Plan of the course

Quantitative Information Flow

- Motivation, application examples
- Secrets and vulnerability
- Channels and leakage
- Multiplicative Bayes-Capacity
- Comparing systems, the lattice of information
- Applications and exercises
A **location-based system** is a system that uses geographical information in order to provide a service.

- Retrieval of Points of Interest (POIs).
- Mapping Applications.
- Deals and discounts applications.
- Location-Aware Social Networks.
Location-Based Systems

- Find restaurants within 300 meters.
- Hide location, **not** identity.
- Provide **approximate** location.
Solution: obfuscation

area of interest

reported position
Solution: obfuscation

area of retrieval

area of interest
Solution: obfuscation
Issues to study

• How can we generate the noise?

• What kind of formal privacy guarantees do we get?

• Which mechanism gives optimal utility?

• What if we use the service repeatedly?
Timing Attacks in Cryptosystems

• Remote timing attack [BonehBrumley03]

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

• Response time depends on the key!
Timing Attacks in Cryptosystems

• What **counter-measures** can we use?

• Make the decryption time **constant**
  - Too **slow**!

• Force the **set of possible decryption times** to be **small**
  - Is it enough?
  - Must be combined with **blinding**
  - **Careful analysis** of the privacy guarantees is required
How can we measure information leakage?
What is a secret?

• My password: $x = \text{"dme3@21!SDFm12"}$

• What does it mean for this password to be secret?

• The adversary should not know it, i.e. $x$ comes from a set $\mathcal{X}$ of possible passwords

• Is $x' = 123$ an equally good password? Why?

• Passwords are drawn randomly from a probability distribution
What is a secret?

• A secret $x$ is something about which the adversary knows only a probability distribution $\pi$

• $\pi$ is called the adversary’s prior knowledge
  - $\pi$ could be the distribution from which $x$ is generated
  - or it could model the adversary’s knowledge on the population the user comes from

• How vulnerable is $x$?

• It’s a property of $\pi$, not of $x$
• How vulnerable is our secret under prior $\pi$?

• The answer highly depends on the application

• Eg: assume uniformly distributed secrets but the adversary knows the first 4 bytes (this can be expressed by a prior $\pi$)

• Is this threat substantial?
  - No: if the secrets are long passwords
  - Yes: if the first 4 bytes is a credit card pin
Vulnerability

• To quantify the threat we need an operational scenario

• What is the goal of the adversary?

• How successful is the adversary in achieving this goal?

• Vulnerability: measure of the adversary’s success
  Uncertainty: measure of the adversary’s failure
First approach

• Assume the adversary can ask for properties of $x$
i.e. questions of the form “is $x \in P$?”

• **Goal:** completely reveal the secret as quickly as possible

• Measure of success: expected number of steps
First approach

eg:

- $\mathcal{X} = \{\text{heads, tails}\}, \pi = (1, 0)$
- $\mathcal{X} = \{\text{heads, tails}\}, \pi = (1/2, 1/2)$
- $\mathcal{X} = \{a, b, c, d, e, f, g, h\}, \pi = (1/8, \ldots, 1/8)$
- $\mathcal{X} = \{a, b, c, d, e, f, g, h\}$
  $\pi = (1/4, 1/4, 1/8, 1/8, 1/16, 1/16, 1/16, 1/16)$
Best strategy: at each step split the search space in sets of equal probability mass
First approach

• At step $i$ the total probability mass is $2^{-i}$.

• If the probability of $x$ is $\pi_x = 2^{-i}$, then at step $i$ the search space will only contain $x$. So it will take $i = -\log_2 \pi_x$ steps to reveal $x$.

• So the expected number of steps is:

$$\sum_x -\pi_x \log_2 \pi_x$$

(if $\pi$ is not powers of 2 this will be a lower bound)
Shannon Entropy

• This is the well known formula of Shannon Entropy:

\[ H(\pi) = \sum_x -\pi_x \log_2 \pi_x \]

• It’s a measure of the adversary’s uncertainty about \( x \)

• Minimum value: \( H(\pi) = 0 \) iff \( \pi_x = 1 \) for some \( x \)

• Maximum value: \( H(\pi) = \log_2 |\mathcal{X}| \) iff \( \pi \) is uniform
Shannon Entropy

The binary case $\mathcal{X} = \{0, 1\}$, $\pi = (x, 1 - x)$:
Shannon Entropy

• Very widely used to measure information flow

• Is it always a good uncertainty measure for privacy?

• Example: $\mathcal{X} = \{0, \ldots, 2^{32}\}$, $\pi = (\frac{1}{8}, \frac{7}{8} 2^{-32}, \ldots, \frac{7}{8} 2^{-32})$

• $H(\pi) = 28.543525$

• But the secret can be guessed with probability $1/8$!

• Undesired in many practical scenarios
Bayes Vulnerability

• **Adversary’s goal:** correctly guess the secret in one try

• **Measure of success:** probability of a correct guess

• **Optimal strategy:** guess the $x$ with the highest $\pi_x$

• **Bayes Vulnerability:**

$$V_b(\pi) = \max_x \pi_x$$
• **Maximum** value: $V_b(\pi) = 1$ iff $\pi_x = 1$ for some $x$

• **Minimum** value: $V_b(\pi) = 1/|\mathcal{X}|$ iff $\pi$ is uniform

• **Min-entropy**: $\log_2(V_b(\pi))$

  **Bayes risk**: $1 - V_b(\pi)$

• Previous example: $\pi = (\frac{1}{8}, \frac{7}{8}2^{-32}, \ldots, \frac{7}{8}2^{-32})$

  $V_b(\pi) = 1/8$
The binary case \( \mathcal{X} = \{0, 1\} \):
Guessing Entropy

- **Adversary’s goal:** correctly guess the secret in many tries
- **Measure of success:** expected number of tries
- **Optimal strategy:** try secrets in **decreasing** order of probability
- **Guessing entropy:**
  \[ G(\pi) = \sum_{i=1}^{\mid\mathcal{X}\mid} i \cdot \pi_{x_i} \]
  \(x_i\): indexing of \(\mathcal{X}\) in decreasing probability order
Guessing Entropy

- **Minimum value:** $G(\pi) = 1$ iff $\pi_x = 1$ for some $x$

- **Maximum value:** $G(\pi) = (|X| + 1)/2$ iff $\pi$ is uniform

- The binary case:
• What if the adversary wants to reveal part of a secret?

• Or is satisfied with an approximative value?

• Or we are interested in the probability of guessing after multiple tries?
Example

- **Secret**: database of 10-bit passwords for 1000 users:
  \[ \text{pwd}_0, \text{pwd}_1, \ldots, \text{pwd}_{999} \]

- The adversary knows that the password of some user is \( z \), but does not know which one (all are equally likely)

- \( A_1 \): guess the complete database
  \[ V_b(\pi) = 2^{-9990} \]

- \( A_2 \): guess the password of a particular user \( i \)
  - Create distribution \( \pi_i \) for that user
  \[ V_b(\pi_i) = 0.001 \cdot 1 + 0.999 \cdot 2^{10} \approx 0.00198 \]

- \( A_3 \): guess the password of any user
  - intuitively, the secret is completely vulnerable
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:

$$g : \mathcal{W} \times \mathcal{X} \to \mathbb{R}$$

• Success measure: the expected gain of a best guess
\textit{g-vulnerability}

- **Expected gain** under guess $\mathcal{W}$: $\sum_x \pi_x g(w, x)$

- Choose the one that \textbf{maximizes} the gain

- **$g$-vulnerability**: 
  \[ V_g(\pi) = \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} \pi_x g(w, x) \]
  (sup if $\mathcal{W}$ is infinite)

- **$l$-uncertainty** measures can be also defined using \textbf{loss functions}:
  \[ U_l(\pi) = \min_w \sum_x \pi_x l(w, x) \]
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:  

\[ N 39.95185 \quad W 75.18749 \]

Dictionary:  

- superman  
- apple-juice  
- johnsmith62  
- secret.flag  
- history123  
- ...
Password database example

**Secret:** database of 10-bit passwords for 1000 users: 
\( \text{pwd}_0, \text{pwd}_1, \ldots, \text{pwd}_{999} \)

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

\( \mathcal{W} = \{ p | p \in \{0 \ldots 1023\} \} \)

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

\( V_g(\pi) = 1 \)
Expressiveness of $V_g$

• Can we express Bayes-vulnerability using $g$-vulnerability?

• $\mathcal{A}$ guesses the exact secret $x$ in one try

• Guesses $\mathcal{W} = \mathcal{X}$

• Gain function:

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

• $V_{g_{id}}$ coincides with $V_b$
Expressiveness of \( V_g \)

- What about guessing entropy?
- It’s an uncertainly measure, so we need loss functions
- Guesses \( \mathcal{W} = \text{permutations of } \mathcal{X} \)
  eg \( w_1 = (x_3, x_1, x_2) \)
  - Think of \( w \) as the order of guesses
- Loss function: \( l_G(w, x) = i \)
  where \( i \) is the position of \( x \) in the permutation \( w \)
- \( U_{l_g}(\pi) \) coincides with \( G(\pi) \)
Expressiveness of $V_g$

• What about Shannon entropy?

• Again we need loss functions, and infinitely many guesses

• Guesses $\mathcal{W} = \text{probability distributions over } \mathcal{X}$
  - Think of $w$ as a way to construct the search tree

• Loss function: $l_S(w, x) = -\log_2 w_x$
  (number of questions to find $x$ using this tree)

• Because of Gibb’s inequality: $U_{l_S}(\pi) = H(\pi)$

• We can restrict to countably many guesses
Expressiveness of $V_g$

- What other measures can we express as $V_g, U_l$?

- What is a reasonable uncertainly measure $f$?

- Let’s fix some desired properties of $f$