Statistical Privacy for Streaming Traffic

Xiaokuan Zhang¹, Jihun Hamm¹, Michael K. Reiter², Yinqian Zhang¹

¹The Ohio State University
²University of North Carolina at Chapel Hill
Traffic Analysis

Server

Encrypted packet sequence

Feature Extraction

Size of packets, Timing of packets, ...

Classification

Sensitive info

Client
Traffic Analysis --- Video Streaming

- Attacks on Encrypted Video Streams based on BURST patterns (Schuster et al. Security’17)

Traffic Analysis --- BURST Patterns

- MPEG-DASH standard: adaptive bitrate streaming technique

Segments (video chunks)

- Segment 1
- Segment 2
- Segment 3
- Segment 4

Buffer below threshold?

Request Next Segment

YES

Server

Client
Traffic Analysis --- BURST Patterns

- MPEG-DASH standard: adaptive bitrate streaming technique

Segments (video chunks)

- Segment 1
- Segment 2
- Segment 3
- Segment 4

Buffer below threshold?

Request Next Segment

YES

BURST in traffic

Server

Client
Traffic Analysis --- BURST Patterns

• Intuition: different videos have different **BURST** patterns
Attack Replication

- Data Collection
  - 40 videos, 100 traces per video (4000 traces)

- Record (timestamp, packet size) of the first 3 mins

- Automated using Selenium + Tshark
Preprocessing
- The raw data (time series) is aggregated into 0.25-second bins
- Each 3-minute video stream → array of 720 elements
Attack Replication

• 5 Classifiers
  • Support Vector Machine (SVM)
  • Logistic Regression (LR)
  • Random Forest (RF)
  • Neural Net
  • Convolutional Neural Net (CNN)

• Classification Result (5-fold cross-validation)

<table>
<thead>
<tr>
<th></th>
<th>SVM</th>
<th>LR</th>
<th>RF</th>
<th>Neural Net</th>
<th>CNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Accuracy</td>
<td>0.809</td>
<td>0.823</td>
<td>0.751</td>
<td>0.831</td>
<td>0.944</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.067</td>
<td>0.063</td>
<td>0.046</td>
<td>0.011</td>
<td>0.004</td>
</tr>
</tbody>
</table>
Traffic Analysis --- Our Work

- Our work: defense using obfuscation
Outline

1. Defense 1: Adversarial Machine Learning
2. Defense 2: Differential Privacy
3. Evaluation
4. Real-world Implementation
5. Discussion
6. Conclusion
Defense 1: Adversarial ML

- Defend against ML adversaries
- Crafting Adversarial Samples
  - Fast Gradient Sign Method (FGSM)

\[ \eta \text{ sign}(\nabla_x L(g(x; \theta), y)) \]
Defense 1: Adversarial ML

- Targets the CNN (eps=0.1): 0.944 -> 0.086

- Limitations of Adversarial Samples

  Attacker may choose a different classifier

  Attacker may conduct adversarial training (0.086 → 0.908)

  Not so effective against others!

  More principled approach?
Outline

1. Defense 1: Adversarial Machine Learning
2. Defense 2: Differential Privacy
3. Evaluation
4. Real-world Implementation
5. Discussion
6. Conclusion
Defense 2: Differential Privacy

- Privacy in database
- Adding noise with a randomized Alg. M

\[ P(M(D) = s) \leq \exp(\epsilon) \times P(M(D') = s) \]
Defence 2: Differential Privacy

- Privacy in database
- Adding noise with a randomized Alg. M

$$P(M(D) = s) \leq \exp(\epsilon) \times P(M(D') = s)$$

Small $\epsilon \rightarrow$ Similar Distribution
Calculating randomized results from data object

Parameterizing the indistinguishability with distance metric $d$

\[ P(M(X) = s) \leq \exp(\epsilon \times d(X, X')) \times P(M(X') = s) \]
Defense 2: Differential Privacy --- d-privacy

Calculating randomized results from data object

Parameterizing the indistinguishability with distance metric $d$

$d(X, X')$

Small $\epsilon \times d(X, X') \rightarrow$ Similar Distribution

$$P(M(X) = s) \leq \exp(\epsilon \times d(X, X')) \times P(M(X') = s)$$
Defense 2: Differential Privacy --- FPA\textsubscript{k} & d*

- Fourier Perturbation Algorithm (FPA\textsubscript{k}): Rastogi et al. (SIGMOD’10)
  \[ FPA\textsubscript{k}(Q, \lambda) \] is $\epsilon$-differentially private for \( \lambda = \sqrt{k} \Delta_2(Q)/\epsilon \), \( \Delta_2(Q) \) denotes the L2 sensitivity of a set of \( Q \)s.

- d*-private Mechanism: Xiao et al. (CCS’15)
  \[
  d^*(x, x') = \sum_{i \geq 1} |(x[i] - x[i - 1]) - (x'[i] - x'[i - 1])|
  \]
  d*-private mechanism is \((d^*, 2\epsilon)\)-private and \((l_1, 4\epsilon)\)-private.

Defense 2: Differential Privacy --- data flow

A: Attacker

D: Defender

\{(t_i, s_i)\} \quad \text{A sequence of original 2-tuples}

D sets window size (w)

x \quad \text{Original time series}

D adds noise

\tilde{x} \quad \text{Noised time series}

D emits packets

\{(\tilde{t}_i, \tilde{s}_i)\} \quad \text{A sequence of noised 2-tuples}

A sets window size (W_A)

\dot{x} \quad \text{Captured time series}

A performs classification

\text{Classification result}
Defense 2: Differential Privacy --- data flow

A: Attacker

D: Defender

\[ w = w_A \]

\[ \{(t_i, s_i)\} \quad \text{A sequence of original 2-tuples} \]

D sets window size (w)

\[ x \quad \text{Original time series} \]

D adds noise

\[ \tilde{x} \quad \text{Noised time series} \]

D emits packets

\[ \{(\tilde{t}_i, \tilde{s}_i)\} \quad \text{A sequence of noised 2-tuples} \]

A sets window size (W_A)

\[ \hat{x} \quad \text{Captured time series} \]

A performs classification

\[ \text{Classification result} \]
Defense 2: Differential Privacy --- data flow

**A: Attacker**

\( \{(t_i, s_i)\} \quad \text{A sequence of original 2-tuples} \)

D sets window size (w)

\( x \quad \text{Original time series} \)

D adds noise

\( \tilde{x} \quad \text{Noised time series} \)

D emits packets

\( \{(\tilde{t}_i, \tilde{s}_i)\} \quad \text{A sequence of noised 2-tuples} \)

**D: Defender**

\( w = w_A \)

\( w \neq w_A \)

A sets window size \( (W_A) \)

\( \dot{x} \quad \text{Captured time series} \)

A performs classification

\( \text{Classification result} \)
Outline

1. Defense 1: Adversarial Machine Learning
2. Defense 2: Differential Privacy
3. Evaluation
4. Real-world Implementation
5. Discussion
6. Conclusion
Evaluation

- 40x100 traces
- Params: $\epsilon = \{5 \times 10^{-8}, 5 \times 10^{-7}, \ldots, 50\}$
  $w = \{0.05s, 0.25s, 0.5s, 1s, 2s\}$
- Clip bound for each window: [0, 1GB]
Evaluation

- 40x100 traces
- Params: \( \epsilon = \{ 5 \times 10^{-8}, 5 \times 10^{-7}, \ldots, 50 \} \)
  \[ w = \{ 0.05s, 0.25s, 0.5s, 1s, 2s \} \]
- Clip bound for each window: [0, 1GB]
Evaluation

- 40x100 traces
- **Params:** \( \epsilon = \{5 \times 10^{-8}, 5 \times 10^{-7}, \ldots, 50\} \)
  \[ w = \{0.05s, 0.25s, 0.5s, 1s, 2s\} \]
- **Clip bound for each window:** [0, 1GB]
Security Evaluation --- $\text{FPA}_k$

$w_A = w$: effect of $w$

\begin{align*}
\epsilon &= 0.05 \\
\epsilon &= 0.5 \\
\epsilon &= 5 \\
\epsilon &= 50
\end{align*}
Security Evaluation --- FPA\textsubscript{k}

\( w \_A = w \): effect of \( \epsilon \)

\( w = 0.05s \)

\( w = 2s \)
Security Evaluation --- FPA_k

\[ w \neq w_A \]

\[ w = 0.05s \]

\[ w = 2s \]

W_A does not matter
Utility Evaluation

- Original cumulative trace $A$, noised cumulative trace $B$
- Waste: $waste = \max_{1 \leq i \leq n} \{ \max(B[i] - A[i], 0) \}$
- Deficit: $deficit = \max_{1 \leq i \leq n} \{ \max(A[i] - B[i], 0) \}$

\[\text{Waste} \quad \text{Deficit} \]
Utility Evaluation --- Waste

$FPA_k$

$d^*$
Utility Evaluation --- Deficit

$$\text{FPA}_k$$

$$d^*$$
FPA_k vs. d*

Baseline Accuracy (2.5%)
Lowest Waste

\[ FPA_k(w = 2s, \epsilon = 0.5) \]
\[ d^*(w = 0.5s, \epsilon = 5e - 6) \]
Outline

1. Defense 1: Adversarial Machine Learning
2. Defense 2: Differential Privacy
3. Evaluation
4. Real-world Implementation
5. Discussion
6. Conclusion
Implementation --- Workflow

- Chrome Extension: change the `range` in the HTTP request ($FPA_k$)

A: Client (Chrome)  
Request

B: Chrome Extension  
Customized Request

C: Youtube Server  
Response

Constant Rate
Implementation --- Effectiveness

• Dataset: 10 videos, 100 traces per video with extension
• 80% training, 20% test

• Settings: $FP\ k_k(w = 1s, \epsilon = 0.5)$  $w_A = \{0.05s, 0.25s, 0.5s, 1s, 2s\}$

• Features:
  • up/down/total bytes per bin (BPB)
  • up/down/total packets per bin (PPB)
  • up/down/total average packet length per bin (LPB)
  • up/down/total bursts (BURST)
  • the combination of all 12 features (ALL)
### Implementation --- Effectiveness

- **Classification result (CNN)**

<table>
<thead>
<tr>
<th>$w_A(s)$</th>
<th>$BPB_{up}$</th>
<th>$BPB_{down}$</th>
<th>$BPB$</th>
<th>$PPB_{up}$</th>
<th>$PPB_{down}$</th>
<th>$PPB$</th>
<th>$LPB_{up}$</th>
<th>$LPB_{down}$</th>
<th>$LPB$</th>
<th>$BURST_{up}$</th>
<th>$BURST_{down}$</th>
<th>$BURST$</th>
<th>$ALL$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>0.16</td>
<td>0.12</td>
<td>0.16</td>
<td>0.12</td>
<td>0.16</td>
<td>0.14</td>
<td>0.14</td>
<td>0.13</td>
<td>0.16</td>
<td>0.14</td>
<td>0.16</td>
<td>0.16</td>
<td>0.15</td>
</tr>
<tr>
<td>0.25</td>
<td>0.20</td>
<td>0.16</td>
<td>0.22</td>
<td>0.18</td>
<td>0.16</td>
<td>0.20</td>
<td>0.12</td>
<td>0.08</td>
<td>0.16</td>
<td>0.23</td>
<td>0.14</td>
<td>0.14</td>
<td>0.19</td>
</tr>
<tr>
<td>0.5</td>
<td>0.19</td>
<td>0.12</td>
<td>0.22</td>
<td>0.14</td>
<td>0.16</td>
<td>0.20</td>
<td>0.14</td>
<td>0.08</td>
<td>0.10</td>
<td>0.19</td>
<td>0.14</td>
<td>0.14</td>
<td>0.15</td>
</tr>
<tr>
<td>1</td>
<td>0.16</td>
<td>0.14</td>
<td>0.18</td>
<td>0.14</td>
<td>0.19</td>
<td>0.13</td>
<td>0.10</td>
<td>0.10</td>
<td>0.11</td>
<td>0.16</td>
<td>0.14</td>
<td>0.14</td>
<td>0.18</td>
</tr>
<tr>
<td>2</td>
<td>0.14</td>
<td>0.12</td>
<td>0.16</td>
<td>0.13</td>
<td>0.14</td>
<td>0.16</td>
<td>0.10</td>
<td>0.10</td>
<td>0.09</td>
<td>0.16</td>
<td>0.16</td>
<td>0.16</td>
<td>0.19</td>
</tr>
</tbody>
</table>
Implementation --- Demo: original
Implementation --- Demo: w. extension
Outline

1. Defense 1: Adversarial Machine Learning
2. Defense 2: Differential Privacy
3. Evaluation
4. Real-world Implementation
5. Discussion
6. Conclusion
Discussion

- Reducing waste:
  - Lowering clip bound (e.g. $[0, 1\text{GB}] \rightarrow [0, 100\text{MB}]$)
  - Increasing $\epsilon$
Discussion

- Leakage through video length
  - Cannot prevent due to utility loss
  - Possible solution: grouping the videos by length and padding them to the longest length in each group
Outline

1. Defense 1: Adversarial Machine Learning
2. Defense 2: Differential Privacy
3. Evaluation
4. Real-world Implementation
5. Discussion
6. Conclusion
Conclusion

• We borrowed techniques from adversarial ML and differential privacy to address privacy concerns of streaming traffic

• We showed that differential privacy effectively defeats inference-based traffic analysis, while remains agnostic to the ML classifiers

• Results suggested that the two differentially private mechanisms offer good security protection with moderate utility loss
Thanks for listening!

Xiaokuan Zhang
zhang.5840@osu.edu
Backup Slides
Security Evaluation --- $\text{FPA}_k$

$w_A = w$: effect of $w$

\begin{align*}
\text{Accuracy} & \quad \text{Accuracy} \\
\epsilon = 0.05 & \quad \epsilon = 0.5
\end{align*}
Security Evaluation --- FPA$_k$

$w_A = w$: effect of $w$

\[ \varepsilon = 0.05 \]

\[ \varepsilon = 0.5 \]

\[ \varepsilon = 5 \]

\[ \varepsilon = 50 \]
Security Evaluation --- $FPA_k$

$w_A = w$: effect of $\epsilon$

$w = 0.05s$

$w = 2s$
Security Evaluation --- FPA$_k$

\[ w \neq w_A \]

\[ w = 0.05s \quad \text{and} \quad w = 2s \]
Security Evaluation

\[ w_A = w: \text{effect of } w \quad d^* \]

- \( \epsilon = 5 \times 10^{-8} \)
- \( \epsilon = 5 \times 10^{-7} \)
- \( \epsilon = 5 \times 10^{-6} \)
- \( \epsilon = 5 \times 10^{-5} \)
Security Evaluation

\[ w_A = w: \text{effect of } \epsilon \]

\[ w = 0.05s \]

\[ w = 2s \]
Security Evaluation

d*: w = 0.05s

d*: w = 2s
Security Evaluation --- Train w. clean, test w. noised

(a) $FPA_k$

(b) $d^*$
Baseline Approach

- Window size: $w$ seconds
- Max value of all bins of all videos (4000 traces): $C$
- Baseline defense mechanism: $C$ bytes per $w$ seconds (all videos)
**Optimal Attacker**

- The Attacker has the knowledge of distribution of both clean data and noised data (but not the mapping between the two)

- First try to remove noise, then perform classification

---

<table>
<thead>
<tr>
<th>$w(s)$</th>
<th>$\epsilon$</th>
<th>$FPA_k$</th>
<th>$d^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.05</td>
<td>0.5</td>
<td>5</td>
</tr>
<tr>
<td>0.05</td>
<td>0.03</td>
<td>0.03</td>
<td>0.25</td>
</tr>
<tr>
<td>0.25</td>
<td>0.03</td>
<td>0.03</td>
<td>0.30</td>
</tr>
<tr>
<td>0.5</td>
<td>0.03</td>
<td>0.03</td>
<td>0.27</td>
</tr>
<tr>
<td>1</td>
<td>0.03</td>
<td>0.03</td>
<td>0.27</td>
</tr>
<tr>
<td>2</td>
<td>0.03</td>
<td>0.03</td>
<td>0.17</td>
</tr>
</tbody>
</table>

Implementation --- Workflow

- Chrome Extension: change the byte range in the HTTP request

A sends a request with a byte range

B fires requests at a constant rate

B returns requested portion from local storage

C sends responses back, B stores responses locally
Discussion

- Comparing $\text{FPA}_k$ with $d^*$
  - Accuracy $\iff$ Security Guarantee
  - $\text{FPA}_k$ requires the knowledge of the entire time series