A short story about five steganalysts

Jan Kodovský, May 18, IH 2011
Background

- Steganalysts A, B, C, D, E

- Target algorithm:
  - Side-informed embedding, output in JPEG format
  - [http://multimedia.korea.ac.kr/home/?mid=src_stegano](http://multimedia.korea.ac.kr/home/?mid=src_stegano)

- Steganalysis approach:
  - Experimental evaluation of security
  - Feature-based supervised binary classification
  - For simplicity: fixed relative payload 0.10 bpac
  - Results in terms of error rate, median over 10 runs
Results

- Steganalyst A – error rate 44.0%
- Steganalyst B – error rate 42.5%
- Steganalyst C – error rate 39.8%
- Steganalyst D – error rate 29.2%
- Steganalyst E – error rate 2.6% → clearly the best steganalyst
Results

- Steganalyst A – error rate 44.0%
- Steganalyst B – error rate 42.5%
- Steganalyst C – error rate 39.8%
- Steganalyst D – error rate 29.2%
- Steganalyst E – error rate 2.6% ← clearly the best steganalyst

But... the story continues.
Common elements

6,000 images
size 512×512
PNG format

Feature extraction:
features

Classification:
training
threshold $\frac{1}{2}(P_{\text{FA}} + P_{\text{MD}})$

training set

ratio 50:50 PRNG seed

testing set

testing

A short story about five steganalysts
Common elements

6,000 images
size $512 \times 512$
PNG format

Feature extraction:

Classification:

A short story about five steganalysts
Common elements

6,000 images
size 512×512
PNG format

Feature extraction:

Classification:

threshold $\frac{1}{2}(P_{FA} + P_{MD})$
Different JPEG compressors!

- Steganalyst A 44.0% → Matlab – fft command (C library fftw)
- Steganalyst B 42.5% → BatchPNGtoJPG (free SW for Windows OS)
- Steganalyst C 39.8% → ImageMagick – convert (free command-line tool)
- Steganalyst D 29.2% → Matlab – imwrite command (C library wjpg8c)
- Steganalyst E 2.6% → XnView – fast option (free image process. tool)
# Detecting JPEG compressors

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matlab (fft)</td>
<td>× 46.9%</td>
<td>43.4%</td>
<td>32.6%</td>
<td>2.8%</td>
<td></td>
</tr>
<tr>
<td>BatchPNGtoJPG</td>
<td></td>
<td>× 45.9%</td>
<td>32.8%</td>
<td>2.9%</td>
<td></td>
</tr>
<tr>
<td>convert</td>
<td></td>
<td></td>
<td>× 32.4%</td>
<td>3.0%</td>
<td></td>
</tr>
<tr>
<td>Matlab (imwrite)</td>
<td></td>
<td></td>
<td></td>
<td>× 2.9%</td>
<td></td>
</tr>
<tr>
<td>XnView</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
</tr>
</tbody>
</table>

- CC-PEV features, FLD classifier, median over 10 runs.
- Matlab: A and D differ in 1.2% nnz DCT coefficients $\approx$ nsF5 at 0.10 bpac. Always holds: $\text{abs(imwrite)} \geq \text{abs(fft)}$
- Note: imwrite seems to use a popular JPEG compressor (C library wjgp8c), it is also used in IrfanView, XnView (slow option), Phil Sallee’s Matlab JPEG Toolbox.
Final thoughts / questions

- What other algorithms may exhibit similar issues? . . . MME

- What we are detecting doesn’t have to be what we think we are detecting.

- Features for blind steganalysis are powerful to detect various statistical discrepancies. What can we use them for?

- Should blind steganalysis be used in practice? Wouldn’t FA rate be too large?

- To what extent does the steganalyst need to have access to the cover source?