coefficient capacity.

$$C_{est}^{total} = \left( \sum_{x=-1}^{-\text{coeff_limit}} \gamma_{2x,2x-1} \cdot (p_{c,2x} + p_{c,2x-1}) \right)$$

$$+ \left( \gamma_{-1,1} \cdot (p_{c,-1} + p_{c,1}) \right)$$

$$+ \left( \sum_{x=1}^{\text{coeff_limit}} \gamma_{2x,2x+1} \cdot (p_{c,2x} + p_{c,2x+1}) \right) \text{ Bits}$$