Quantifying information flow using min-entropy and $g$-leakage

Geoffrey Smith
Florida International University

UIUC Summer School lecture, 25 July 2013
Protecting the **confidentiality** of secret information is a fundamental issue in computer security:

- Access control and encryption are important tools.
- But they cannot stop authorized systems from leaking confidential information, maliciously or accidentally, to their publicly observable outputs.

**Crucial (and subtle) question:** what is publicly observable?

<table>
<thead>
<tr>
<th>Blood type:</th>
<th>AB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birth date:</td>
<td>9/5/46</td>
</tr>
<tr>
<td>HIV:</td>
<td><strong>[redacted]</strong></td>
</tr>
</tbody>
</table>
The Denning Restrictions and Noninterference

- Assume X is a secret input, Y a public output.
- [DenningDenning77]
  - Disallow **explicit** flows: \( Y = \frac{X}{3} + 1 \);
  - Disallow **implicit** flows: if \( X \mod 2 == 0 \) \( Y = 0 \); else \( Y = 1 \);
- [VolpanoSmithIrvine96]
  - A type system can enforce the Denning restrictions.
  - Well-typed deterministic programs satisfy a **noninterference** [GoguenMeseguer82] property:
    - Running with two different initial values of X gives the **same** final value of Y (so long as both runs terminate successfully).
- [SabelfeldMyers03] survey.
Unfortunately, noninterference is often too strong

Password checking

Election tabulation

Timings of decryptions
Can we quantify the flow of information to an adversary $A$ who sees the observable output?

If so, then “small” leaks can perhaps be tolerated.

This has been an active area of research for the past decade [ClarkHuntMalacaria02, ...]

A first, clearcut, example: $Y = X \& 0x01ff$;

If $X$ is a 64-bit integer with all $2^{64}$ values equally likely, then this program leaks 9 bits (out of 64) to $Y$. 
Plan of the talk

- Motivation
- Information-theoretic channels
- Vulnerability and min-entropy leakage
- Gain functions and g-leakage
- Comparing channels
Information-theoretic channels

- Finite sets $X$ (secret inputs), $Y$ (observable outputs).
- On input $X$, the channel probabilistically outputs $Y$.
- A channel matrix $C$ specifies the behavior:

<table>
<thead>
<tr>
<th></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</td>
<td>0</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.
- Special case: $C$ is deterministic if each row contains exactly one 1.
Prior and posterior distributions

- We assume that $X$ has some prior distribution $\pi$.
- We assume that adversary $A$ knows both $\pi$ and $C$.
- Seeing output $y$, $A$ can inspect column $y$ of $C$ to update prior $\pi$ to posterior distribution $p_{X|y}$.

For example, seeing output $y_4$ tells $A$ that $X$ was $x_2$.

Leakage depends on the way $C$ maps prior distribution $\pi$ to a set of posterior distributions.
Abstractly, a channel is a function from priors to hyper-distributions.
Plan of the talk

- Motivation
- Information-theoretic channels
- Vulnerability and min-entropy leakage
- Gain functions and g-leakage
- Comparing channels
Measuring confidentiality

- Assume $X$ is a random variable with distribution $\pi$.
- Assume, for the worst case, that $\pi$ is known to the adversary $\mathcal{A}$.
- Initially, how “confidential” or “uncertain” is $X$ to $\mathcal{A}$?
- Shannon entropy [1948] is a classic measure:
  \[ H(\pi) = -\sum_x \pi(x) \log \pi(x) \]
- But its operational significance for confidentiality is quite weak:
  - If $\pi = (1/2, 2^{-1000}, 2^{-1000}, 2^{-1000}, 2^{-1000}, ..., 2^{-1000})$, then $H(\pi) = 500.5$ bits.
  - But half the time $\mathcal{A}$ can guess $X$ correctly in one try!
So [Smith09] proposed to focus instead on X's vulnerability to be guessed by $\mathcal{A}$ in one try:

**Definition:** $V(\pi) = \max_x \pi[x]$

- If $\pi = (1/2, 2^{-1000}, 2^{-1000}, 2^{-1000}, ..., 2^{-1000})$, then $V(\pi) = 1/2$.

**Posterior vulnerability** is the average vulnerability after $Y$ is observed:

**Definition:** $V(\pi, C) = \sum_y p(y) V(p_{X|Y})$

- Equivalently, $V(\pi, C) = \sum_y \max_x p(x,y)$
  - “sum of column maximums of joint matrix”

$V(\pi, C)$ is the complement of the Bayes risk.
V(\(\pi\)) and V(\(\pi,C\)) on example channel

- V(\(\pi\)) = \(\max_x \pi[x]\) = 1/2
- V(\(\pi,C\)) = \(\sum_y \max_x p(x,y)\) = 1/4 + 1/8 + 1/4 + 1/8 = 3/4
- A priori, \(\mathcal{A}'s\) best guess is that X is x₂.
- A posteriori, \(\mathcal{A}'s\) best guess for X depends on Y:
  \[\gamma_1 \rightarrow x_1, \quad \gamma_2 \rightarrow x_2, \quad \gamma_3 \rightarrow x_2, \quad \gamma_4 \rightarrow x_2\]

| Prior | Channel matrix \(\pi\) | | | \(\pi\) | | | \(\pi\) | | | \(\pi\) | | | \(\pi\) | | | \(\pi\) |
|-------|-----------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 1/4   | \(x_1\) | 1  | 0  | 0  | 0  | \(x_1\) | 1/4 | 0  | 0  | 0  | \(x_1\) | 1/4 | 0  | 0  | 0  |
| 1/2   | \(x_2\) | 0  | 1/4| 1/2| 1/4| \(x_2\) | 0  | 1/8| 1/4| 1/8| \(x_2\) | 0  | 1/8| 1/4| 1/8|
| 1/4   | \(x_3\) | 1/2| 1/3| 1/6| 0  | \(x_3\) | 1/8| 1/12| 1/24| 0  | \(x_3\) | 1/8| 1/12| 1/24| 0  |

<table>
<thead>
<tr>
<th>Joint matrix</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>1/4</td>
<td>0</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>
Definition: Min-entropy leakage

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

Convert from vulnerability to uncertainty, in bits, by taking the negative log, giving min-entropy [Rényi61].

- \( H_\infty(\pi) = -\log V(\pi) \) “prior uncertainty”
- \( H_\infty(\pi, C) = -\log V(\pi, C) \) “posterior uncertainty”
  - Not defined by Rényi. \( \text{Not } H_\infty(\pi, C) = \sum_y p(y) H_\infty(p_{X|y}) \).
  - So, equivalently, \( \mathcal{L}(\pi, C) = H_\infty(\pi) - H_\infty(\pi, C) \).
Min-capacity

- Sometimes the prior distribution $\pi$ might be unknown.
- In that case, it is useful to abstract away from it, considering the maximum leakage, over all $\pi$.

**Definition:** Min-capacity

$$\mathcal{ML}(C) = \sup_{\pi} \mathcal{L}(\pi, C)$$

**Remark:** Min-capacity is the min-entropy analogue of Shannon capacity, which is the maximum mutual information $I(X;Y)$ over all priors $\pi$. 
Properties of min-entropy leakage

- **Theorem**: $V(\pi, C) \geq V(\pi)$, so $\mathcal{L}(\pi, C) \geq 0$.

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

- **Corollary**: If $C$ is deterministic, then $\mathcal{ML}(C)$ is the log of the number of feasible outputs.

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

- $\mathcal{L}(\pi, C) = 0$ if $X$ and $Y$ are independent. Not conversely!

- Indeed $\mathcal{L}(\pi, C) = 0$ if $Y$ never affects $A$’s best guess.
Consider a good, but imperfect, test for cancer:

<table>
<thead>
<tr>
<th></th>
<th>positive</th>
<th>negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancer</td>
<td>0.90</td>
<td>0.10</td>
</tr>
<tr>
<td>no cancer</td>
<td>0.07</td>
<td>0.93</td>
</tr>
</tbody>
</table>

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

\[ \pi[\text{cancer}] = 0.008 \quad \pi[\text{no cancer}] = 0.992 \]

<table>
<thead>
<tr>
<th></th>
<th>positive</th>
<th>negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancer</td>
<td>0.00720</td>
<td>0.00080</td>
</tr>
<tr>
<td>no cancer</td>
<td>0.06944</td>
<td>0.92256</td>
</tr>
</tbody>
</table>

\[ V(\pi,C) = 0.992 = V(\pi), \text{ so } L(\pi,C) = 0. \]

Always guess “no cancer”! \[ p_X|_{\text{positive}} \approx (0.094, 0.906) \]
Min-capacity versus Shannon capacity

**Theorem**: On deterministic channels, min-capacity and Shannon capacity *coincide*.

On *probabilistic* channels, min-capacity can exceed Shannon capacity by an arbitrary factor:

$$\begin{array}{cccc}
2^{-10} & 2^{-64} & 2^{-64} & \ldots \\
2^{-64} & 2^{-10} & 2^{-64} & \ldots \\
2^{-64} & 2^{-64} & 2^{-10} & \ldots \\
\ldots & \ldots & \ldots & \ldots \\
2^{-64} & 2^{-64} & 2^{-64} & 2^{-10}
\end{array}$$

Shannon capacity $\approx 0.05$

Min-capacity $\approx 54.00$

**Theorem**: Shannon capacity is always less than or equal to min-capacity.
Channels in Cascade $C_1C_2$

- Formed by **multiplying** two channel matrices, or by **factoring** a channel matrix.

- **Theorem** [EspinozaSmith11]: $\mathcal{L}(\pi,C_1C_2) \leq \mathcal{L}(\pi,C_1)$
  - Analogue of the data-processing inequality.
  - Curiously, we can have $\mathcal{L}(\pi,C_1C_2) > \mathcal{L}(p_Y,C_2)$.

- **Theorem**: $\mathcal{ML}(C_1C_2) \leq \min \{ \mathcal{ML}(C_1), \mathcal{ML}(C_2) \}$
Repeated independent runs $C^{(n)}$

- $C^{(n)}[x, (y_1, y_2, \ldots, y_n)] = \prod_i C[x, y_i]$
- We find that $\mathcal{ML}(C^{(n)})$ grows only logarithmically in $n$.
- Theorem [KöpfSmith10]: $\mathcal{ML}(C^{(n)}) \leq |\mathcal{Y}| \log(n+1)$.
  - Here $\mathcal{Y}$ is the set of feasible values of $\mathcal{Y}$.
  - The proof factors $C^{(n)}$ into the cascade of two channels with a small set $T'$ of intermediate values.
Application: timing attacks on cryptography

- [KöpfSmith10] proved the effectiveness of **blinding** and **bucketing** against timing attacks.
- **Blinding**: randomize ciphertext before decryption; de-randomize after decryption.
- **Bucketing**: force decryption to take one of a small number of possible times.
- **Theorem**: With blinding and bucketing, the number of min-entropy bits leaked is logarithmic in the number of timing observations.
  - Because of blinding, the n-observation timing attack is a repeated independent runs channel $C^{(n)}$.
  - Because of bucketing, $|\mathcal{Y}|$ is small.
On deterministic programs, the min-capacity (and Shannon capacity) is the log of the number of feasible final values of \( Y \).

Example (\( X \) and \( Y \) are 32-bit unsigned integers):

\[
\begin{align*}
Y &= ((X \gg 16) \, ^{\land} \, X) \, & & \& \, 0xffff; \\
Y &= Y \, | \, Y \, << \, 16;
\end{align*}
\]

\( Y \) has \( 2^{16} \) feasible values, since its left and right halves will be equal.

Hence the min-capacity is \( \log 2^{16} = 16 \) bits.
Bounding min-capacity using two-bit patterns

[MengSmith11]

- Determine **patterns** that the bits of Y must satisfy.
- Single bits can be either **Zero**, **One**, or **Non-fixed**.
- On the example, all one-bit patterns are **Non-fixed**.
- Pairs of Non-fixed bits satisfy one of seven relations: **Eq**, **Neq**, **Geq**, **Leq**, **Nand**, **Or**, **Free**.
- On the example, we get 16 non-Free patterns:
  - Eq(16,0), Eq(17,1), Eq(18,2), ..., Eq(31,15)
- The number of satisfying assignments to the bit patterns is an upper bound on the number of feasible outputs. (Here it’s $2^{16}$, which is exact.)
Plan of the talk

- Motivation
- Information-theoretic channels
- Vulnerability and min-entropy leakage
- Gain functions and $g$-leakage
- Comparing channels
Min-entropy leakage implicitly assumes that adversary $A$ benefits only by guessing the secret exactly, in one try.

But what about other scenarios? What if $A$ benefits by guessing the secret only partially or approximately? Or if $A$ is allowed to make multiple guesses? Can any single leakage measure be appropriate in all scenarios?
When min-entropy leakage is misleading: an example

- Let the secret be an array $X$ containing 10-bit, uniformly distributed passwords for 1000 users.

- Consider the following probabilistic channel, which leaks some randomly-chosen user’s password:
  \[ u \leftarrow \{0..999\}; \; Y = (u, X[u]); \]

- The min-entropy leakage is $L = 10$ bits out of 10000.

- Looking instead at the vulnerability of some particular user $i$’s password, we get a leakage of 1.016 out of 10 bits.

- Neither of these results expresses that some user’s password is always leaked!
Gain functions and g-leakage
[AlvimChatzikokolakisPalamidessiSmith12]

- We generalize min-entropy leakage by introducing gain functions to model the operational scenario.
- In any scenario, there is a set $W$ of guesses that $A$ can make about the secret.
- For each guess $w$ and secret value $x$, there is a gain $g(w,x)$ that $A$ gets by choosing $w$ when the secret’s actual value is $x$.

**Definition**: gain function $g : W \times X \rightarrow [0, 1]$.

**Example**: Min-entropy leakage implicitly uses $g_{id}(w,x) = \begin{cases} 1, & \text{if } w = x \\ 0, & \text{otherwise} \end{cases}$
Definition: Prior g-vulnerability:

\[ V_g(\pi) = \max_w \sum_x \pi[x]g(w,x) \]

“A’s maximum expected gain, over all possible guesses.”

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

- 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: \[ L_g(\pi, C) = \log \frac{V_g(\pi, C)}{V_g(\pi)} \]
- g-capacity: \[ M L_g(C) = \sup_\pi L_g(\pi, C) \]
The power of gain functions

Guessing a secret approximately.
\[ g(w, x) = 1 - \text{dist}(w, x) \]

Guessing a property of a secret.
\[ g(w, x) = \text{Is } x \text{ of gender } w? \]

Guessing a part of a secret.
\[ g(w, x) = \text{Does } w \text{ match the high-order bits of } x? \]

Guessing a secret in 3 tries.
\[ g_3(w, x) = \text{Is } x \text{ an element of set } w \text{ of size 3?} \]

Lab location:
\[
\begin{align*}
N & 39.95185 \\
W & 75.18749
\end{align*}
\]

Dictionary:
- superman
- apple-juice
- johnsmith62
- secret.flag
- history123
- ...
A gain function for our password array example

- We specify that guessing any user's password is bad:
  - \( W = \{ (u, x) \mid 0 \leq u \leq 999 \text{ and } 0 \leq x \leq 1023 \} \)
  - \( g((u, x), X) = \begin{cases} 1, & \text{if } X[u] = x \\ 0, & \text{otherwise} \end{cases} \)
  - We get \( V_g(\pi) = 2^{-10} \) (as compared with \( V(\pi) = 2^{-10000} \)).
  - We get \( V_g(\pi, C) = 1 \) (as compared with \( V(\pi, C) = 2^{-9990} \)).
  - So the g-leakage is 10 bits out of 10.
  - (Recall that the min-entropy leakage is 10 bits out of 10000.)

- Remark: If the channel had instead leaked the last bit of 10 random passwords, then its g-leakage would be only 1 bit.
  Under g, some groups of 10 bits are worth more than others!
Properties of g-leakage and g-capacity

- No general relation holds between g-leakage and min-entropy leakage. Each can be greater than the other.

- **Theorem** ("Miracle"): Min-capacity is an upper bound on g-capacity, for every gain function g.
  - Hence a channel with small min-capacity has small g-leakage under every prior and every gain function.
  - (Of course, g does affect the prior g-vulnerability...)

- **Corollary**: A channel’s k-tries capacity cannot exceed its 1-try capacity.
Plan of the talk

- Motivation
- Information-theoretic channels
- Vulnerability and min-entropy leakage
- Gain functions and g-leakage
- Comparing channels
The Lattice of Information

- A deterministic channel $C$ from $X$ to $Y$ induces a partition on $X$. $x_1$ and $x_2$ are in the same block iff they map to the same output.
  - Example: $C_{\text{country}}$ maps a person $x$ to the country of birth:
    
    
    
    
    $C_{\text{country}}$

- **Partition refinement** $\sqsubseteq$: Subdivide zero or more of the blocks.
  - Example: $C_{\text{state}}$ also includes the state of birth for Americans:
    
    
    
    
    $C_{\text{state}}$

- $C_{\text{country}} \sqsubseteq C_{\text{state}}$
Leakage testing of deterministic channels

- **Theorem** [YasuokaTerauchi10, Malacaria11]
  \[ C_1 \sqsubseteq C_2 \]
  iff
  \[ C_1 \text{ never leaks more than } C_2 \], on any prior, under any of the usual leakage measures (Shannon-, min-, or guessing entropy).

- Hence we have a partial order on deterministic channels with both a structural and a testing characterization.

- Can we generalize it to probabilistic channels?
Orderings on probabilistic channels

- **Def:** $C_1 \sqsubseteq_o C_2$ ("$C_1$ is composition refined by $C_2$") if $C_1$ factors as $C_2$ followed by some post-processing $C_3$; that is, $C_1 = C_2C_3$ for some $C_3$.

- On deterministic channels, composition refinement $\sqsubseteq_o$ coincides with partition refinement $\sqsubseteq$.

- So $\sqsubseteq_o$ generalizes $\sqsubseteq$ to probabilistic channels.

- **Def:** $C_1 \leq_G C_2$ ("$C_1$ never out-leaks $C_2$") if the $g$-leakage of $C_1$ never exceeds that of $C_2$, for any prior $\pi$ and gain function $g$. 
Relationship between $\sqsubseteq_0$ and $\leq_G$

- **Theorem**: [Generalized data-processing inequality]
  
  If $C_1 \sqsubseteq_0 C_2$ then $C_1 \leq_G C_2$.
  
  - Shown in [ACPS12].

- **Theorem**: ["Coriaceous Conjecture"]
  
  If $C_1 \leq_G C_2$ then $C_1 \sqsubseteq_0 C_2$.
  
  - Conjectured in [ACPS12], resolved using ideas in [McIverMeinickeMorgan10].

- So we have a partial order on probabilistic channels, with both **structural** and **leakage-testing** significance.
Conclusion

Min-entropy leakage and g-leakage offer nice opportunities for
- theory,
- applications to real security threats, and
- engineering of static analysis techniques.

Thanks to all my collaborators:
(LIX) Catuscia Palamidessi, Kostas Chatzikokolakis, Miguel Andrés, Mário Alvim
(IMDEA) Boris Köpf
(FIU) Ziyuan Meng, Barbara Espinoza
(Australia) Carroll Morgan, Annabelle McIver
Some references

Questions?