APPROACHING OPTIMAL EMBEDDING IN AUDIO STEGANOGRAPHY WITH GAN

Jianhua Yang\textsuperscript{1}, Huilin Zheng\textsuperscript{1}, Xiangui Kang\textsuperscript{1*}, Yun-Qing Shi\textsuperscript{2}

\textsuperscript{1}Guangdong Key Lab of Information Security, Sun Yat-Sen University, Guangzhou, China

\textsuperscript{2}Department of ECE, New Jersey Institute of Technology, Newark, NJ, USA

Acknowledgement: This work was supported by NSFC (Grant nos. U1536204, 61772571).
Outline

1. Introduction(#3)
2. The proposed framework(#6)
3. Experimental results(#15)
4. Conclusion(#23)
INTRODUCTION
Introduction

❑ Steganography
  ❖ a kind of covert communication method which uses human perceptual redundancy to hidden messages into digital media, such as images, audio or video, without raising any suspicion.

❑ Challenges
  ❖ Deep learning based audio steganalysis

❑ Hand-crafted methods
  ❖ Can not adjust the embedding cost automatically according to the deep learning based steganalyzers.
Introduction

- Image steganography based on Generative Adversarial Networks
  - ASDL-GAN, UT-GAN, and JS-GAN.
- GAN-based audio steganography
  - “probability map generation” approach
  - embedding for temporal domain
THE PROPOSED FRAMEWORK
Two phases

1. Training phase: training the framework to obtain generator for “probability map generation”
Two phases

2. Steganography: using the generator for practical applications of steganography with STC

![Diagram of steganography process]

- **Cover audio**
- **Probability map**
- **Message**
- **STC**
- **Stego audio**
The U-Net based generator is used to generate an embedding probability for each sample of the cover audio.

Four types of blocks
Embedding Simulator

- Embedding simulator is used to translate the probability map into modification map in training phase.

- In conventional steganography methods, the optimal embedding simulator which can be used to convert the probability to modification, is a three-stage staircase function and cannot backpropagate gradients through neural network.

\[
m_i = \begin{cases} 
-1, & \text{if } r_i < \frac{p_i}{2} \\
1, & \text{if } r_i > 1 - \frac{p_i}{2} \\
0, & \text{otherwise}
\end{cases}
\]
Embedding Simulator

- **Double-tanh function**

\[
m_i = -0.5 \times \tanh(\lambda(p_i - 2 \times r_i)) + 0.5 \times \tanh(\lambda(p_i - 2 \times (1 - r_i)))
\]

- \(\{m_i\} \in [-1,1]^{1 \times n}\): modification map
- \(\{p_i\} \in [0,0.5]^{1 \times n}\): probability map
- \(\{r_i\} \in [0,1]^{1 \times n}\): random numbers obeying uniform distribution ranging from 0 to 1
- \(\lambda = 60\)
The discriminator is composed of
- a high-pass filtering (HPF) layer
- five convolutional blocks
- average pooling layers (one global average pooling)
- a fully-connected layer
- a softmax layer
Loss Function

- The discriminator loss function: 
  \[ l_D = - \sum_{i=1}^{2} y_i \log(y_i') \]

- The generator loss function
  \[ l_G = \alpha \times l_G^1 + \beta \times l_G^2 \]
  \[ l_G^1 = -l_D \]
  \[ l_G^2 = (\text{capacity} - n \times \text{payload}), \]

Where

\[ \text{capacity} = \sum_{i=1}^{n} \left( -p_{i+1} \log_2 p_{i+1} - p_{i-1} \log_2 p_{i-1} - p_i \log_2 p_i \right) \]

\[ p_{i+1} = p_{i-1} = \frac{p_i}{2}, \quad p_{i+1} + p_{i-1} + p_i = 1 \]
Embedding

- Cost calculation
  \[ \rho_i = \ln\left(\frac{2}{p_i} - 2\right) \]

- Embedding message
  \[ \text{stego} = \text{STC}(\text{cover}, \text{Message}, \rho) \]
EXPERIMENTAL RESULTS
Datasets and Settings

- **Dataset**
  - **UME-ERJ**: sampling rate is 16 kHz, 20,000 speech clips with length of 1 second
  - **WSJ0**: sampling rate is 16 kHz, 4,000 speech clips from original testing set and 30,000 from original training set with length of 1 second

- **Usage**
  - **UME** – used to train the proposed framework.
  - **WSJ** – steganography dataset, used to evaluate the security of different steganography
Datasets and Settings

- **Hyperparameters**
  - Learning rate: 0.001 for 0.4bps(bit per sample), 0.0001 for other embedding rate
    - Finetune: 0.4bps → 0.3bps & 0.5bps, 0.3bps → 0.2bps, etc.
  - Batch size: 64
  - Training iterations: 7,000
  - Adam optimizer
  - Weights of the generator loss function: $\alpha = 1$, $\beta = 10^{-7}$

- **Steganalysis method**: ChenNet[1], a CNN based audio steganalysis

Datasets and Settings

❑ Selection of $\beta$

- Fixed $\alpha = 1$, then selected $\beta$ from $\{10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}\}$
- When $\beta$ was less than $10^{-7}$, the capacity calculated by

\[
capacity = \sum_{i=1}^{n} (-p_i^{+1} \log_2 p_i^{+1} - p_i^{-1} \log_2 p_i^{-1} - p_i^0 \log_2 p_i^0)
\]

cannot be well fitted to the desired embedding capacity

- The security decreased as $\beta$ increased from $10^{-7}$

Table 1. Detection error rate (%) of different value of $\beta$ using CNN based steganalyzer

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>$10^{-7}$</th>
<th>$10^{-6}$</th>
<th>$10^{-5}$</th>
<th>$10^{-4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>detection error</td>
<td>38.24</td>
<td>35.50</td>
<td>32.28</td>
<td>29.24</td>
</tr>
</tbody>
</table>
Adversarial Training

Fig. 1. Simulating results for the proposed framework with different training iterations when embedding rate is 0.4 bps for `00aa010a.wav' in WSJ. (a) is the origin audio, and (b)-(e) are the embedding probability generated by GAN trained after 500, 1,000, 2,000 and 7,000 iterations respectively.

Table 2. Detection error rate with respect to different training iterations(%) when embedding rate is 0.4 bps.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>500</th>
<th>1000</th>
<th>2000</th>
<th>3000</th>
<th>4000</th>
<th>5000</th>
<th>6000</th>
<th>7000</th>
</tr>
</thead>
<tbody>
<tr>
<td>detection error</td>
<td>28.16</td>
<td>33.35</td>
<td>37.69</td>
<td>36.57</td>
<td>38.56</td>
<td>37.88</td>
<td>37.50</td>
<td>38.82</td>
</tr>
</tbody>
</table>
Comparison with Existing Methods

❑ Additional experiment

<table>
<thead>
<tr>
<th>Dataset for GAN training</th>
<th>Dataset for embedding</th>
</tr>
</thead>
<tbody>
<tr>
<td>UME</td>
<td>WSJ</td>
</tr>
<tr>
<td>WSJ</td>
<td>UME</td>
</tr>
</tbody>
</table>

❑ Comparison methods

❖ LSB Matching [1]
❖ AAC based audio steganography [2]


Table 3. Detection error rate (%) of different steganography.

<table>
<thead>
<tr>
<th>Training dataset for proposed framework</th>
<th>steganography</th>
<th>Embedding rate (bps)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td>UME</td>
<td>LSB Matching</td>
<td>37.76</td>
</tr>
<tr>
<td></td>
<td>AAC based</td>
<td>47.68</td>
</tr>
<tr>
<td></td>
<td>The proposed</td>
<td><strong>48.34</strong></td>
</tr>
<tr>
<td>WSJ</td>
<td>LSB Matching</td>
<td>24.45</td>
</tr>
<tr>
<td></td>
<td>AAC based</td>
<td>37.89</td>
</tr>
<tr>
<td></td>
<td>The proposed</td>
<td><strong>40.93</strong></td>
</tr>
</tbody>
</table>
Comparison with Existing Methods

(a) training GAN with UME

(b) training GAN with WSJ
CONCLUSION
Conclusion

- In this work, we have proposed a framework to learn the embedding probability automatically for audio steganography.
- The experimental results showed that the proposed framework can learn the adaptive embedding probability automatically and obtain better security than hand-crafted audio steganography LSB matching and AAC based method.
- In future research, we will investigate automatic cost learning for audio steganography in the frequency domain and coded domain.
Thank you!