Recent Developments in Quantitative Information Flow

Geoffrey Smith
Florida International University

LICS Tutorial, 9 July 2015
Confidentiality

- Protecting the confidentiality of private information is a fundamental issue in computer security.

- Access control and encryption are valuable tools, but they cannot stop authorized systems from leaking their secret inputs, maliciously or accidentally, to their observable outputs.

- We need to control the flow of information in systems.
Noninterference [Cohen77, GoguenMeseguer82] requires that a system’s observable output be independent of its secret input.

Noninterference can be guaranteed by means of type systems.

Unfortunately, noninterference is too strong, because some leakage of sensitive information is often unavoidable in practice.
Motivating example: Statistical database query

- **Secret input**: database of confidential entries
- **Observable output**: result of statistical query (e.g. percentage of population with some disease), possibly with noise added
Motivating example: Password checker

- Check whether guess is equal to password:

```java
result = true;
for (i=0; i < N; i++) {
    if (password[i] != guess[i]) {
        result = false;
        break;
    }
}
```

- Secret input: password

- Observable output:
  - result
  - running time?
    - (If so, the length of the correct prefix is also leaked!)
Motivating example: Crowds protocol [ReiterRubin98]

- Crowd members wish to communicate anonymously with a server.
- The *initiator* first sends the message to a randomly-chosen forwarder (possibly itself).
- Each forwarder forwards it again with probability $p_f$, or sends it to the server with probability $1-p_f$.
- But some crowd members are collaborators that report who sends them a message.
- **Secret input:** identity of *initiator*
- **Observable output:** first *sender* of a message to a collaborator (or no one)
Motivating example: Timing attack on cryptography
[BonehBrumley03]

1024-bit RSA private key recovered in 2 hours from standard OpenSSL implementation across LAN.

- Secret input: RSA private key
- Observable output: approximate timings of decryptions of a sequence of nonces
Plan of the talk

- Motivation
- Concepts of Quantitative Information Flow
  - Channels, hyper-distributions, vulnerability, min-entropy leakage, $g$-leakage
- Robustness
  - Robust channel ordering: composition refinement
  - Capacity: multiplicative and additive
- Conclusion
Information-theoretic channels

- The earlier examples can all be modeled as **channels**.
- Finite sets $\mathcal{X}$ (secret inputs), $\mathcal{Y}$ (observable outputs).
  - The choice of $\mathcal{Y}$ is subtle, and crucial!
- On input $\mathcal{X}$, the channel probabilistically outputs $\mathcal{Y}$.
- **Channel matrix** $C$ gives the conditional probabilities $p(y|x)$:

<table>
<thead>
<tr>
<th></th>
<th>$\gamma_1$</th>
<th>$\gamma_2$</th>
<th>$\gamma_3$</th>
<th>$\gamma_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>1/4</td>
<td>1/2</td>
<td>1/4</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1/2</td>
<td>1/3</td>
<td>1/6</td>
<td>0</td>
</tr>
</tbody>
</table>

- The rows of $C$ sum to 1.
- $C$ is **deterministic** if each row contains exactly one 1.
We wish to quantify the leakage of secret input $X$ to observable output $Y$ caused by channel $C$, allowing us to argue that some leaks are “small”.

The secrecy of $X$ is modeled by a prior distribution $\pi$ on $X$.

Both $\pi$ and $C$ are assumed known by the adversary $A$.

Key insight: The (information-theoretic) essence of $C$ is a mapping from priors $\pi$ to hyper-distributions $[\pi,C]$, which are distributions on posterior distributions.
Example

<table>
<thead>
<tr>
<th>Prior</th>
<th>Channel matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi$</td>
<td>$C$ $y_1$ $y_2$ $y_3$ $y_4$</td>
</tr>
<tr>
<td>$1/4$</td>
<td>$x_1$ 1/2 1/2 0 0</td>
</tr>
<tr>
<td>$1/2$</td>
<td>$x_2$ 0 1/4 1/2 1/4</td>
</tr>
<tr>
<td>$1/4$</td>
<td>$x_3$ 1/2 1/3 1/6 0</td>
</tr>
</tbody>
</table>
## Example

<table>
<thead>
<tr>
<th>Prior</th>
<th>Channel matrix</th>
<th>Joint matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( C ) ( y_1 ) ( y_2 ) ( y_3 ) ( y_4 )</td>
<td></td>
</tr>
<tr>
<td>( 1/4 )</td>
<td>( x_1 ) 1/2 1/2 0 0</td>
<td>( x_1 ) 1/8 1/8 0 0</td>
</tr>
<tr>
<td>( 1/2 )</td>
<td>( x_2 ) 0 1/4 1/2 1/4</td>
<td>( x_2 ) 0 1/8 1/4 1/8</td>
</tr>
<tr>
<td>( 1/4 )</td>
<td>( x_3 ) 1/2 1/3 1/6 0</td>
<td>( x_3 ) 1/8 1/12 1/24 0</td>
</tr>
</tbody>
</table>

Scale rows with \( \pi \).
### Example

<table>
<thead>
<tr>
<th>Prior</th>
<th>Channel matrix</th>
<th>Joint matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \pi )</td>
<td>( C )</td>
<td>( x_1 )</td>
</tr>
<tr>
<td>1/4</td>
<td>1/2 1/2 0 0</td>
<td>1/8 1/8 0 0</td>
</tr>
<tr>
<td>1/2</td>
<td>0   1/4 1/2 1/4</td>
<td>0   1/8 1/4 1/8</td>
</tr>
<tr>
<td>1/4</td>
<td>1/2 1/3 1/6 0</td>
<td>1/8 1/12 1/24 0</td>
</tr>
</tbody>
</table>

- **Prior**: \( \pi \) = [1/4, 1/2, 1/4]
- **Channel matrix**: 
  - \( x_1 \) = [1/2, 1/2, 0, 0]
  - \( x_2 \) = [0, 1/4, 1/2, 1/4]
  - \( x_3 \) = [1/2, 1/3, 1/6, 0]
- **Joint matrix**: 
  - \( J \) = [1/8, 1/8, 0, 0]
  - \( J \) = [0, 1/8, 1/4, 1/8]
  - \( J \) = [1/8, 1/12, 1/24, 0]

**Scale rows with \( \pi \).**

**Add up columns.**

**Distribution on \( Y \)**: 
- \( p_Y \) = [1/4, 1/3, 7/24, 1/8]
## Example

### Prior
<table>
<thead>
<tr>
<th>$\pi$</th>
<th>$c$</th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
<th>$y_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td></td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1/2</td>
<td></td>
<td>0</td>
<td>1/4</td>
<td>1/2</td>
<td>1/4</td>
</tr>
<tr>
<td>1/4</td>
<td></td>
<td>1/2</td>
<td>1/3</td>
<td>1/6</td>
<td>0</td>
</tr>
</tbody>
</table>

### Channel matrix

### Joint matrix

Scale rows with $\pi$.

### Joint matrix

<table>
<thead>
<tr>
<th>$j$</th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
<th>$y_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>1/8</td>
<td>1/8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>1/8</td>
<td>1/4</td>
<td>1/8</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1/8</td>
<td>1/12</td>
<td>1/24</td>
<td>0</td>
</tr>
</tbody>
</table>

### Distribution on $Y$

| $p_Y$ | 1/4 | 1/3 | 7/24 | 1/8 |

### Posterior distributions

| $p_{X|Y_1}$ | $p_{X|Y_2}$ | $p_{X|Y_3}$ | $p_{X|Y_4}$ |
|-------------|-------------|-------------|-------------|
| $x_1$       | 1/2         | 3/8         | 0           | 0           |
| $x_2$       | 0           | 3/8         | 6/7         | 1           |
| $x_3$       | 1/2         | 1/4         | 1/7         | 0           |

Add up columns.

Normalize columns.
## Example

### Prior
<table>
<thead>
<tr>
<th>$\pi$</th>
<th>$C$</th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
<th>$y_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td></td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1/2</td>
<td></td>
<td>0</td>
<td>1/4</td>
<td>1/2</td>
<td>1/4</td>
</tr>
<tr>
<td>1/4</td>
<td></td>
<td>1/2</td>
<td>1/3</td>
<td>1/6</td>
<td>0</td>
</tr>
</tbody>
</table>

### Channel matrix

### Joint matrix
<table>
<thead>
<tr>
<th>$J$</th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
<th>$y_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>1/8</td>
<td>1/8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>1/8</td>
<td>1/4</td>
<td>1/8</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1/8</td>
<td>1/12</td>
<td>1/24</td>
<td>0</td>
</tr>
</tbody>
</table>

### Distribution on Y
<table>
<thead>
<tr>
<th>$p_Y$</th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
<th>$y_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1/4</td>
<td>1/3</td>
<td>7/24</td>
<td>1/8</td>
</tr>
</tbody>
</table>

### Hyper-distribution on X
<table>
<thead>
<tr>
<th>$[\pi,C]$</th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
<th>$y_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1/2</td>
<td>3/8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1/3</td>
<td>0</td>
<td>3/8</td>
<td>6/7</td>
<td>1</td>
</tr>
<tr>
<td>7/24</td>
<td>1/2</td>
<td>1/4</td>
<td>1/7</td>
<td>0</td>
</tr>
<tr>
<td>1/8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Posterior distributions
| $p_{X|Y}$ | $y_1$ | $y_2$ | $y_3$ | $y_4$ |
|-----------|-------|-------|-------|-------|
| $x_1$     | 1/2   | 3/8   | 0     | 0     |
| $x_2$     | 0     | 3/8   | 6/7   | 1     |
| $x_3$     | 1/2   | 1/4   | 1/7   | 0     |

- **Scale rows with $\pi$.**
- **Add up columns.**
- **Normalize columns.**
- **Drop output labels.**
Abstractly, a channel is a mapping from priors to hyper-distributions [McIverMeinickeMorgan10].

Hyper-distribution on $X$

<table>
<thead>
<tr>
<th>$[\pi,C]$</th>
<th>1/4</th>
<th>1/3</th>
<th>7/24</th>
<th>1/8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>1/2</td>
<td>3/8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>3/8</td>
<td>6/7</td>
<td>1</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1/2</td>
<td>1/4</td>
<td>1/7</td>
<td>0</td>
</tr>
</tbody>
</table>

Channel matrix

<table>
<thead>
<tr>
<th>$C$</th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
<th>$y_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>1/4</td>
<td>1/2</td>
<td>1/4</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1/2</td>
<td>1/3</td>
<td>1/6</td>
<td>0</td>
</tr>
</tbody>
</table>

Joint matrix

<table>
<thead>
<tr>
<th>$J$</th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
<th>$y_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>1/8</td>
<td>1/8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>1/8</td>
<td>1/4</td>
<td>1/8</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1/8</td>
<td>1/12</td>
<td>1/24</td>
<td>0</td>
</tr>
</tbody>
</table>

Distribution on $Y$

| $p_y$ | 1/4 | 1/3 | 7/24 | 1/8 |

Posterior distributions

| $p_{X|y_1}$ | $p_{X|y_2}$ | $p_{X|y_3}$ | $p_{X|y_4}$ |
|-------------|-------------|-------------|-------------|
| $x_1$ | 1/2 | 3/8 | 0 | 0 |
| $x_2$ | 0 | 3/8 | 6/7 | 1 |
| $x_3$ | 1/2 | 1/4 | 1/7 | 0 |
Graphical representation of example

\[ \pi = \left( \frac{1}{4}, \frac{1}{2}, \frac{1}{4} \right) \]
Graphical representation of example
Leakage = mutual information $I(X;Y)$?

- It is tempting to measure leakage using Shannon entropy and mutual information:
  - $H(X) = -\sum x \pi_x \log \pi_x$ “prior uncertainty”
  - $H(X|Y) = \sum y p(y) H[p_X|y]$ “posterior uncertainty”
  - $I(X;Y) = H(X) - H(X|Y)$ “leakage”

- But consider the channel
  \[ Y = \begin{cases} X & \frac{1}{8} \\ -1 & \frac{7}{8} \end{cases} \]

- If $X$ is a uniformly-distributed 64-bit unsigned int, we get $H(X) = 64$, $H(X|Y) = 56$, and $I(X;Y) = 8$ bits.

- We might expect that $H(X|Y) = 56$ means that the secrecy of $X$ has not been harmed much.

- But $\frac{1}{8}$ of the time the adversary learns $X$ exactly!
Vulnerability [Smith09]

- For confidentiality, it seems more useful to measure leakage based on X’s vulnerability to be guessed correctly by A in one try.

- Prior vulnerability:
  \[ V[\pi] = \max_x \pi_x \]

- Posterior vulnerability:
  \[ V[\pi, C] = \sum_y p(y) V[p_x | y] \]
  - \( V[\pi, C] \) is the average vulnerability in the hyper-distribution.
  - \( V[\pi, C] \) is the complement of the Bayes risk.
Operational significance of vulnerability

- $V[\pi]$ is an optimal adversary $A$'s probability of winning the following game:
  $$x \leftarrow \pi$$
  $$w \leftarrow A(\pi)$$
  if $w = x$ then win else lose

- $V[\pi,C]$ is an optimal adversary $A$'s probability of winning the following game:
  $$x \leftarrow \pi$$
  $$y \leftarrow C_{x,-}$$
  $$w \leftarrow A(\pi, C, y)$$
  if $w = x$ then win else lose
Min-entropy leakage is defined multiplicatively:

\[
\mathcal{L}(\pi, C) = \log \frac{V[\pi, C]}{V[\pi]}
\]

Note: -\log V[\pi] is Rényi’s min-entropy \(H_\infty[\pi]\).

Later, we will also consider additive leakage:

\[
\mathcal{L}^+(\pi, C) = V[\pi, C] - V[\pi]
\]
A surprising example ("base-rate fallacy")

Consider a good, but imperfect, test for cancer:

<table>
<thead>
<tr>
<th>$C$</th>
<th>positive</th>
<th>negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancer</td>
<td>9/10</td>
<td>1/10</td>
</tr>
<tr>
<td>no cancer</td>
<td>1/10</td>
<td>9/10</td>
</tr>
</tbody>
</table>

Prior (age 40-50, no symptoms, no family history)

$\pi[\text{cancer}] = 1/100 \quad \pi[\text{no cancer}] = 99/100$
A surprising example ("base-rate fallacy")

- Consider a good, but imperfect, test for cancer:

<table>
<thead>
<tr>
<th>C</th>
<th>positive</th>
<th>negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancer</td>
<td>9/10</td>
<td>1/10</td>
</tr>
<tr>
<td>no cancer</td>
<td>1/10</td>
<td>9/10</td>
</tr>
</tbody>
</table>

- Prior (age 40-50, no symptoms, no family history)
  \[ \pi[\text{cancer}] = 1/100 \quad \pi[\text{no cancer}] = 99/100 \]

<table>
<thead>
<tr>
<th>[\pi, C]</th>
<th>27/250</th>
<th>223/250</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancer</td>
<td>1/12</td>
<td>1/892</td>
</tr>
<tr>
<td>no cancer</td>
<td>11/12</td>
<td>891/892</td>
</tr>
</tbody>
</table>

- A’s best guess is always guess “no cancer”!

- \( V[\pi, C] = 0.99 = V[\pi] \), so \( L(\pi, C) = 0 \).
Limitations of min-entropy leakage

- Vulnerability V has a clear operational significance, but it implicitly assumes that adversary $A$ benefits only by guessing $X$ exactly and in one try.

- But many other scenarios are possible:
  - Maybe $A$ can benefit by guessing $X$ partially or approximately.
  - Maybe $A$ is allowed to make multiple guesses.
  - Maybe $A$ is penalized for making a wrong guess.

- No single leakage measure is appropriate in all scenarios.
We can model each scenario with a gain function $g$.

- Finite set $W$ of guesses (or “actions”) about $X$.
- $g : W \times X \rightarrow [0, 1]$
- $g(w,x)$ gives the value of $w$ if the secret is $x$.
- Note: Ordinary vulnerability implicitly uses $g_{id}(w,x) = \begin{cases} 1, & \text{if } w = x \\ 0, & \text{otherwise} \end{cases}$

- **Prior $g$-vulnerability**: $V_g[\pi] = \max_w \sum_x \pi_x g(w,x)$
  - maximum expected gain over all possible guesses
- **Posterior $g$-vulnerability**: $V_g[\pi, C] = \sum_y p(y) V_g[p_{X|Y}]$
- **$g$-leakage**: $\mathcal{L}_g(\pi, C) = \log \left( \frac{V_g[\pi, C]}{V_g[\pi]} \right)$
Plan of the talk

- Motivation
- Concepts of Quantitative Information Flow
  - Channels, hyper-distributions, vulnerability, min-entropy leakage, g-leakage
- Robustness
  - Robust channel ordering: composition refinement
  - Capacity: multiplicative and additive
- Conclusion
Robustness worries

- Using $g$-leakage, we can express precisely a rich variety of operational scenarios.
- But we could worry about the robustness of our conclusions about leakage.
- The $g$-leakage $L_g(\pi,C)$ depends on both $\pi$ and $g$.
  - $\pi$ models adversary $A$'s prior knowledge about $X$
  - $g$ models (among other things) what is valuable to $A$.
- How confident can we be about these?
- Can we minimize sensitivity to questionable assumptions about $\pi$ and $g$?
Channel ordering

- Given channels C and D on secret input X, the question of which leaks more will ordinarily depend on the prior and gain function used.
- Example: (assume 64-bit, uniform unsigned X)
  
  C. \( Y = X^{\frac{1}{8}} \)  \( Y = -1 \)
  
  D. \( Y = X \lor 0x7; \)

- Both have min-entropy leakage of 61.0 bits.
- We can distinguish them with gain functions.
- \( g_3 \), which allows 3 tries, makes D leak more than C.
- \( g_{\text{tiger}} \), which gives a penalty for a wrong guess (allowing “⊥” for “don’t guess”) makes C leak more.
Robust channel ordering

- Is there a robust ordering?
  - This could support a stepwise refinement methodology.
  - Yes!

- Def: $C$ is composition refined by $D$, written $C \sqsubseteq \circ D$, if $D = CE$, for some channel $E$.
  - $CE$ is the cascade of $C$ and $E$, formed by multiplying the channel matrices $C$ and $E$.
  - Intuitively, the adversary should never prefer $D$ to $C$, since he could do the “post-processing” $E$ himself.
Composition refinement and leakage

- **Theorem [MMSEM14]:**
  - $C$ is composition refined by $D$ iff
  - $D$ never leaks more than $C$, regardless of $\pi$ and $g$.

- The forward direction is a generalized data-processing inequality.
- The backward ("coriaceous") direction uses the separating hyperplane lemma to construct the $g$ needed in the contraposition.
- We later learned that this theorem was proved in 1953 by statistician David Blackwell!
Structure of channels under composition refinement

- Composition refinement is only a **pre-order** on channel matrices.
- But channel matrices contain **redundant structure** with respect to their abstract denotation as mappings from priors to hyper-distributions.

<table>
<thead>
<tr>
<th>C</th>
<th>y₁</th>
<th>y₂</th>
<th>y₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>x₁</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>x₂</td>
<td>1/4</td>
<td>1/2</td>
<td>1/4</td>
</tr>
<tr>
<td>x₃</td>
<td>1/2</td>
<td>1/3</td>
<td>1/6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D</th>
<th>z₁</th>
<th>z₂</th>
<th>z₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>x₁</td>
<td>2/5</td>
<td>0</td>
<td>3/5</td>
</tr>
<tr>
<td>x₂</td>
<td>1/10</td>
<td>3/4</td>
<td>3/20</td>
</tr>
<tr>
<td>x₃</td>
<td>1/5</td>
<td>1/2</td>
<td>3/10</td>
</tr>
</tbody>
</table>

C and D composition refine each other, but they are actually the same abstract channel!

- **Theorem**: On abstract channels, composition refinement is a **partial order**.
Another approach to robustness is to consider \textit{capacity}, the \textbf{maximum} leakage of a channel $C$ over all priors $\pi$ and/or gain functions $g$.

This gives \textbf{worst-case} bounds on leakage.

There are six capacity scenarios:
- multiplicative or additive leakage
- maximize over $\pi$, over $g$, or over \textbf{both} $\pi$ and $g$.

I will discuss just a few of these cases.

A number of them are not well understood yet.
Multiplicative capacity

- Fixing $g$ to $g_{id}$ (giving ordinary vulnerability) and maximizing over $\pi$ gives interesting results.

- **Def:** Min-capacity $\mathcal{ML}(C) = \sup_{\pi} \log(V[\pi,C]/V[\pi])$

- **Theorem:** $\mathcal{ML}(C)$ is the log of the sum of the column maximums of $C$. It is realized on a uniform prior $\pi$.
  - **Corollary:** $\mathcal{ML}(C) = 0$ iff the rows of $C$ are identical.
  - **Corollary:** For deterministic $C$, $\mathcal{ML}(C)$ is the log of the number of feasible output values.
Fixing $g$ to $g_{id}$ (giving ordinary vulnerability) and maximizing over $\pi$ gives interesting results.

**Def:** Min-capacity $\mathcal{ML}(C) = \sup_{\pi} \log(V[\pi,C]/V[\pi])$

**Theorem:** $\mathcal{ML}(C)$ is the log of the sum of the column maximums of $C$. It is realized on a uniform prior $\pi$.

**Corollary:** $\mathcal{ML}(C) = 0$ iff the rows of $C$ are identical.

**Corollary:** For deterministic $C$, $\mathcal{ML}(C)$ is the log of the number of feasible output values.

**Theorem** ("Miracle"): Min-capacity is an upper bound on $g$-leakage, for every prior $\pi$ and gain function $g$: $\mathcal{ML}(C) \geq \mathcal{L}_g(\pi,C)$. 
Example

- Let the secret be an array X containing 10-bit, uniformly distributed passwords for 1000 users.
- Let C be 
  \[ u \leftarrow \{0..999\}; \ Y = (u, X[u]); \]
- \[ V[\pi] = 2^{-10000} \text{ and } V[\pi, C] = 2^{-9990}, \text{ so } \mathcal{ML}(C) = 10 \]
- Now specify that A gains by guessing any user’s password:
  - \[ W = \{ (u, x) \mid 0 \leq u \leq 999 \text{ and } 0 \leq x \leq 1023 \} \]
  - \[ g((u, x), X) = \begin{cases} 
  1, & \text{if } X[u] = x \\
  0, & \text{otherwise} 
\end{cases} \]
  - \[ V_g[\pi] = 2^{-10} \text{ and } V_g[\pi, C] = 1, \text{ so } \mathcal{L}_g(\pi, C) = 10 \]
  - (The Miracle Theorem bounds \( \mathcal{L}_g(\pi, C) \), not \( V_g[\pi, C] \).)
Repeated independent runs $C^{(n)}$

**Theorem:** $ML(C^{(n)}) \leq |Y| \log(n+1)$.

(Useful only when $C$ is probabilistic!)
Repeated independent runs $C^{(n)}$

Theorem: $\mathcal{ML}(C^{(n)}) \leq |\mathcal{Y}| \log(n+1)$.

Application to timing attacks on **blinded** cryptography:
- **Blinding** randomizes the ciphertext before decryption, and de-randomizes after decryption.
- As a result, the $n$-observation timing attack is a repeated independent runs channel $C^{(n)}$.
- Hence its min-capacity grows only logarithmically in $n$. (Useful only when $C$ is probabilistic!)
Theorem: With respect to $g_{id}$, the additive capacity of $C$ over all priors $\pi$ is \textbf{NP-complete}.

Notice that the input here is the \textit{channel matrix} $C$, rather than a concise program.

Theorem: If we fix $\pi$ and universally quantify over $g$ (ranging over “1-spanning” gain functions), then the additive capacity of $C$ is the \textit{Kantorovich distance} between the prior $[\pi]$ and hyper-distribution $[\pi,C]$.

Hence it is the \textit{earth-moving distance} between $[\pi]$ and $[\pi,C]$, which can be computed in time linear in $|C|$.
If we want the additive capacity over all \( g \), we take the average earth-moving distance between \( \pi \) and each of the posterior distributions.

E.g. the distance between \( \pi \) and \((1/2, 0, 1/2)\) is 1/2.

Overall we get

\[
1/4 \cdot 1/2 + 1/3 \cdot 1/8 + 7/24 \cdot 5/14 + 1/8 \cdot 1/2 = 1/3
\]
Plan of the talk

- Motivation
- Concepts of Quantitative Information Flow
  - Channels, hyper-distributions, vulnerability, min-entropy leakage, g-leakage
- Robustness
  - Robust channel ordering: composition refinement
  - Capacity: multiplicative and additive
- Conclusion
Conclusion

- Min-entropy and $g$-leakage allows the quantification of leakage with strong operational significance for confidentiality.

- Composition refinement and capacity support robust conclusions about leakage.

- Research directions:
  - Static analysis of leakage in programs
  - Relationship with differential privacy
  - Computational measures of leakage
  - Generalizing from channels to Hidden Markov Models
Many thanks to my collaborators

Catuscia Palamidessi
Kostas Chatzikokolakis
Annabelle McIver
Carroll Morgan
Máximo Andrés
Mário Alvim
Boris Köpf
Ziyuan Meng
Barbara Espinoza
Marco Stronati
Nico Bordenabe
Questions?