METEOR

Steganography for Realistic Distributions

Gabriel Kaptchuk (Boston University)
Tushar Jois, Matthew Green, Aviel Rubin (Johns Hopkins University)
Widespread Success of Encrypted Systems

- Encrypted Messengers: ~2 Billion Monthly Users
- Encrypted Browsing: Ubiquitous Adoption and Significant Usability Progress
- Censorship Resistance: >2 Million Daily Connections
Perfect Tool: Universal Steganography!
Problem: Univ. Stegano. For Realistic Distributions Has Never Been Deployed.
Our Contributions

Identify and overcome main barriers to realistic steganography

Analyze prior public key steganography protocols

Propose new symmetric key construction with better performance
Related Work

Classical Steganography and FTE

- Seminal work of Simmons [Sim83]
- MANY follow ups [AP98, ZFK+98, Mit99, Cac00, Hlv02, RR03, Le03, LK03, vH04, BC05]
Related Work

Classical Steganography and FTE

- Seminal work of Simmons [Sim83]
- MANY follow ups [AP98, ZFK+98, Mit99, Cac00, HLv02, RR03, Le03, LK03, vH04, BC05]
- Keyless Steganography [ACI+20]
Related Work

Classical Steganography and FTE

- Seminal work of Simmons [Sim83]
- MANY follow ups [AP98, ZFK+98, Mit99, Cac00, HLv02, RR03, Le03, LK03, vH04, BC05]
- Keyless Steganography [ACI+20]
- Lysyanskaya and Meyerovich look at limits of using Markov Models [LM06]
Related Work

Classical Steganography and FTE

- Seminal work of Simmons [Sim83]
- MANY follow ups [AP98, ZFK+98, Mit99, Cac00, HLv02, RR03, Le03, LK03, vH04, BC05]
- Keyless Steganography [ACI+20]
- Lysyanskaya and Meyerovich look at limits of using Markov Models [LM06]
- Format Transforming Encryption [LDJ+14, DCRS13b, DCS15, OYZ+20]
Related Work

Classical Steganography and FTE

- Seminal work of Simmons [Sim83]
- MANY follow ups [AP98, ZFK+98, Mit99, Cac00, HLv02, RR03, Le03, LK03, vH04, BC05]
- Keyless Steganography [ACI+20]
- Lysyanskaya and Meyerovich look at limits of using Markov Models [LM06]
- Format Transforming Encryption [LDJ+14, DCRS13b, DCS15, OYZ+20]

Censorship Avoidance Tools

- obfs4/ScrambleSuit [WPF13]
- Domain Fronting [FLH+15]
- Skypemorph [MLDG12]
- FTEProxy [DCRS13a]
- StegoTorus [WWY+12]
- CensorProofer [WGN+12]
- FreeWave [HRBS13]
Related Work

Classical Steganography and FTE

- Seminal work of Simmons [Sim83]
- MANY follow ups [AP98, ZFK+98, Mit99, Cac00, HLv02, RR03, Le03, LK03, vH04, BC05]
- Keyless Steganography [ACI+20]
- Lysyanskaya and Meyerovich look at limits of using Markov Models [LM06]
- Format Transforming Encryption [LDJ+14, DCRS13b, DCS15, OYZ+20]

Censorship Avoidance Tools

- obfs4/ScrambleSuit [WPF13]
- Domain Fronting [FLH+15]
- Skypemorph [MLDG12]
- FTEProxy [DCRS13a]
- StegoTorus [WWY+12]
- CensorProofer [WGN+12]
- FreeWave [HRBS13]

Ad-Hoc Steganography + Generative Models

- ML Steganography constructions [GGA+05, SSSS07, YHC+09, CC10, CC14, FJA17, VNBB17, YJH+18, Xia18, YGC+19, HH19, DC19, ZDR19]
- Attacking constructions [YHZ19, YWL+19, YWS+18, WBK15, KFH12, MHC+08]
Talk Outline

01 Steganography Refresher

02 Classical Schemes + Generative Models

03 METEOR: Dealing with Low Entropy
Talk Outline

01 Steganography Refresher

02 Classical Schemes + Generative Models

03 METEOR: Dealing with Low Entropy
Universal Steganography Refresher [Hop04]

Encode Message

1. Encrypt message $m$ as $x$ with IND$^*$-CPA scheme
Universal Steganography Refresher [Hop04]

Encode Message

1. Encrypt message $m$ as $x$ with IND$^*$-CPA scheme

2. For each bit $x_i$ of the ciphertext:
Universal Steganography Refresher [Hop04]

Encode Message

1. Encrypt message $m$ as $x$ with IND$\$\$-CPA scheme

2. For each bit $x_i$ of the ciphertext:
   
   a. Sample random $c_i$ from covertext distribution
   
   b. If $h(c_i) = x_i$ (where $h$ is an unbiased hash function):
      
      - Yes: append $c_i$ to the stegotext, and proceed to next $x_i$
      
      - No: return to (a)
Universal Steganography Refresher [Hop04]

**Encode Message**

1. Encrypt message $m$ as $x$ with IND$\$-CPA scheme
2. For each bit $x_i$ of the ciphertext:
   
   a. Sample random $c_i$ from covertext distribution
   b. If $h(c_i) = x_i$ (where $h$ is an unbiased hash function):
      
      -- Yes: append $c_i$ to the stegotext, and proceed to next $x_i$
      
      -- No: return to (a)

**Decode Message**

1. Recover $x_i$ as $h(c_i)$
2. Decrypt $x$ to recover $m$
Universal Steganography Refresher [Hop04]

**Encode Message**

1. Encrypt message $m$ as $x$ with IND$^\$-CPA scheme

2. For each bit $x_i$ of the ciphertext:
   a. Sample random $c_i$ from covertext distribution
   b. If $h(c_i) = x_i$ (where $h$ is an **unbiased** hash function):
      - Yes: append $c_i$ to the stegotext, and proceed to next $x_i$
      - No: return to (a)

**Decode Message**

1. Recover $x_i$ as $h(c_i)$

2. Decrypt $x$ to recover $m$

**Security Intuition**

1. $x_i$ are all random

2. $h$ introduces no bias

3. Therefore, $c_i$ are distributed as the covertext distribution
Barriers To Practical Universal Steganography

1. Lack of Appropriate Samplers
Barriers To Practical Universal Steganography

1. Lack of Appropriate Samplers
   - Covertext distribution too complex
   - Covertext distribution fundamentally unknowable (eg. human text)
   - Best option: good approximation
Barriers To Practical Universal Steganography

1. Lack of Appropriate Samplers
   - Covertext distribution too complex
   - Covertext distribution fundamentally unknowable (eg. human text)
   - Best option: good approximation

2. Unrealistic Entropy Requirements
Barriers To Practical Universal Steganography

1. Lack of Appropriate Samplers
   - Covertext distribution too complex
   - Covertext distribution fundamentally unknowable (eg. human text)
   - Best option: good approximation

2. Unrealistic Entropy Requirements
   - Low entropy means hash function likely must be biased
   - Two potential outcomes:
     1) Sampler never finds “good” sample
     2) Resampling amplifies bias
Barriers To Practical Universal Steganography

1. Lack of Appropriate Samplers

2. Unrealistic Entropy Requirements

Use (Public) Generative Models
Generative Models
Generative Models
Generative Models
Generative Models
Evidence indicates that the asteroid fell in the Yucatan Peninsula, at Chicxulub, Mexico.
Context:
“Evidence indicates that the asteroid fell in the Yucatan Peninsula, at Chicxulub, Mexico.”

Next Word Prediction:
32% - “An”
17% - “The”
12% - “A”
23% - “However”
15% - “Since”
1% - Other Options
Evidence indicates that the asteroid fell in the Yucatan Peninsula, at Chicxulub, Mexico.

Next Word Prediction:
- 32% - “An”
- 17% - “The”
- 12% - “A”
- 23% - “However”
- 15% - “Since”
- 1% - Other Options
“Evidence indicates that the asteroid fell in the Yucatan Peninsula, at Chicxulub, Mexico. The”
Context:
“Evidence indicates that the asteroid fell in the Yucatan Peninsula, at Chicxulub, Mexico. The”

“first importance of Yucatan Peninsula is demonstrated with the following conclusion: the Pliocene Earth has lost about seven times as much vegetation as the Jurassic in regular parts of the globe, from northern India to Siberia…”
Barriers To Practical Universal Steganography

1. Lack of Appropriate Samplers
   - Use (Public) Generative Models

2. Unrealistic Entropy Requirements
   - Naturally Adapt Encoding Rate To Entropy
Talk Outline

01  Steganography Refresher

02  Classical Schemes + Generative Models

03  METEOR: Dealing with Low Entropy
Universal Steganography Barriers

Encode Message

1. Encrypt message $m$ as $x$ with IND$\!$-CPA scheme

2. For each bit $x_i$ of the ciphertext:
   a. Sample random $c_i$ from covertext distribution
   b. If $h(c_i) = x_i$ (where $h$ is an unbiased hash function):
      - Yes: append $c_i$ to the stegotext, and proceed to next $x_i$
      - No: return to (a)
Universal Steganography Barriers

Encode Message

1. Encrypt message $m$ as $x$ with IND-CPA scheme

2. For each bit $x_i$ of the ciphertext:
   a. Sample random $c_i$ using GENERATIVE MODEL
   b. If $h(c_i) = x_i$ (where $h$ is an unbiased hash function):
      - Yes: append $c_i$ to the stegotext, and proceed to next $x_i$
      - No: return to (a)
Universal Steganography Barriers

Encode Message

1. Encrypt message $m$ as $x$ with IND$\$-CPA scheme

2. For each bit $x_i$ of the ciphertext:
   a. Sample random $c_i$ using GENERATIVE MODEL
   b. If $h(c_i) = x_i$ (where $h$ is an unbiased hash function):
      - Yes: append $c_i$ to the stegotext, and proceed to next $x_i$
      - No: return to (a)
1. Encrypt message $m$ as $x$ with IND$\text{-}$CPA scheme

2. For each bit $x_i$ of the ciphertext:
   a. Sample random $c_i$ using GENERATIVE MODEL
   b. If $h(c_i) = x_i$ (where $h$ is a cryptographic hash function):
      - Yes: append $c_i$ to the stegotext, and proceed to next $x_i$
      - No: return to (a)
1. Encrypt message $m$ as $x$ with IND$\$-CPA scheme

2. For each bit $x_i$ of the ciphertext:
   a. Sample random $c_i$ using GENERATIVE MODEL
   b. If $h(c_i) = x_i$ (where $h$ is a cryptographic hash function):
      - Yes: append $c_i$ to the stegotext, and proceed to next $x_i$
      - No: return to (a)

Universal Steganography Barriers

Context + $c_j$ (for $j < i$)

Distribution over $c_i$

Might introduce bias over low entropy distributions of $c_i$
Instantaneous Entropy Over GPT-2

Entropy over time for 4 samples

- Entropy
- Token Position

![Graph showing entropy over token position for 4 samples.](image-url)
Adaptation Options

1. Skip Low Distribution Moments
Adaptation Options

1. Skip Low Distribution Moments
   - Model is public information
   - Entropy is public information
   - Skip all low entropy sampling events (eg. Entropy < 4.5)
Adaptation Options

1. Skip Low Distribution Moments
Adaptation Options

1. Skip Low Distribution Moments

2. Accumulate Entropy
   - Compile channel such that it has sufficient entropy
   - Sample many tokens together
Adaptation Options

1. Skip Low Distribution Moments

2. Accumulate Entropy
Performance When Accumulating Entropy

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Samples (Tokens)</th>
<th>Time (Sec)</th>
<th>Stegotext Len. (KiB)</th>
<th>Overhead (Length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_p = k = 16$</td>
<td>502.8</td>
<td>42.69</td>
<td>2.3</td>
<td>149.4x</td>
</tr>
<tr>
<td>$H_p = k = 32$</td>
<td>880.4</td>
<td>128.41</td>
<td>4.1</td>
<td>261.8x</td>
</tr>
<tr>
<td>$H_p = k = 64$</td>
<td>1645.0</td>
<td>361.28</td>
<td>7.5</td>
<td>482.1x</td>
</tr>
<tr>
<td>$H_p = k = 128$</td>
<td>2994.6</td>
<td>765.40</td>
<td>13.6</td>
<td>870.7x</td>
</tr>
</tbody>
</table>
Talk Outline

01  Steganography Refresher

02  Classical Schemes + Generative Models

03  METEOR: Dealing with Low Entropy
Can We Do Better In The Symmetric Key Setting?
Next Word Prediction:
- 32% - “An”
- 17% - “The”
- 12% - “A”
- 23% - “However”
- 15% - “Since”
- 1% - Other Options
Encoding Intuition

Next Word Prediction:

32% - “An”
17% - “The”
12% - “A”
23% - “However”
15% - “Since”
1% - Other Options

Context

Encrypted Message (as bits)

“The”
Decoding Intuition

Context → Decoded Message

Next Word Prediction:
32% - “An”
17% - “The”
12% - “A”
23% - “However”
15% - “Since”
1% - Other Options

Encrypted Message (as bits)

“The”
Encoding Intuition

An The A However Since

50%
Encoding Intuition

Encrypted Message:
00011...

An The A However Since

50%
Encoding Intuition

Encrypted Message: 00011...

An

The

A

However

Since

Encrypted Message: 11110...

50%
Encoding Intuition

Encrypted Message: 00011...
Encrypted Message: 01101...
Encrypted Message: 11101...

An The A However Since

50%
Decoding Intuition

Encrypted Message begins with 01

[01001... , 01110...]

50%

An The A However Since
Decoding Intuition

No information learned about encrypted message

[01110..., 10011...]

50%

An The A However Since
1. While message not fully encoded:
1. While message not fully encoded:
   a. Sample and apply random mask (from PRG)
Encode Message

1. While message not fully encoded:
   a. Sample and apply random mask (from PRG)
   b. Sample distribution for next $c_1$ from model
1. While message not fully encoded:
   a. Sample and apply random mask (from PRG)
   b. Sample distribution for next $c_i$ from model
   c. Use masked message to determine $c_i$
Encode Message

1. While message not fully encoded:
   a. Sample and apply random mask (from PRG)
   b. Sample distribution for next $c_1$ from model
   c. Use masked message to determine $c_1$
   d. Compute number of bits transferred
1. While message not fully encoded:
   a. Sample and apply random mask (from PRG)
   b. Sample distribution for next $c_1$ from model
   c. Use masked message to determine $c_1$
   d. Compute number of bits transferred
   e. Mark transferred bits as encoded and add $c_1$ to message
1. While message not fully encoded:
   a. Sample and apply random mask (from PRG)
   b. Sample distribution for next $c_1$ from model
   c. Use masked message to determine $c_1$
   d. Compute number of bits transferred
   e. Mark transferred bits as encoded and add $c_1$ to message

1. While message not fully decoded:
   a. Sample distribution for next $c_1$ from model
   b. Compute number of bits transferred by $c_1$
   c. Sample and apply random mask (from PRG)
   d. Mark transferred bits as encoded and add recovered bits to message
<table>
<thead>
<tr>
<th>Mode</th>
<th>Desktop/GPU (sec)</th>
<th>Laptop/CPU (sec)</th>
<th>Stegotext Length (bytes)</th>
<th>Overhead (length)</th>
<th>Capacity (bits/token)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT-2</td>
<td>18.089</td>
<td>82.214</td>
<td>1976</td>
<td>12.36×</td>
<td>3.09</td>
</tr>
<tr>
<td>GPT-2 (Reorder)</td>
<td>30.570</td>
<td>82.638</td>
<td>1391</td>
<td>8.69×</td>
<td>4.11</td>
</tr>
<tr>
<td>GPT-2 (Compress)</td>
<td>11.070</td>
<td>42.942</td>
<td>938</td>
<td>3.39×</td>
<td>3.39</td>
</tr>
<tr>
<td>Wikipedia</td>
<td>19.791</td>
<td>46.583</td>
<td>2002</td>
<td>12.51×</td>
<td>0.64</td>
</tr>
<tr>
<td>Wikipedia (Reorder)</td>
<td>15.515</td>
<td>39.450</td>
<td>1547</td>
<td>9.67×</td>
<td>0.83</td>
</tr>
<tr>
<td>HTTP Headers</td>
<td>49.380</td>
<td>103.280</td>
<td>6144</td>
<td>38.4×</td>
<td>0.21</td>
</tr>
<tr>
<td>HTTP Headers (Reorder)</td>
<td>57.864</td>
<td>127.759</td>
<td>7237</td>
<td>45.23×</td>
<td>0.18</td>
</tr>
<tr>
<td>Device</td>
<td>Load</td>
<td>Encode</td>
<td>Decode</td>
<td>Overhead (time)</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>---------</td>
<td>---------</td>
<td>-----------------</td>
<td></td>
</tr>
<tr>
<td>GPU</td>
<td>5.867</td>
<td>6.899</td>
<td>6.095</td>
<td>1×</td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>5.234</td>
<td>41.221</td>
<td>40.334</td>
<td>4.6×</td>
<td></td>
</tr>
<tr>
<td>Mobile</td>
<td>1.830</td>
<td>473.58</td>
<td>457.57</td>
<td>49.5×</td>
<td></td>
</tr>
</tbody>
</table>
Benefits of Meteor’s Approach

1. Implicit Adjustment

   Encoding rate is asymptotically equal to entropy
Benefits of Meteor’s Approach

1. Implicit Adjustment
   Encoding rate is asymptotically equal to entropy

2. Concretely Efficient Enough to Really Run In Practice
   Implemented and benchmarked run on GPU, CPU, and Mobile
Benefits of Meteor’s Approach

1. Implicit Adjustment
   Encoding rate is asymptotically equal to entropy

2. Concretely Efficient Enough to Really Run In Practice
   Implemented and benchmarked run on GPU, CPU, and Mobile

3. Clear Security Analysis
   Straightforward reduction to security of PRG
Other Parts of Our Work

- Comparison to Prior (Informal) Work
- Ad-hoc Optimizations For Performance
- Easy-to-use Code Demo on Google Co-Lab
Thanks!

ia.cr/2021/686  meteorfrom.space

Gabriel Kaptchuk (Boston University)

Tushar Jois, Matthew Green, Aviel Rubin
(Johns Hopkins University)