Quantitative Information Leakage

Lecture 10

Limitations of min-entropy leakage

- Min-entropy leakage implicitly assumes an operational scenario where adversary $\mathcal{A}$ benefits only by guessing secret S exactly, and in one try.
- But many other scenarios are possible:
  - Maybe $\mathcal{A}$ can benefit by guessing S partially or approximately.
  - Maybe $\mathcal{A}$ is allowed to make multiple guesses.
  - Maybe $\mathcal{A}$ is penalized for making a wrong guess.
- How can any single leakage measure be appropriate in all scenarios?

Notation

- π prior probability
- $x, x_1, x_2 \ldots \quad X$ secrets
- $x, y_1, y_2 \ldots \quad Y$ observables
- $w, w_1, w_2 \ldots \quad W$ guesses
  (they may be different from the secrets)

Gain functions and g-leakage

- We generalize min-entropy leakage by introducing gain functions to model the operational scenario.
- In any scenario, there is a finite set $\mathcal{W}$ of guesses that $\mathcal{A}$ can make about the secret.
- For each guess w and secret value x, there is a gain g(w,x) that $\mathcal{A}$ gets by choosing w when the secret's actual value is x.
- **Definition**: gain function $g : \mathcal{W} \times \mathcal{X} \rightarrow [0, 1]$
- **Example**: Min-entropy leakage implicitly uses

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

g-vulnerability and g-leakage

- Definition: Prior g-vulnerability:

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

"$\mathcal{A}$'s maximum expected gain, over all possible guesses."

- Posterior g-vulnerability:

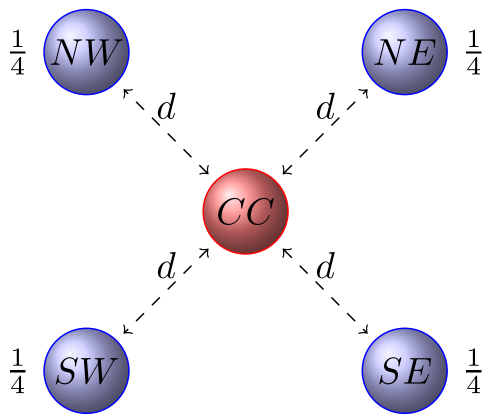
$$V_g[\pi,C] = \sum_y p(y)\, V_g[p_{X|y}]$$

- g-leakage: $\mathcal{L}_g(\pi,C) = \log V_g[\pi,C] - \log V_g[\pi]$
- g-capacity: $\mathcal{ML}_g(C) = \sup_\pi \mathcal{L}_g(\pi,C)$

The power of gain functions

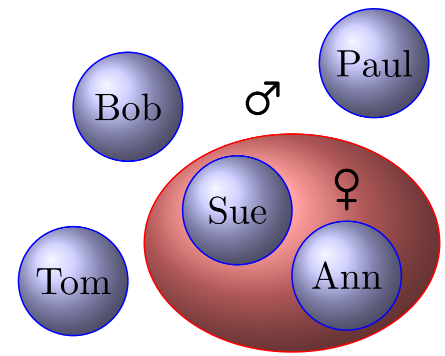
Guessing a secret approximately.

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



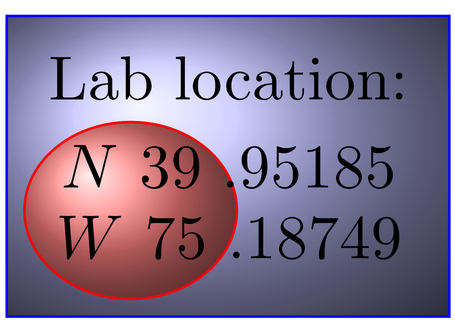
Guessing a property of a secret.

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



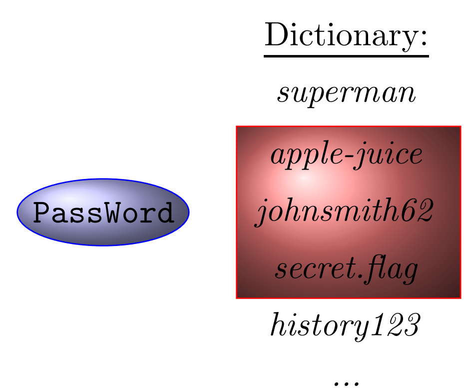
Guessing a part of a secret.

$g(w, x)$ = 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?



Distinguishing channels with gain functions

- Two channels on a uniformly distributed, 64-bit x:

  A. y = (x or 00000… 0111);

  B. if (x % 8 == 0) then y = x; else y = 1;

  - A always leaks all but the last three bits of x.
  - B leaks all of x one-eighth of the time, and almost nothing seven-eighths of the time.
  - Both have min-entropy leakage of 61 bits out of 64.
- We can distinguish them with gain functions.
- g_8, which allows 8 tries, makes A worse than B.
- g_{tiger}, which gives a penalty for a wrong guess (allowing "⊥" to mean "don't guess") makes B worse.

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 $\mathcal{L}_g(\pi, C)$ depends on both π and g.
  - π models adversary $\mathcal{A}$'s prior knowledge about X
  - g models (among other things) what is valuable to $\mathcal{A}$.
- How confident can we be about these?
- Can we minimize sensitivity to questionable assumptions about π and g?

Capacity results

- **Capacity** (the maximum leakage over all priors) eliminates assumptions about the prior π.
- Capacity relationships between **different** leakage measures are particularly useful.
- **Theorem**: Min-capacity is an upper bound on Shannon capacity: $\mathcal{ML}(C) \geq \mathcal{SC}(C)$.
- **Theorem** ("Miracle"): Min-capacity is an upper bound on g-capacity, for **every** g: $\mathcal{ML}(C) \geq \mathcal{ML}_g(C)$.
  - Hence if C has small min-capacity, then it has small g-leakage under every prior and every gain function.
  - (Note that the choice of g does affect both the prior and the posterior g-vulnerability.)

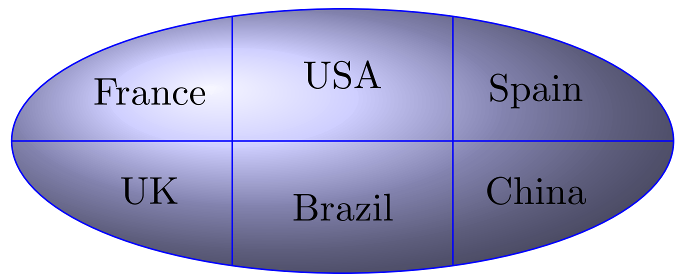
Robust channel ordering

- Given channels A and B on secret input X, the question of which leaks more will usually depend on the prior and the particular gain function used.
- Is there a **robust** ordering?
  - This could allow a stepwise refinement methodology.
  - This is arguably **indispensable** for security.
- For **deterministic** channels, a robust ordering has long been understood: the Lattice of Information [Landauer & Redmond '93].

The Lattice of Information

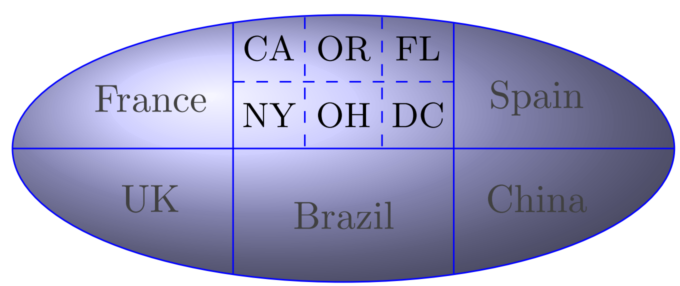
- A deterministic channel from X to Y induces a partition on $\mathcal{X}$: secrets are in the same block iff they map to the same output.
  - Example: $C_{country}$ maps a person x to the country of birth.

$C_{country}$'s partition:



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

C_{state}'s partition:



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

Partition refinement and leakage

- If A $\sqsubseteq$ B, then B leaks at least as much as A under **any** of the standard leakage measures (Shannon-, min-, and guessing entropy. The latter is the expected number of questions of the form "is S=s?" to figure out the secret entirely).
- Interestingly, the converse also holds:
  Theorem [Yasuoka &Terauchi '10, Malacaria '11]

  A $\sqsubseteq$ B

  iff

  A never leaks more than B on any prior, under **any** of the standard leakage measures
- Hence $\sqsubseteq$ is an ordering on deterministic channels with **both** a structural and a leakage-testing characterization.
- Can we generalize it to probabilistic channels?

Composition refinement

- Note that $C_{country}$ is the **composition** of C_{state} and C_{merge}, where C_{merge} post-processes by mapping all American states to USA.

$$C_{country} = C_{state} \, C_{merge}$$

- **Def**: $A \sqsubseteq_{\circ} B$ ("A is **composition refined** by B") if there exists a (post-processing) C such that A = BC.
- On deterministic channels, composition refinement $\sqsubseteq_{\circ}$ **coincides** with partition refinement $\sqsubseteq$.
  - So $\sqsubseteq_{\circ}$ **generalizes** $\sqsubseteq$ to probabilistic channels.

Strong leakage ordering

- **Def:** $A \leq_{\mathcal{G}} B$ ("A never out-leaks B") if the g-leakage of A never exceeds that of B, for any prior π and any gain function g.

A =

	z_1	z_2
x_1	2/3	1/3
x_2	2/3	1/3
x_3	1/4	3/4

B =

	y_1	y_2	y_3
x_1	1/2	1/2	0
x_2	1/2	0	1/2
x_3	0	1/2	1/2

- **Def:** $A \leq_{min} B$ if the min-entropy leakage of A never exceeds that of B, for any prior π.
- It turns out that $A \leq_{min} B$, even though $A \not\leq_{\mathcal{G}} B$

Relationship between $\sqsubseteq_\circ$ and $\leq_{\mathcal{G}}$

- **Theorem**: [Generalized data-processing inequality]

  If A $\sqsubseteq_\circ$ B then A $\leq_{\mathcal{G}}$ B.

  - Intuitively, the adversary should never prefer BC to B.

- **Theorem**: ["Coriaceous"]

  If A $\leq_{\mathcal{G}}$ B then A $\sqsubseteq_\circ$ B.

  - Conjectured for a long time. Proved by McIver et al. in 2014 using geometrical techniques (the Separating Hyperplane Lemma).

- So we have an ordering of probabilistic channels, with both **structural** and **leakage-testing** significance.

Mathematical structure of channels under $\sqsubseteq_\circ$

- $\sqsubseteq_\circ$ 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.

C	y_1	y_2	y_3
x_1	1	0	0
x_2	1/4	1/2	1/4
x_3	1/2	1/3	1/6

D	z_1	z_2	z_3
x_1	2/5	0	3/5
x_2	1/10	3/4	3/20
x_3	1/5	1/2	3/10

C and D are actually the same abstract channel!

- **Theorem:** On abstract channels, $\sqsubseteq_\circ$ is a **partial order**.
  - But it is **not** a lattice.

Exercises

Consider again the two programs A and B on a uniformly distributed, 64-bit x:

A. y = (x or 00000… 0111);

B. if (x % 8 == 0) then y = x; else y = 0;

8. Show that they both have min-entropy leakage 61 bits.
9. Define g_8, which allows 8 tries, and show that it makes A worse than B.
10. Define g_{tiger}, which gives a penalty for a wrong guess (allowing guess "⊥" to mean "don't guess") and show that it makes B worse. For simplicity, allow g_{tiger} to range in [-1,1]

Thank you !