Information Theory - Speaker's Notes (15 min)

Perfect Security (0-2:30)

[BOARD] Write: $P[x|y] = P[x]$

Definition: A cryptosystem has perfect security if for all $x \in P$ and $y \in C$, it holds that $P[x|y] = P[x]$

- Core idea: seeing ciphertext gives adversary zero information
- Plaintext distribution unchanged after observing ciphertext
- By Bayes' theorem: equivalent to $P[y|x] = P[y]$
- Ciphertext distribution independent of plaintext choice
- Extremely strong requirement—few systems achieve this

[TRANSITION] "To understand the cost of perfect security, we need Shannon's concept of entropy—how much information is actually in our messages and keys?"

Entropy (2:30-6:30)

[BOARD] Write: $H(X) = \sum_{i=1}^{n} p_i \log_2(1/p_i)$

Basic Definition

- Uncertainty measure of random variable X
- Shannon's entropy measures bits needed to encode random variable
- Event with probability p carries $\log_2(\frac{1}{p})$ bits of information
- Rare events = more information when they occur

Properties

[BOARD] Add: $0 \leq H(X) \leq \log_2(n)$

- Minimum $H(X) = 0$ when one value has probability 1 (deterministic)
- Maximum $H(X) = \log_2(n)$ when uniformly distributed
- Uniform distribution has maximum uncertainty

Conditional Entropy

[BOARD] Write: $H(X, Y) = H(Y) + H(X|Y)$

- Joint entropy decomposes into parts
- Key inequality: $H(X|Y) \leq H(X)$
- Equality only when X and Y are independent

- Observing Y can only reduce uncertainty about X

Connection to Cryptography

[BOARD] Write: $H(K|C) = H(K) + H(P) - H(C)$

- This formula applies to deterministic encryption systems
- Think about what $H(K|C)$ means:
  - **Remaining uncertainty about key** after seeing ciphertext
  - How many bits of key info are still hidden?

[BOARD] Write: Perfect secrecy: $H(K|C) = H(K)$

- For perfect secrecy: ciphertext reveals nothing about key
- So we need $H(K|C) = H(K)$ (uncertainty unchanged)
- Substitute into our formula:
  - $H(K) = H(K) + H(P) - H(C)$
  - Simplify: $H(C) = H(P)$
  - But also need $H(K) \geq H(C)$ (to encrypt all ciphertexts)
  - **Therefore:** $H(K) \geq H(P)$

Key Insight:

- **Keyspace entropy must match or exceed plaintext entropy**
- In simple terms: need at least as many keys as plaintexts
- This is why perfect security is expensive!

[TRANSITION] "This entropy constraint leads us to the only practical perfectly secure system..."

One-Time Pad (6:30-9:00)

[BOARD] Write: "OTP = Shift cipher with $n = 2$, each key used once"

- Essentially binary shift cipher ($n = 2$)
- Addition/subtraction mod 2 = XOR operation
- Every key used exactly once—critical requirement

Why It Works

- Achieves perfect security when used correctly
- Key as long as message satisfies $H(K) \geq H(P)$
- Encryption and decryption are identical operations (XOR both ways)

Practical Considerations

- Most practical perfectly secure cryptosystem

- Easy to implement in hardware/software
- **Major limitation**: key must be at least as long as message
- Key distribution becomes the real problem

[TRANSITION] "In practice, we often can't afford perfect security. So how much ciphertext does an attacker need to break a weaker system?"

Unicity Distance (9:00-12:30)

[BOARD] Write: $n_0 \approx \frac{\log_2(|K|)}{R_L \cdot \log(|P|)}$ (~ H(K)/redundancy of the language per character)

Definition

- Minimum ciphertext length where brute force leaves only one meaningful decryption
- After trying all keys, only the correct one produces valid language

The Spurious Key Problem

- Short ciphertext: many keys produce "valid-looking" plaintext
- These are **spurious keys**—incorrect keys that pass by chance
- As ciphertext grows longer, spurious keys disappear
- Unicity distance: where expected number of spurious keys drops to zero

Language Redundancy

[BOARD] Write: $R_L = 0.75$

- Natural language is highly redundant—carries less info than maximum possible
- English: $R_L \approx 0.75$ (75% redundant!)
- This redundancy helps us understand partial speech
- Also helps cryptanalysts distinguish real text from gibberish

What this means

- If you compress before encrypting making $R_L \approx 0$, you get $n_0 \approx \infty$.
- This sounds good, but in reality the Adv can probably do more than ciphertext-only attacks.
- If the adversary gets a known plaintext attack, he may be able to compute the key easily, despite the fact that the redundancy is (almost) 0
- Also, compression algorithms are usually not perfect.
- still: minimize the redundancy if possible:
    - compress before encryption

- use error correction codes after encryption

Total Time: ~14 minutes + 1 min buffer for questions/transitions