Measuring Information Leakage using Gain Functions

Kostas Chatzikokolakis

joint work with
Mário Alvim, Catuscia Palamidessi and Geoffrey Smith
CSF 2012

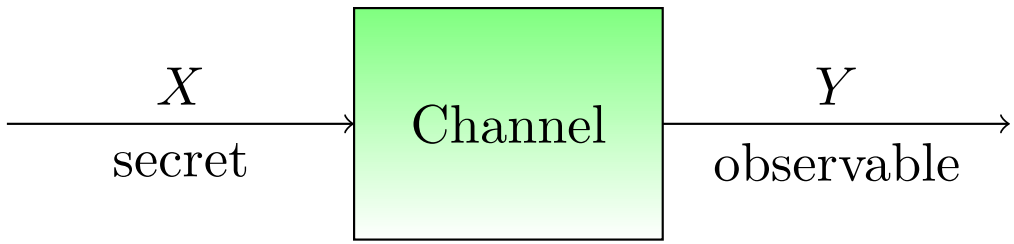
Hiding and Disclosing Information
Dec 3, 2013

Secure Information Flow

- Goal: protecting the confidentiality of secret information
- Problem: information leakage through observable ouputs
- Unfortunately, preventing all leakage is often impractical
  - A login program that rejects an incorrect password leaks some information about the correct password
  - The time taken by cryptographic operations may leak information about secret keys
  - Can we argue that these leaks are "small"?

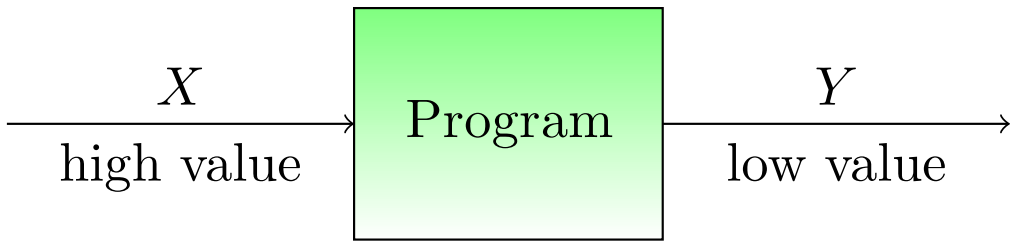
Quantitative Information Flow

Model: information-theoretic channels



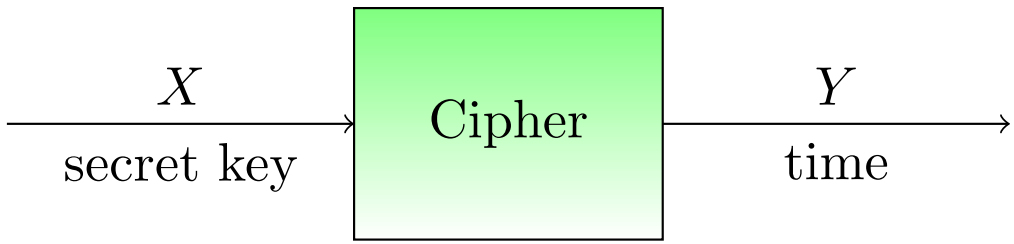
Quantitative Information Flow

Model: information-theoretic channels



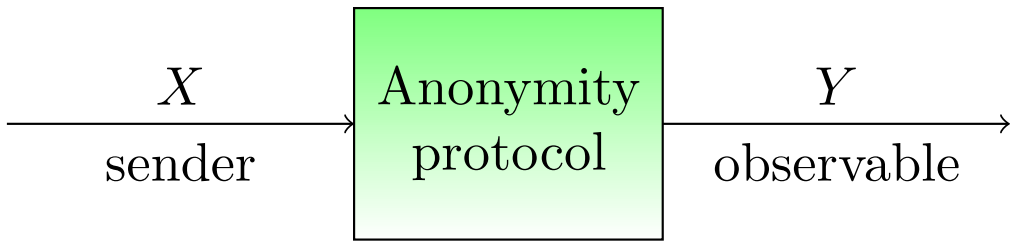
Quantitative Information Flow

Model: information-theoretic channels



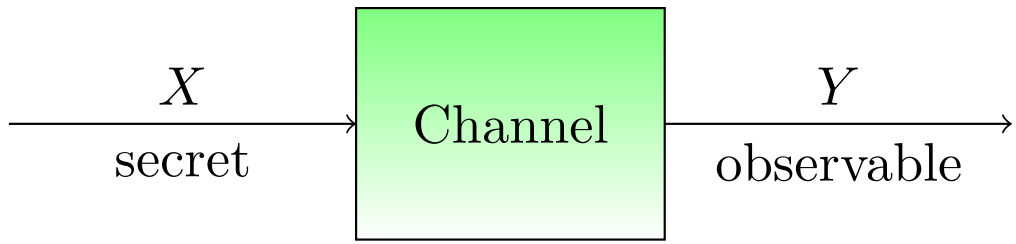
Quantitative Information Flow

Model: information-theoretic channels



Quantitative Information Flow

Model: information-theoretic channels



Given by a channel matrix C
$C[x, y]$: prob. of observing y when the input is x.

X is governed by a prior distribution π.

Both C and π are assumed known to the adversary $\mathcal{A}$.

How can we quantify the leakage from X to Y?

Vulnerability and min-entropy leakage

Operational scenario: $\mathcal{A}$ tries to guess the secret x in one try

Success measure: probability of correct guess

Definition

- Prior vulnerability:
  $V(\pi) = \max_x \pi[x]$
- Posterior vulnerability:
  $V(\pi, C) = \sum_y p(y) V(p_{X|y})$

Min-entropy, min-entropy leakage, and min-capacity

Convert from vulnerability to min-entropy.

H_∞ measures uncertainty in bits.

Definition

- Prior and posterior min-entropy:
  $H_\infty(\pi) = -\log V(\pi)$
  $H_\infty(\pi, C) = -\log V(\pi, C)$

- Min-entropy leakage:
  $\mathcal{L}(\pi, C) = H_\infty(\pi) - H_\infty(\pi, C) = \log \frac{V(\pi, C)}{V(\pi)}$

- Min-capacity:
  $\mathcal{ML}(C) = \sup_\pi \mathcal{L}(\pi, C)$

Limitations of min-entropy leakage

Implicit assumptions:

- $\mathcal{A}$ has to guess the exact secret
- in one try

But what about other scenarios?

- guess part of the secret
- guess the secret approximately
- multiple tries

When min-entropy leakage is not sufficient

pwd_0	pwd_1	...	pwd_{999}

$u \xleftarrow{?} \{0..999\};$
$Y = (u, X[u]);$

Secret: database of 10-bit passwords for 1000 users

Channel C: reveal the password of some randomly chosen user

When min-entropy leakage is not sufficient

pwd_0	pwd_1	...	pwd_{999}

$u \xleftarrow{?} \{0..999\};$
$Y = (u, X[u]);$

Secret: database of 10-bit passwords for 1000 users

Channel C: reveal the password of some randomly chosen user

$\mathcal{A}_1$: guess the complete database

- modelled by the big channel, leaks 10 out of 10000 bits

When min-entropy leakage is not sufficient

pwd_0	pwd_1	...	pwd_{999}

$u \xleftarrow{?} \{0..999\};$
$Y = (u, X[u]);$

Secret: database of 10-bit passwords for 1000 users

Channel C: reveal the password of some randomly chosen user

$\mathcal{A}_1$: guess the complete database

- modelled by the big channel, leaks 10 out of 10000 bits

$\mathcal{A}_2$: guess the password of a particular user i

- sub-channel for user i, leaks 1.016 out of 10 bits

When min-entropy leakage is not sufficient

pwd_0	pwd_1	...	pwd_{999}

$u \xleftarrow{?} \{0..999\};$
$Y = (u, X[u]);$

Secret: database of 10-bit passwords for 1000 users

Channel C: reveal the password of some randomly chosen user

$\mathcal{A}_1$: guess the complete database

- modelled by the big channel, leaks 10 out of 10000 bits

$\mathcal{A}_2$: guess the password of a particular user i

- sub-channel for user i, leaks 1.016 out of 10 bits

$\mathcal{A}_3$: guess the password of any user

- intuitively for such an attacker C leaks everything
- how can we capture this leakage?

Plan of the talk

- Introduction
- Gain functions and g-leakage
- Properties of g-leakage and g-capacity
- Comparing channels
- Conclusion

Gain functions, g-vulnerability

Abstract operational scenario:

$\mathcal{A}$ makes a guess $w \in \mathcal{W}$ about the secret

The benefit provided by guessing w when the secret is x is given by a gain function:

$$g : \mathcal{W} \times \mathcal{X} \to [0, 1]$$

Success measure: the expected gain of a best guess

Definition
Prior g-vulnerability: $V_g(\pi) = \max_w \sum_x \pi[x] g(w, x)$

g-entropy, g-leakage, and g-capacity

Everything else is defined exactly as with min-entropy leakage:

Definition

- Posterior g-vulnerability:
  $V_g(\pi, C) = \sum_y p(y) V_g(p_{X|y})$
- Prior and posterior g-entropy:
  $H_g(\pi) = -\log V_g(\pi)$
  $H_g(\pi, C) = -\log V_g(\pi, C)$
- g-leakage: $\mathcal{L}_g(\pi, C) = H_g(\pi) - H_g(\pi, C) = \log \frac{V_g(\pi, C)}{V_g(\pi)}$
- g-capacity: $\mathcal{ML}_g(C) = \sup_\pi \mathcal{L}_g(\pi, C)$

The power of gain functions

Gain functions can model a great variety of attackers and operational scenarios.

First example: $\mathcal{A}$ guesses the exact secret x in one try

$$\mathcal{W} = \mathcal{X}$$

$$g_{id}(w, x) = \begin{cases} 1, & \text{if } w = x, \\ 0, & \text{if } w \neq x. \end{cases}$$

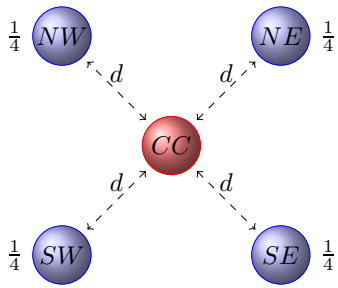
$V_{g_{id}}$ coincides with V

So g-leakage is a generalization of min-entropy leakage.

The power of gain functions

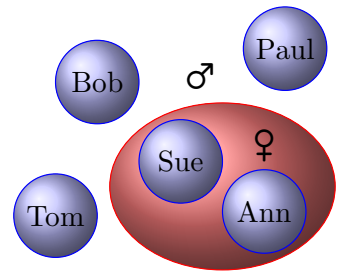
Guessing a secret approximately.

$g(w, x) = 1 - dist(w, x)$



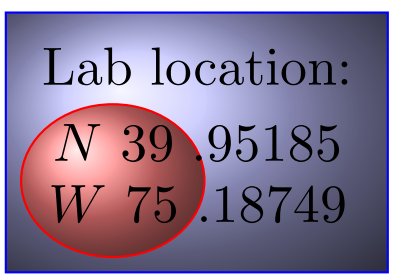
Guessing a property of a secret.

$g(w, x) =$ Is x of gender w?



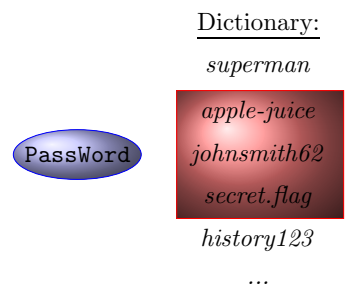
Guessing a part of a secret.

$g(w, x) =$ Does w match the high-order bits of x?



Guessing a secret in 3 tries.

$g_3(w, x) =$ Is x an element of set w of size 3?



A gain function for the password database example

pwd_0	pwd_1	...	pwd_{999}

$u \xleftarrow{?} \{0..999\};$
$Y = (u, X[u]);$

$\mathcal{A}_3$: guess the password of any user

$\mathcal{W} = \{(u, p) \mid u \in \{0 \ldots 999\}, p \in \{0 \ldots 1023\}\}$

$$g((u,p),x) = \begin{cases} 1, & \text{if } x[u] = p \\ 0, & \text{otherwise.} \end{cases}$$

g-leakage: 10 bits out of 10

Two channels with the same min-entropy leakage

Two programs from [Smith 2009].

- C_1: completely reveal X one-eighth of the time
- C_2: always reveal all but the last three bits of X

Quite different threats, is the one better than the other?

- min-entropy leakage: same for both channels
- g_3 : allow 3 guesses
  - C_2 leaks more than C_1
- g_{tiger}: penalize wrong guesses
  - C_1 leaks more than C_2

Plan of the talk

- Introduction
- Gain functions and g-leakage
- Properties of g-leakage and g-capacity
- Comparing channels
- Conclusion

Properties g-capacity and min-capacity

Theorem ("Miracle")

g-capacity $\leq$ *min-capacity* *for all gain functions g*

Properties g-capacity and min-capacity

Theorem ("Miracle")

g-capacity $\leq$ min-capacity for all gain functions g

- Hence a channel with small min-capacity has small g-leakage under every prior and every gain function.

Corollary

A channel's k-tries capacity cannot exceed its 1-try capacity. (although the k-tries vulnerability will be higher)

Theorem

Min-capacity is an upper bound on Shannon capacity.

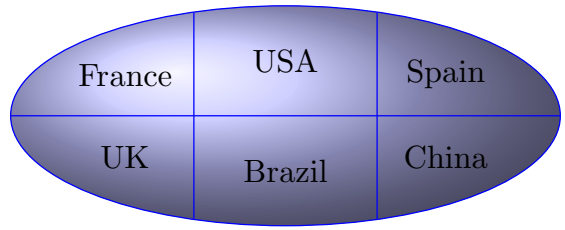
Plan of the talk

- Introduction
- Gain functions and g-leakage
- Properties of g-leakage and g-capacity
- Comparing channels
- Conclusion

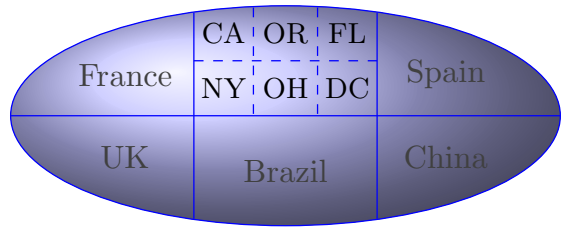
Partition refinement

Any deterministic channel C induces a partition on $\mathcal{X}$.
x_1 and x_2 are in the same block iff they map to the same output.

$C_{country}$
person $\rightarrow$ country of birth



C_{state}
person $\rightarrow$ state of birth



Partition refinement $\sqsubseteq$: Subdivide zero or more of the blocks.

$$C_{country} \sqsubseteq C_{state}$$

Leakage ordering

$C_1 \leq_m C_2 \quad m \in \{\textit{Shannon}, \textit{min-entropy}, \textit{guessing entropy}\}$

the leakage of C_1 is no greater than that of C_2 for all priors

Theorem (Yasuoka, Terauchi, Malacaria)

On deterministic channels, the relations below coincide:

- $\leq_{\textit{Shannon entropy}}$
- $\leq_{\textit{min-entropy}}$
- $\leq_{\textit{guessung entropy}}$
- $\sqsubseteq$

How can we generalize this to probabilistic channels?

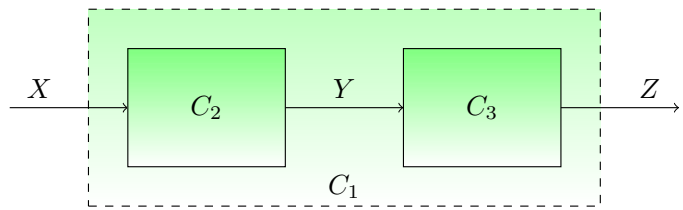
Composition refinement

First issue: $\sqsubseteq$ is not defined for probabilistic chanels

C_{merge}: state $\to$ country
$C_{country} = C_{state} C_{merge}$

Definition (composition-refinement)

$C_1 \sqsubseteq_\circ C_2$ iff $C_1 = C_2 C_3$ for some C_3,



Theorem

For deterministic channels, $\sqsubseteq$ and $\sqsubseteq_\circ$ coincide.

$\sqsubseteq_\circ$ is a promising generalization of $\sqsubseteq$ to probabilistic channels.

Composition refinement and strong leakage ordering

composition refinement $\overset{?}{\Leftrightarrow}$ leakage order

for probabilistic channels?

Definition

$C_1 \leq_{\mathcal{G}} C_2$ iff $\mathcal{L}_g(\pi, C_1) \leq \mathcal{L}_g(\pi, C_2)$ for all π, g

Theorem

$C_1 \sqsubseteq_\circ C_2 \;\Rightarrow\; C_1 \leq_{\mathcal{G}} C_2$

an analogue of the data-processing inequality for g-leakage
("post-processing can only destroy information")

What about the converse?

It turns out that $\leq_{\textit{min-entropy}} \not\Rightarrow \sqsubseteq_\circ$

On the other hand $\leq_{\mathcal{G}}$ is strong enough:

Theorem ("Coriaceous")

$C_1 \leq_{\mathcal{G}} C_2 \Rightarrow C_1 \sqsubseteq_\circ C_2$

The proof turned out to be challenging (at least for us!).

Conjecture

In [CSF'12], we conjectured this result, and proved it in several special cases:

- C_2 is invertible;
- C_2's columns are linearly independent;
- C_1 is deterministic.

In those cases we can even restrict g to be a binary gain function, i.e. one that returns only 0 or 1.

But binary gain functions are not sufficient in general.

Conjecture

In [CSF'12], we conjectured this result, and proved it in several special cases:

- C_2 is invertible;
- C_2's columns are linearly independent;
- C_1 is deterministic.

In those cases we can even restrict g to be a binary gain function, i.e. one that returns only 0 or 1.

But binary gain functions are not sufficient in general.

The full proof came by McIver et al. It is a geometric proof based on the Hyperplane separation lemma.

Conclusion

- g-leakage allows the accurate quantification of leakage in a rich variety of operational scenarios.
- Min-capacity is an upper bound on g-leakage, so we may not need to worry about the multitude of possible gain functions.
- Composition refinement is a promising generalization of the Lattice of Information to probabilistic channels.

Future directions:

- Find algorithms to calculate g-capacity, possibly using linear programming.
- Explore the relation between g-leakage and diff. privacy.