Preventing Side-Channel Leaks in Web Traffic: A Formal Approach

Goran Doychev
IMDEA Software Institute

joint work with

Michael Backes and Boris Köpf

20th Network & Distributed System Security Symposium
San Diego, CA
February 25, 2013
Leaks in Web Traffic

...
Leaks in Web Traffic

How to show that a countermeasure is “good”?

▶ previous work: empirically show that a particular attack does not work
Leaks in Web Traffic

How to show that a countermeasure is "good"?

▶ previous work: empirically show that a particular attack does not work
▶ an attack: compare distributions of packet sizes (Liberatore et al.'06)
▶ a countermeasure: Traffic morphing (Wright et al.'09)
▶ other attacks still possible (Lu et al.'10, Dyer et al.'12)
Leaks in Web Traffic

▶ an attack: compare distributions of packet sizes (Liberatore et al.'06)
▶ a countermeasure: Traffic morphing (Wright et al.'09)
▶ other attacks still possible (Lu et al.'10, Dyer et al.'12)

How to show that a countermeasure is "good"?
▶ previous work: empirically show that a particular attack does not work
Leaks in Web Traffic

...↑778, ↓720, ↑621, ↓735, ↑615, ↓746, ↑607, ↓726...

...↑633, ↓720, ↓738, ↑66, ↓1320, ↑66, ↓1291, ↓619...
Leaks in Web Traffic

...↑778, ↓720, ↑621, ↓735, ↑615, ↓746, ↑607, ↓726...

...↑633, ↓720, ↓738, ↑66, ↓1320, ↑66, ↓1291, ↓619...

- an attack: compare distributions of packet sizes (Liberatore et al.’06)
Leaks in Web Traffic

...↑778, ↓720, ↑621, ↓735, ↑615, ↓746, ↑607, ↓726...

...↑633, ↓720, ↓738, ↑66, ↓1320, ↑66, ↓1291, ↓619...

▶ an attack: compare distributions of packet sizes (Liberatore et al.’06)
▶ a countermeasure: Traffic morphing (Wright et al.’09)
Leaks in Web Traffic

...↑778, ↓720, ↑621, ↓735, ↑615, ↓746, ↑607, ↓726...

...↑633, ↓720, ↓738, ↑66, ↓1320, ↑66, ↓1291, ↓619...

▶ an attack: compare distributions of packet sizes (Liberatore et al.’06)
▶ a countermeasure: Traffic morphing (Wright et al.’09)
Leaks in Web Traffic

- an attack: compare distributions of packet sizes (Liberatore et al.’06)
- a countermeasure: Traffic morphing (Wright et al.’09)
- other attacks still possible (Lu et al.’10, Dyer et al.’12)
Leaks in Web Traffic

...↑778, ↓720, ↑621, ↓735, ↑615, ↓746, ↑607, ↓726...

...↑633, ↓720, ↓738, ↑66, ↓1320, ↑66, ↓1291, ↓619...

total size:  57.9KB                           total size:  72.4KB

▶ an attack: compare distributions of packet sizes (Liberatore et al.'06)
▶ a countermeasure: Traffic morphing (Wright et al.'09)
▶ other attacks still possible (Lu et al.'10, Dyer et al.'12)
Leaks in Web Traffic

▶ an attack: compare distributions of packet sizes (Liberatore et al.’06)
▶ a countermeasure: Traffic morphing (Wright et al.’09)
▶ other attacks still possible (Lu et al.’10, Dyer et al.’12)

How to show that a countermeasure is “good”? 
Leaks in Web Traffic

...↑778, ↓720, ↑621, ↓735, ↑615, ↓746, ↑607, ↓726...

...↑633, ↓720, ↓738, ↑66, ↓1320, ↑66, ↓1291, ↓619...

total size: 57.9KB

total size: 72.4KB

- an attack: compare distributions of packet sizes (Liberatore et al.’06)
- a countermeasure: Traffic morphing (Wright et al.’09)
- other attacks still possible (Lu et al.’10, Dyer et al.’12)

How to show that a countermeasure is “good”?

- previous work: empirically show that a particular attack does not work
Our approach

Reason formally about strength of countermeasures
Our approach

Reason *formally* about strength of countermeasures

1. models of web applications, web traffic, users and adversaries
Our approach

Reason *formally* about strength of countermeasures

1. models of web applications, web traffic, users and adversaries
2. derive security guarantees based on model
Our approach

Reason *formally* about strength of countermeasures

1. models of web applications, web traffic, users and adversaries
2. derive security guarantees based on model

→ models provide explicit assumptions
Our approach

Reason *formally* about strength of countermeasures

1. models of web applications, web traffic, users and adversaries
2. derive security guarantees based on model
   → models provide explicit assumptions

Main contributions:
Our approach

Reason *formally* about strength of countermeasures

  1. models of web applications, web traffic, users and adversaries
  2. derive security guarantees based on model

→ models provide explicit assumptions

Main contributions:
  ▶ simple, yet realistic models
Our approach

Reason *formally* about strength of countermeasures

1. models of web applications, web traffic, users and adversaries
2. derive security guarantees based on model

→ models provide explicit assumptions

Main contributions:

- simple, yet realistic models
- efficient algorithms for measuring and reducing information leakage
Our approach

Reason formally about strength of countermeasures

1. models of web applications, web traffic, users and adversaries
2. derive security guarantees based on model

→ models provide explicit assumptions

Main contributions:

▶ simple, yet realistic models
▶ efficient algorithms for measuring and reducing information leakage
▶ demonstrate approach in case studies
Modeling web applications

Static website

1. [Image of static website example]
2. [Image of static website example]
3. [Image of static website example]
4. [Image of static website example]

Auto-suggest input field

1. [Image of auto-suggest input field example]
2. [Image of auto-suggest input field example]
3. [Image of auto-suggest input field example]
4. [Image of auto-suggest input field example]
Modeling web applications

Static website

Auto-suggest input field
Modeling web applications

Static website

Auto-suggest input field

Webpage A ➔ Webpage B ➔ Webpage C ➔ Webpage D
The traffic channel
The traffic channel
The traffic channel
The traffic channel
The traffic channel

P[ ]
The traffic channel
The traffic channel

\[ P \]
The traffic channel
Measuring security in the system

P

Webpage A

Webpage B

Webpage C

Webpage D

security measure: difficulty of guessing \(X\) when \(Y\) is known

expected # guesses: captured by entropy \(H(M)\) (Massey'94)

initial uncertainty \(H(X)\)

remaining uncertainty \(H(X|Y)\)
Measuring security in the system

security measure: difficulty of guessing $X$ when $Y$ is known

expected # guesses: captured by entropy $H$ (Massey'94)

initial uncertainty $H(X)$

remaining uncertainty $H(X|Y)$
Measuring security in the system

security measure: difficulty of guessing $X$ when $Y$ is known
Measuring security in the system

- security measure: difficulty of guessing $X$ when $Y$ is known
- expected # guesses: captured by entropy $H$ (Massey’94)
Measuring security in the system

- security measure: difficulty of guessing $X$ when $Y$ is known
- expected # guesses: captured by entropy $H$ (Massey’94)

Initial uncertainty $H(X)$
Measuring security in the system

- security measure: difficulty of guessing $X$ when $Y$ is known
- expected # guesses: captured by entropy $H$ (Massey’94)

- initial uncertainty $H(X)$
- remaining uncertainty $H(X|Y)$
Traffic modifiers: countermeasures, network protocols
Traffic modifiers: countermeasures, network protocols
Traffic modifiers: countermeasures, network protocols

Basic traffic modifiers:

- **padding**: original data $\rightarrow$ padded data
- **dummy**: original data $\rightarrow$ dummy data
- **split**: original data $\rightarrow$ split data
- **shuffle**: original data $\rightarrow$ shuffled data
Traffic modifiers: countermeasures, network protocols

Basic traffic modifiers:

- **padding**:

- **dummy**:

- **split**:

- **shuffle**:

Example (Packet segmentation)

HTTP message $\xrightarrow{split}$ segment 1 $\xrightarrow{split}$ segment 2 $\xrightarrow{split}$ segment 3 $\xrightarrow{padding}$ header segment 1 $\xrightarrow{padding}$ header segment 2 $\xrightarrow{padding}$ header segment 3
Composition theorem

Composed traffic modifier $f_2 \circ f_1$:

\[ H(X | Y_2 \circ Y_1) \geq H(X | Y_1) \]

▶ proof relies on data processing inequality

Consequence: relative security guarantees for free

▶ countermeasure $f_2 \circ f_1$ at least as strong as $f_1$

▶ security guarantees preserved when message passes protocol stack
Composition theorem

Composed traffic modifier $f_2 \circ f_1$:

\[ H(X | Y_2 \circ Y_1) \geq H(X | Y_1) \]
Composition theorem

Composed traffic modifier $f_2 \circ f_1$:

\[ H(X|Y_2 \circ Y_1) \geq H(X|Y_1) \]

- proof relies on data processing inequality

Consequence: relative security guarantees for free

▶ countermeasure $f_2 \circ f_1$ at least as strong as $f_1$

▶ security guarantees preserved when message passes protocol stack
Composition theorem

Composed traffic modifier $f_2 \circ f_1$:

\[ H(X|Y_2 \circ Y_1) \geq H(X|Y_1) \]

- proof relies on \textit{data processing inequality}

Consequence: relative security guarantees for free
Composition theorem

Composed traffic modifier $f_2 \circ f_1$:

Theorem

\[ H(X|Y_2 \circ Y_1) \geq H(X|Y_1) \]

- proof relies on *data processing inequality*

Consequence: relative security guarantees for free

- countermeasure $f_2 \circ f_1$ at least as strong as $f_1$
Composition theorem

Composed traffic modifier $f_2 \circ f_1$:

Theorem

$$H(X \mid Y_2 \circ Y_1) \geq H(X \mid Y_1)$$

▶ proof relies on *data processing inequality*

Consequence: relative security guarantees for free

▶ countermeasure $f_2 \circ f_1$ at least as strong as $f_1$

▶ security guarantees preserved when message passes protocol stack

<table>
<thead>
<tr>
<th>Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
</tr>
<tr>
<td>TCP</td>
</tr>
<tr>
<td>IP</td>
</tr>
<tr>
<td>Ethernet</td>
</tr>
</tbody>
</table>
How to evaluate real-world websites?

Computing the remaining uncertainty:

\[ H(X | Y) \geq H(X) - H(Y) \]

Direct computation of \( H(X) \) not feasible: have to enumerate all paths.

Our approach:

- Assume \( X = X_1, \ldots, X_\ell \) is a Markov chain
- Assume \( P[X_1] = \) stationary distribution

\[ H(X) = H(X_1) + (\ell - 1) H(X_2 | X_1) \]

Obtaining the stationary distribution: use PageRank algorithm.

PageRank algorithm computes the stationary distribution of \( X \):

Random surfer: follow random link or jump to random page.
How to evaluate real-world websites?

Computing the remaining uncertainty:

- $H(X|Y) \geq H(X) - H(Y)$
How to evaluate real-world websites?

Computing the remaining uncertainty:

- $H(X|Y) \geq H(X) - H(Y)$
- direct computation of $H(X)$ not feasible: have to enumerate of all paths
How to evaluate real-world websites?

Computing the remaining uncertainty:

- \( H(X|Y) \geq H(X) - H(Y) \)
- direct computation of \( H(X) \) not feasible: have to enumerate of all paths

Our approach:
How to evaluate real-world websites?

Computing the remaining uncertainty:

- \( H(X|Y) \geq H(X) - H(Y) \)
- direct computation of \( H(X) \) not feasible: have to enumerate of all paths

Our approach:
- assume \( X = X_1, \ldots, X_\ell \) is a Markov chain
How to evaluate real-world websites?

Computing the remaining uncertainty:
- \( H(X|Y) \geq H(X) - H(Y) \)
- direct computation of \( H(X) \) not feasible: have to enumerate of all paths

Our approach:
- assume \( X = X_1, \ldots, X_\ell \) is a Markov chain
- assume \( P[X_1] = \) stationary distribution
How to evaluate real-world websites?

Computing the remaining uncertainty:

▶ $H(X|Y) \geq H(X) - H(Y)$
▶ direct computation of $H(X)$ not feasible: have to enumerate of all paths

Our approach:

▶ assume $X = X_1, \ldots, X_\ell$ is a Markov chain
▶ assume $P[X_1] = \text{stationary distribution}$

$\Rightarrow H(X) = H(X_1) + (\ell - 1)H(X_2|X_1)$
How to evaluate real-world websites?

Computing the remaining uncertainty:
- \( H(X|Y) \geq H(X) - H(Y) \)
- direct computation of \( H(X) \) not feasible: have to enumerate of all paths

Our approach:
- assume \( X = X_1, \ldots, X_\ell \) is a Markov chain
- assume \( P[X_1] = \) stationary distribution
  \[ H(X) = H(X_1) + (\ell - 1)H(X_2|X_1) \]

Obtaining the stationary distribution: use PageRank algorithm
How to evaluate real-world websites?

Computing the remaining uncertainty:

- \( H(X|Y) \geq H(X) - H(Y) \)
- direct computation of \( H(X) \) not feasible: have to enumerate of all paths

Our approach:

- assume \( X = X_1, \ldots, X_\ell \) is a Markov chain
- assume \( P[X_1] = \) stationary distribution

\[ H(X) = H(X_1) + (\ell - 1)H(X_2|X_1) \]

Obtaining the stationary distribution: use PageRank algorithm

- PageRank algorithm computes the stationary distribution of \( X \)
How to evaluate real-world websites?

Computing the remaining uncertainty:

- \( H(X|Y) \geq H(X) - H(Y) \)
- direct computation of \( H(X) \) not feasible: have to enumerate of all paths

Our approach:

- assume \( X = X_1, \ldots, X_\ell \) is a Markov chain
- assume \( P[X_1] = \) stationary distribution

\[ H(X) = H(X_1) + (\ell - 1)H(X_2|X_1) \]

Obtaining the stationary distribution: use PageRank algorithm

- PageRank algorithm computes the stationary distribution of \( X \)
- random surfer: follow random link or jump to random page
Path-aware countermeasures

Countermeasures make vertices indistinguishable
- e.g. order objects by size, pad, add dummy objects
- countermeasure induces partition of vertices

Paths may not be indistinguishable
⇒ ensure partition of vertices is a probabilistic bisimulation
Path-aware countermeasures

Countermeasures make vertices indistinguishable

- e.g. order: order objects by size, pad, add dummy objects

- countermeasure induces partition of vertices

Paths may not be indistinguishable

⇒ ensure partition of vertices is a probabilistic bisimulation

\[
\begin{align*}
A & \rightarrow \frac{2}{3} \rightarrow C \\
B & \rightarrow \frac{1}{3} \rightarrow C \\
E & \rightarrow \frac{1}{3} \rightarrow B \\
& \rightarrow \frac{1}{3} \rightarrow D \\
& \rightarrow \frac{1}{3} \rightarrow D \\
\end{align*}
\]
Path-aware countermeasures

Countermeasures make vertices indistinguishable
Path-aware countermeasures

Countermeasures make vertices indistinguishable

- e.g. c_order: order objects by size, pad, add dummy objects
Path-aware countermeasures

Countermeasures make vertices indistinguishable

- e.g. `c_order`: order objects by size, pad, add dummy objects
- countermeasure induces partition of vertices
Countermeasures make vertices indistinguishable
- e.g. c_order: order objects by size, pad, add dummy objects
- countermeasure induces partition of vertices

Paths may not be indistinguishable
Path-aware countermeasures

Countermeasures make vertices indistinguishable
  - e.g. \( \text{c\_order} \): order objects by size, pad, add dummy objects
  - countermeasure induces partition of vertices

Paths may not be indistinguishable
  \( \Rightarrow \) ensure partition of vertices is a \textit{probabilistic bisimulation}
Path-aware countermeasures (2)

- there are many possible bisimulations
there are many possible bisimulations
Path-aware countermeasures (2)

- there are many possible bisimulations
- our approach: consider random bisimulations
there are many possible bisimulations

our approach: consider random bisimulations
  1. start from random bi-partition
Path-aware countermeasures (2)

- there are many possible bisimulations
- our approach: consider random bisimulations
  1. start from random bi-partition
  2. refine it to coarsest bisimulation /* Derisavi et al.’03 */
Path-aware countermeasures (2)

- there are many possible bisimulations
- our approach: consider random bisimulations
  1. start from random bi-partition
  2. refine it to coarsest bisimulation /* Derisavi et al.’03 */
  3. repeat
Case study

Trading security for overhead : 500 random bisimulations

![Graph showing the relationship between relative overhead and H(X|Y).]
Analyzed website: bar.wikipedia.org (≈ 3,500 pages)

Initial uncertainty:

\[ H(X) \]

\[ \ell \]

# paths

No countermeasure:

\[ H(X|Y) = 0 \]

Applying path-aware countermeasures (path length \( \ell = 5 \)):

- make all webpages have the same fingerprint:
  - expected overhead 73.5 \times \text{original size}
Case study (2)

Analyzed website:

- bar.wikipedia.org (≈ 3,500 pages)
Case study (2)

Analyzed website:

- bar.wikipedia.org (≈ 3,500 pages)

Initial uncertainty:

<table>
<thead>
<tr>
<th>$\ell$</th>
<th>1</th>
<th>3</th>
<th>5</th>
<th>9</th>
<th>15</th>
<th>25</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H(X)$</td>
<td>10.1</td>
<td>21</td>
<td>31.8</td>
<td>53.4</td>
<td>85.9</td>
<td>139.9</td>
<td>221</td>
</tr>
<tr>
<td># paths</td>
<td>3496</td>
<td>$2^{36.5}$</td>
<td>$2^{59.8}$</td>
<td>$2^{106}$</td>
<td>$2^{176}$</td>
<td>$2^{295}$</td>
<td>$2^{472}$</td>
</tr>
</tbody>
</table>

No countermeasure:

$H(X|Y) = 0$

Applying path-aware countermeasures ($\ell = 5$):

- make all webpages have the same fingerprint:
  - expected overhead $73.5 \times$ original size
Case study (2)

Analyzed website:
   ▶ bar.wikipedia.org ($\approx 3,500$ pages)

Initial uncertainty:

<table>
<thead>
<tr>
<th>$\ell$</th>
<th>1</th>
<th>3</th>
<th>5</th>
<th>9</th>
<th>15</th>
<th>25</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H(X)$</td>
<td>10.1</td>
<td>21</td>
<td>31.8</td>
<td>53.4</td>
<td>85.9</td>
<td>139.9</td>
<td>221</td>
</tr>
<tr>
<td># paths</td>
<td>3496</td>
<td>$2^{36.5}$</td>
<td>$2^{59.8}$</td>
<td>$2^{106}$</td>
<td>$2^{176}$</td>
<td>$2^{295}$</td>
<td>$2^{472}$</td>
</tr>
</tbody>
</table>

No countermeasure:
   ▶ $H(X|Y) = 0$
Case study (2)

Analyzed website:
- bar.wikipedia.org (≈ 3,500 pages)

Initial uncertainty:

<table>
<thead>
<tr>
<th>ℓ</th>
<th>1</th>
<th>3</th>
<th>5</th>
<th>9</th>
<th>15</th>
<th>25</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>H(X)</td>
<td>10.1</td>
<td>21</td>
<td>31.8</td>
<td>53.4</td>
<td>85.9</td>
<td>139.9</td>
<td>221</td>
</tr>
<tr>
<td># paths</td>
<td>3496</td>
<td>$2^{36.5}$</td>
<td>$2^{59.8}$</td>
<td>$2^{106}$</td>
<td>$2^{176}$</td>
<td>$2^{295}$</td>
<td>$2^{472}$</td>
</tr>
</tbody>
</table>

No countermeasure:
- $H(X|Y) = 0$

Applying path-aware countermeasures (path length $ℓ = 5$):
Case study (2)

Analyzed website:
- bar.wikipedia.org ($\approx$ 3,500 pages)

Initial uncertainty:

<table>
<thead>
<tr>
<th>$\ell$</th>
<th>1</th>
<th>3</th>
<th>5</th>
<th>9</th>
<th>15</th>
<th>25</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H(X)$</td>
<td>10.1</td>
<td>21</td>
<td>31.8</td>
<td>53.4</td>
<td>85.9</td>
<td>139.9</td>
<td>221</td>
</tr>
<tr>
<td># paths</td>
<td>3496</td>
<td>$2^{36.5}$</td>
<td>$2^{59.8}$</td>
<td>$2^{106}$</td>
<td>$2^{176}$</td>
<td>$2^{295}$</td>
<td>$2^{472}$</td>
</tr>
</tbody>
</table>

No countermeasure:
- $H(X|Y) = 0$

Applying path-aware countermeasures (path length $\ell = 5$):
- make all webpages have the same fingerprint:
  - expected overhead $73.5 \times$ original size
Trading security for overhead: 500 random bisimiulations
Bonus material in the paper

- limits on overhead of path-aware countermeasure
- case study: auto-complete field
- using other entropy measures
- timing leaks: combining security guarantees with predictive timing mitigation (Askarov et al.’10)
Summary
Summary

- formal framework for reasoning about security of web applications
Summary

- formal framework for reasoning about security of web applications
- models of web traffic, user and adversary
Summary

- formal framework for reasoning about security of web applications
- models of web traffic, user and adversary
- algorithm for practical evaluation of websites using PageRank
Summary

- formal framework for reasoning about security of web applications
- models of web traffic, user and adversary
- algorithm for practical evaluation of websites using PageRank
- path-aware countermeasures based on probabilistic bisimulations
Summary

- formal framework for reasoning about security of web applications
- models of web traffic, user and adversary
- algorithm for practical evaluation of websites using PageRank
- path-aware countermeasures based on probabilistic bisimulations
- demonstrate approach in case studies
Summary

- formal framework for reasoning about security of web applications
- models of web traffic, user and adversary
- algorithm for practical evaluation of websites using PageRank
- path-aware countermeasures based on probabilistic bisimulations
- demonstrate approach in case studies