

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



The screenshot shows a web browser window with the address bar displaying "https://en.wikipedia.org/wiki/Puppy". The page title is "Puppy - Wikipedia, the free encyclopedia". The main content area features the article title "Puppy" and a sub-header "From Wikipedia, the free encyclopedia". Below this, there is a short introductory paragraph, a "Development" section, and a photograph of a golden retriever puppy. The left sidebar contains the Wikipedia logo and various navigation links. The browser's address bar and search bar are visible at the top.

W Puppy - Wikipedia, the free encyclopedia - Google Chrome
https://en.wikipedia.org/wiki/Puppy
Google Mail News Sport Reference Work HCC4 TAC Vista Screenshot radio Other Bookmarks
Create account Log in

WIKIPEDIA
The Free Encyclopedia

Article Talk
Read View source View history Search

Puppy

From Wikipedia, the free encyclopedia

This article is about the domestic dog. For other uses, see *Puppy* (disambiguation).

A **puppy** is a juvenile dog. Some puppies may weigh 1–3 lb (0.45–1.4 kg), while larger ones can weigh up to 15–25 lb (6.8–11.3 kg). All healthy puppies grow quickly after birth. A puppy's coat color may change as the puppy grows older, as is commonly seen in breeds such as the Yorkshire Terrier. In vernacular English, puppy refers specifically to dogs while pup may often be used for other mammals such as seals, graffiti, guinea pigs, or even cats.

Contents [view]

- Development
- 1.1 Drinking and suckling
- See also
- References
- External links

Development

Born after an average of 63 days of gestation, puppies emerge in an amnion that is taken off and eaten by the mother dog.^[c] Puppies begin to nurse shortly afterwards, if the dog exceeds its suction, particularly if



Three-month-old Golden Retriever puppy

Development

Leaks in Web Traffic

The screenshot shows the Wikipedia article for "Puppy". The page title is "Puppy - wikipedia, the free encyclopedia - Google Chrome". The URL is "https://en.wikipedia.org/wiki/Puppy". The article text states: "This article is about the domestic dog. For other uses, see **Puppy** (disambiguation). A **puppy** is a juvenile dog. Some puppies may weigh 1–3 lb (0.45–1.4 kg), while larger ones can weigh up to 15–25 lb (6.8–11.3 kg). All healthy puppies grow quickly after birth. A puppy's coat color may change as the puppy grows older, as is commonly seen in breeds such as the Yorkshire Terrier. In vernacular English, puppy refers specifically to dogs while pup may often be used for other mammals such as seals, giraffes, guinea pigs, or even rats." There is a photograph of a golden retriever puppy. The page also includes a "Development" section and a "Contents" table of contents.

The screenshot shows the Wikipedia article for "Bankruptcy in the United States". The page title is "Bankruptcy in the United States - wikipedia, the free encyclopedia - Google Chrome". The URL is "https://en.wikipedia.org/wiki/Bankruptcy_in_the_United_States". The article text states: "Bankruptcy in the United States is governed under the United States Constitution (Article I, Section 8, Clause 4) which authorizes Congress to enact "uniform Laws on the subject of Bankruptcies throughout the United States." Congress has exercised this authority several times since 1787, most recently by adopting the Bankruptcy Reform Act of 1978, as amended, codified in Title 11 of the United States Code and commonly referred to as the "Bankruptcy Code" ("Code"). The Code has been amended several times since, with the most significant recent changes enacted in 2005 through the Bankruptcy Abuse Prevention and Consumer Protection Act of 2005 (BAPCPA). Some law relevant to bankruptcy is found in other parts of the United States Code. For example, bankruptcy courts are found in Title 28 of the United States Code (USC). Tax implications of bankruptcy are found in Title 26 of the United States Code (Internal Revenue Code), and the creation and jurisdiction of bankruptcy courts are found in Title 20 of the United States Code (Judiciary and Judicial procedure). While bankruptcy cases are filed in United States Bankruptcy Court (judge¹) or United States District Courts, and federal law governs procedure in bankruptcy cases, state laws are often applied when determining property rights. For example, law governing the validity of liens or rules protecting certain property from creditors (known as exemptions) may derive from state law or federal law. Because state law plays a major role in many bankruptcy cases, it is often unwise to generalize some bankruptcy issues across state lines." The page also includes a "Contents" table of contents.

Leaks in Web Traffic



The screenshot shows a Google Chrome browser window with the URL <https://en.wikipedia.org/wiki/Puppy>. The page title is "Puppy - Wikipedia, the free encyclopedia". The article text states: "This article is about the domestic dog. For other uses, see [Puppy \(disambiguation\)](#). A puppy is a juvenile dog. Some puppies may weigh 1–3 lb (0.45–1.4 kg), while larger ones can weigh up to 15–22 lb (6.8–10 kg). All healthy puppies grow quickly after birth. A puppy's coat color may change as the puppy grows older, as is commonly seen in breeds such as the Yorkshire Terrier. In vernacular English, puppy refers specifically to dogs while pup may often be used for other mammals such as seals, giraffes, guppies, etc. or even bats."

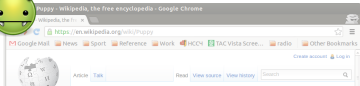
The article includes a "Contents" table with links to "Development", "3.1 Littering and declawing", "2 See also", "3 References", and "4 External links". There is a "Development" section that begins with "Born after an average of 63 days of gestation, puppies emerge in an amnion that is broken off and eaten by the mother dog.^[c] Puppies begin to nurse almost immediately, if the flow exceeds its suction, approximately 7

A photograph of a golden retriever puppy is visible on the right side of the article.

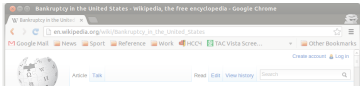
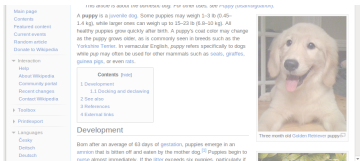
The screenshot shows a Google Chrome browser window with the URL en.wikipedia.org/wiki/Bankruptcy_in_the_United_States. The page title is "Bankruptcy in the United States - Wikipedia, the free encyclopedia". The article text states: "Bankruptcy in the United States is governed under the United States Constitution (Article I, Section 8, Clause 4) which authorizes Congress to enact "uniform Laws on the subject of Bankruptcies throughout the United States." Congress has exercised this authority several times since 1787, most recently by adopting the [Bankruptcy Reform Act of 1978](#), as amended, codified in Title 11 of the United States Code and commonly referred to as the "Bankruptcy Code" ("Code"). The Code has been amended several times since, with the most significant recent changes enacted in 2005 through the [Bankruptcy Abuse Prevention and Consumer Protection Act of 2005](#) (BAPCPA). Some law relevant to bankruptcy is found in other parts of the United States Code. For example, bankruptcy courts are found in Title 28 of the United States Code (USC) (Courts). Tax implications of bankruptcy are found in Title 26 of the United States Code (Internal Revenue Code), and the creation and jurisdiction of bankruptcy courts are found in Title 28 of the United States Code (Judiciary and Judicial procedure). While bankruptcy cases are filed in United States Bankruptcy Court (judge^[a] or United States District Court), and federal law governs procedure in bankruptcy cases, state laws are often applied when determining property rights. For example, law governing the validity of liens or rules protecting certain property from creditors (known as exemptions) may derive from state law or federal law. Because state law plays a major role in many bankruptcy cases, it is often useful to generalize some bankruptcy issues across state lines."

The article includes a "Contents" table with links to "1 History", "2 Chapters of the Bankruptcy Code", and "3 Chapter 11 Liquidation".

Leaks in Web Traffic



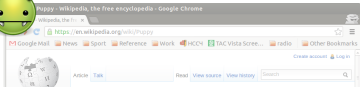
... ↑778, ↓720, ↑621, ↓735, ↑615, ↓746, ↑607, ↓726...



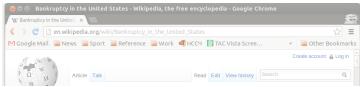
... ↑633, ↓720, ↓738, ↑66, ↓1320, ↑66, ↓1291, ↓619...



Leaks in Web Traffic



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

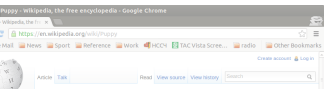


... ↑633, ↓720, ↓738, ↑66, ↓1320, ↑66, ↓1291, ↓619...



- ▶ an attack: compare distributions of packet sizes (Liberatore et al.'06)

Leaks in Web Traffic



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

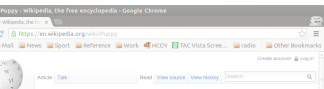


... ↑633, ↓720, ↓738, ↑66, ↓1320, ↑66, ↓1291, ↓619 ...



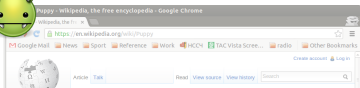
- ▶ an attack: compare distributions of packet sizes (Liberatore et al.'06)
- ▶ a countermeasure: Traffic morphing (Wright et al.'09)

Leaks in Web Traffic

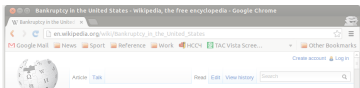


- ▶ an attack: compare distributions of packet sizes (Liberatore et al.'06)
- ▶ a countermeasure: Traffic morphing (Wright et al.'09)

Leaks in Web Traffic



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

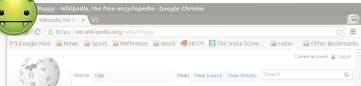


... ↑633, ↓720, ↓738, ↑66, ↓1320, ↑66, ↓1291, ↓619...

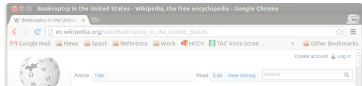


- ▶ an attack: compare distributions of packet sizes (Liberatore et al.'06)
- ▶ a countermeasure: Traffic morphing (Wright et al.'09)
- ▶ other attacks still possible (Lu et al.'10, Dyer et al.'12)

Leaks in Web Traffic



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

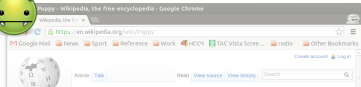


... ↑633, ↓720, ↓738, ↑66, ↓1320, ↑66, ↓1291, ↓619...

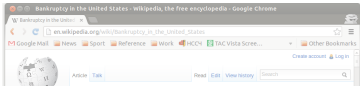
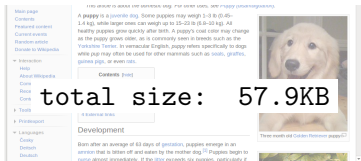


- ▶ an attack: compare distributions of packet sizes (Liberatore et al.'06)
- ▶ a countermeasure: Traffic morphing (Wright et al.'09)
- ▶ other attacks still possible (Lu et al.'10, Dyer et al.'12)

Leaks in Web Traffic



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



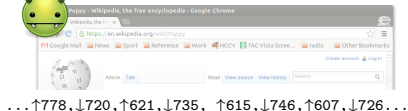
... ↑633, ↓720, ↓738, ↑66, ↓1320, ↑66, ↓1291, ↓619...



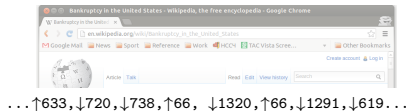
- ▶ an attack: compare distributions of packet sizes (Liberatore et al.'06)
- ▶ a countermeasure: Traffic morphing (Wright et al.'09)
- ▶ other attacks still possible (Lu et al.'10, Dyer et al.'12)

How to show that a countermeasure is “good”?

Leaks in Web Traffic



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

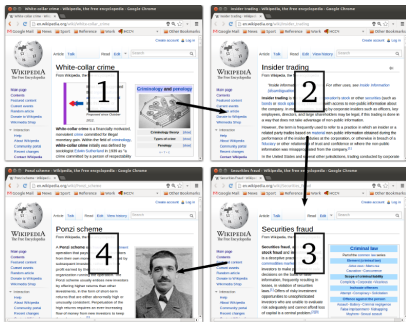
1. models of web applications, web traffic, users and adversaries
 2. derive security guarantees based on model
- ↔ models provide explicit assumptions

Main contributions:

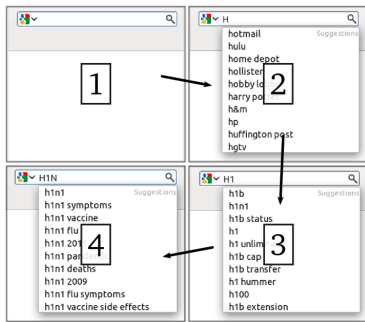
- ▶ simple, yet realistic models
- ▶ efficient algorithms for measuring and reducing information leakage
- ▶ demonstrate approach in case studies

Modeling web applications

Static website

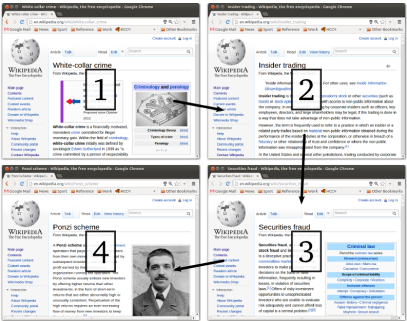


Auto-suggest input field

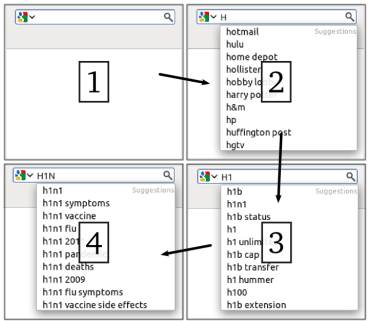


Modeling web applications

Static website

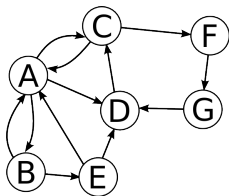


Auto-suggest input field

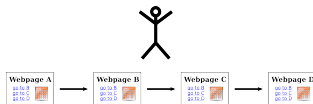
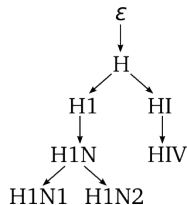


Modeling web applications

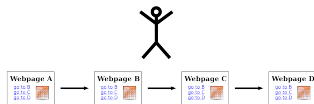
Static website



Auto-suggest input field



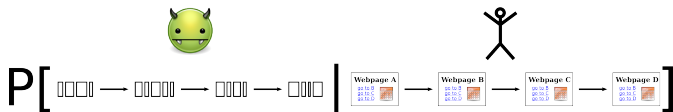
The traffic channel



The traffic channel



The traffic channel



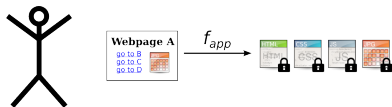
The traffic channel



The traffic channel



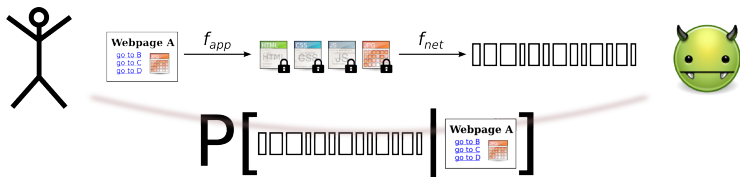
The traffic channel



The traffic channel



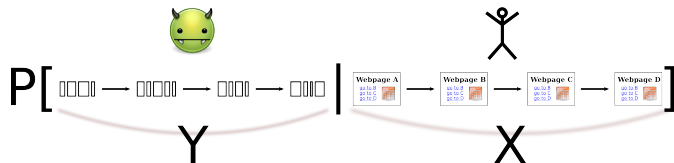
The traffic channel



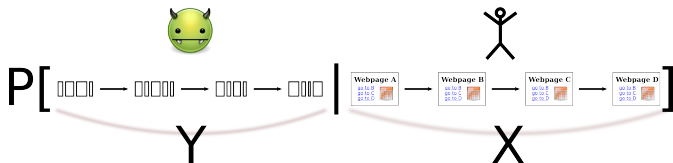
Measuring security in the system



Measuring security in the system

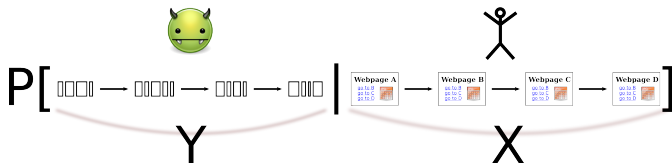


Measuring security in the system



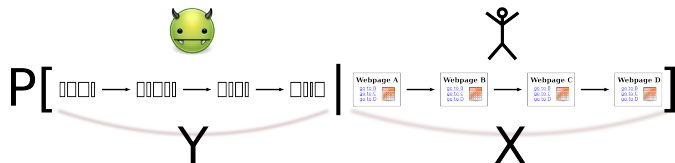
- ▶ security measure: difficulty of guessing X when Y is known

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

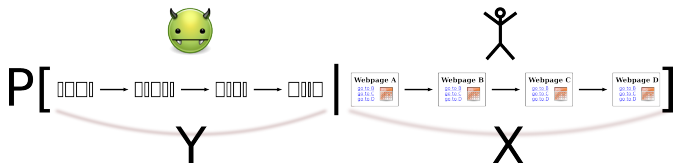


- ▶ security measure: difficulty of guessing X when Y is known
- ▶ expected # guesses: captured by entropy H (Massey'94)

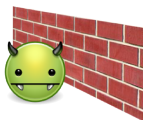


- ▶ initial uncertainty $H(X)$

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 

split 

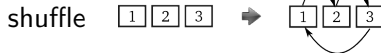
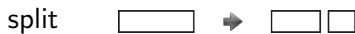
dummy 

shuffle 

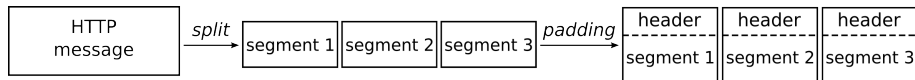
Traffic modifiers: countermeasures, network protocols



Basic traffic modifiers:

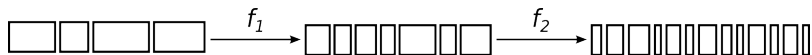


Example (Packet segmentation)



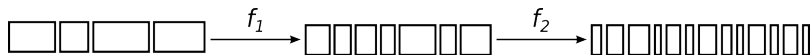
Composition theorem

Composed traffic modifier $f_2 \circ f_1$:



Composition theorem

Composed traffic modifier $f_2 \circ f_1$:

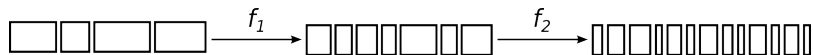


Theorem

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

Composition theorem

Composed traffic modifier $f_2 \circ f_1$:



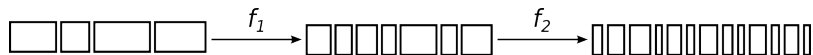
Theorem

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

- ▶ proof relies on *data processing inequality*

Composition theorem

Composed traffic modifier $f_2 \circ f_1$:



Theorem

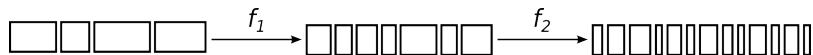
$$H(X|Y_2 \circ Y_1) \geq H(X|Y_1)$$

- ▶ proof relies on *data processing inequality*

Consequence: relative security guarantees for free

Composition theorem

Composed traffic modifier $f_2 \circ f_1$:



Theorem

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

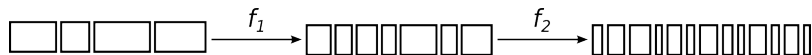
- ▶ proof relies on *data processing inequality*

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

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

- ▶ 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

HTTP
TCP
IP
Ethernet

How to evaluate real-world websites?

How to evaluate real-world websites?

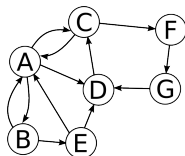
Computing the remaining uncertainty:

- ▶ $H(X|Y) \geq H(X) - H(Y)$

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

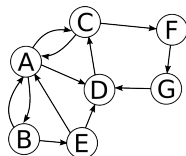


How to evaluate real-world websites?

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:



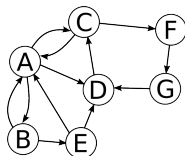
How to evaluate real-world websites?

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 $X = X_1, \dots, X_\ell$ is a Markov chain



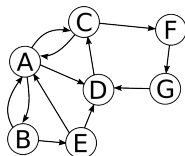
How to evaluate real-world websites?

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 $X = X_1, \dots, X_\ell$ is a Markov chain
- ▶ assume $P[X_1]$ = stationary distribution



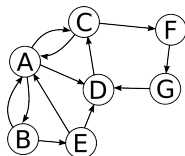
How to evaluate real-world websites?

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:

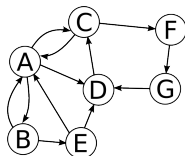
- ▶ assume $X = X_1, \dots, X_\ell$ is a Markov chain
 - ▶ assume $P[X_1]$ = stationary distribution
- $\Rightarrow H(X) = H(X_1) + (\ell - 1)H(X_2|X_1)$



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:

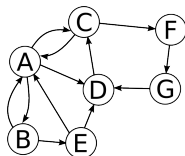
- ▶ assume $X = X_1, \dots, X_\ell$ is a Markov chain
 - ▶ assume $P[X_1]$ = stationary distribution
- ⇒ $H(X) = H(X_1) + (\ell - 1)H(X_2|X_1)$

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 $X = X_1, \dots, X_\ell$ is a Markov chain
 - ▶ assume $P[X_1]$ = stationary distribution
- $\Rightarrow H(X) = H(X_1) + (\ell - 1)H(X_2|X_1)$

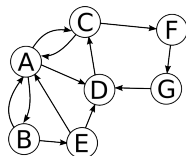
Obtaining the stationary distribution: use PageRank algorithm

- ▶ 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



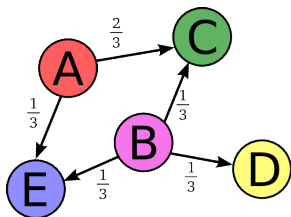
Our approach:

- ▶ assume $X = X_1, \dots, X_\ell$ is a Markov chain
 - ▶ assume $P[X_1]$ = stationary distribution
- $\Rightarrow H(X) = H(X_1) + (\ell - 1)H(X_2|X_1)$

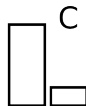
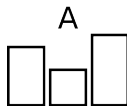
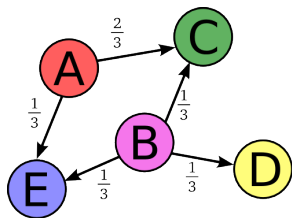
Obtaining the stationary distribution: use PageRank algorithm

- ▶ PageRank algorithm computes the stationary distribution of X
- ▶ random surfer: follow random link or jump to random page

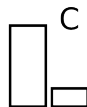
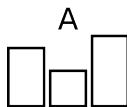
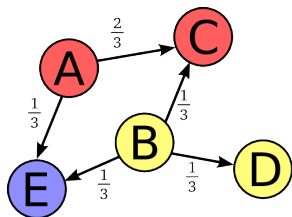
Path-aware countermeasures



Path-aware countermeasures

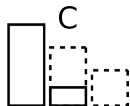
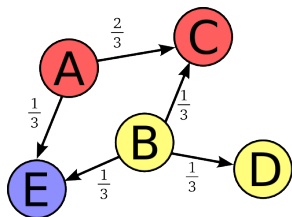


Path-aware countermeasures



Countermeasures make vertices indistinguishable

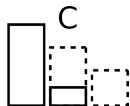
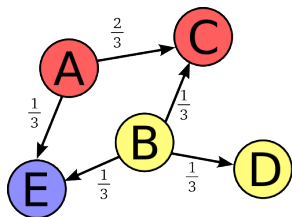
Path-aware countermeasures



Countermeasures make vertices indistinguishable

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

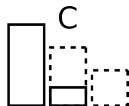
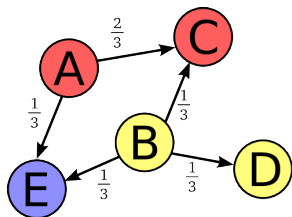
Path-aware countermeasures



Countermeasures make vertices indistinguishable

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

Path-aware countermeasures

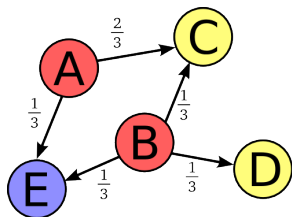


Countermeasures make vertices indistinguishable

- ▶ 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



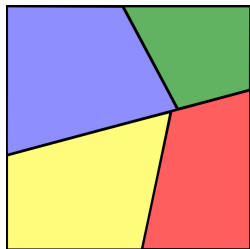
Countermeasures make vertices indistinguishable

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

Paths may not be indistinguishable

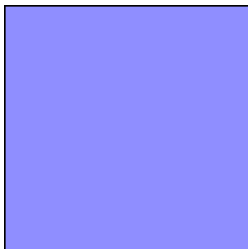
⇒ ensure partition of vertices is a *probabilistic bisimulation*

Path-aware countermeasures (2)



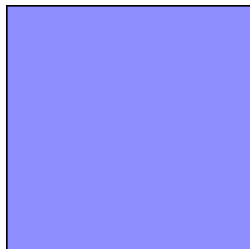
- ▶ there are many possible bisimulations

Path-aware countermeasures (2)



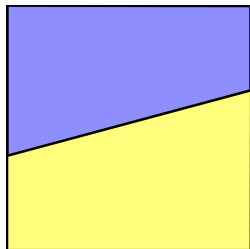
- ▶ there are many possible bisimulations

Path-aware countermeasures (2)



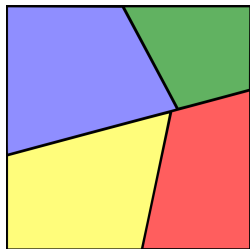
- ▶ there are many possible bisimulations
- ▶ our approach: consider random bisimulations

Path-aware countermeasures (2)



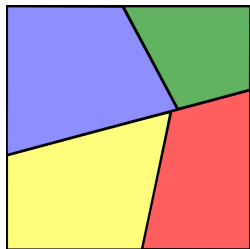
- ▶ there are many possible bisimulations
- ▶ our approach: consider random bisimulations
 1. start from random bi-partition

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 */

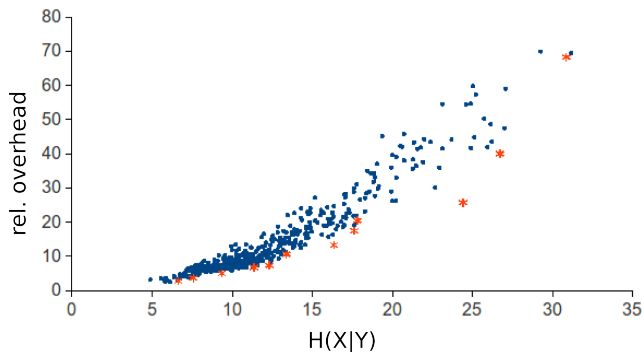
Path-aware countermeasures (2)



- ▶ 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



Case study (2)

Case study (2)

Analyzed website:

- ▶ `bar.wikipedia.org` (\approx 3,500 pages)

Case study (2)

Analyzed website:

- ▶ `bar.wikipedia.org` ($\approx 3,500$ pages)

Initial uncertainty:

ℓ	1	3	5	9	15	25	40
$H(X)$	10.1	21	31.8	53.4	85.9	139.9	221
# paths	3496	$2^{36.5}$	$2^{59.8}$	2^{106}	2^{176}	2^{295}	2^{472}

Case study (2)

Analyzed website:

- ▶ `bar.wikipedia.org` ($\approx 3,500$ pages)

Initial uncertainty:

ℓ	1	3	5	9	15	25	40
$H(X)$	10.1	21	31.8	53.4	85.9	139.9	221
# paths	3496	$2^{36.5}$	$2^{59.8}$	2^{106}	2^{176}	2^{295}	2^{472}

No countermeasure:

- ▶ $H(X|Y) = 0$

Case study (2)

Analyzed website:

- ▶ `bar.wikipedia.org` ($\approx 3,500$ pages)

Initial uncertainty:

ℓ	1	3	5	9	15	25	40
$H(X)$	10.1	21	31.8	53.4	85.9	139.9	221
# paths	3496	$2^{36.5}$	$2^{59.8}$	2^{106}	2^{176}	2^{295}	2^{472}

No countermeasure:

- ▶ $H(X|Y) = 0$

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

Case study (2)

Analyzed website:

- ▶ `bar.wikipedia.org` ($\approx 3,500$ pages)

Initial uncertainty:

ℓ	1	3	5	9	15	25	40
$H(X)$	10.1	21	31.8	53.4	85.9	139.9	221
# paths	3496	$2^{36.5}$	$2^{59.8}$	2^{106}	2^{176}	2^{295}	2^{472}

No countermeasure:

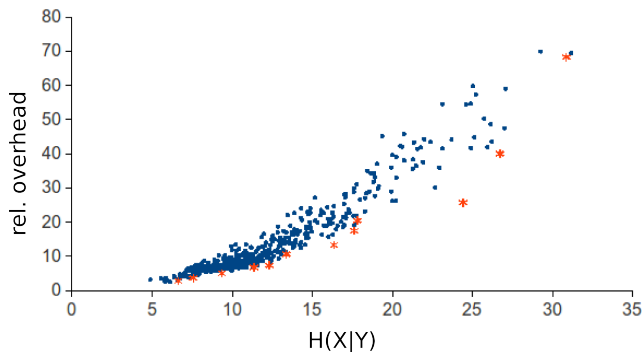
- ▶ $H(X|Y) = 0$

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

- ▶ make all webpages have the same fingerprint:
expected overhead $73.5 \times$ original size

Case study (3)

Trading security for overhead: 500 random bisimulations



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