Purdue CS555 Cryptography Lecture 2

August 28, 2025

Two Types of Leakage

Intermediate Secrecy (Cryptography)

- Goal: Protect messages in transit between Alice and Bob
- Example: Alice sends encrypted message, Eve intercepts ciphertext
- Key idea: Key holders have a secret "trapdoor" that others don't
- Can achieve "free lunch" - perfect utility AND perfect privacy
- Main research problems:
  - Find weaker assumptions (reduce to fundamental hard problems)
  - Improve computational efficiency
- Target: Perfect indistinguishability of ciphertexts

Output Secrecy (Information Theory)

- Goal: Release information publicly while protecting privacy
- NO free lunch - always tradeoff between utility and privacy
- Main research problem: Find optimal utility-privacy tradeoff
- Must accept non-zero posterior advantage for adversaries

Symmetric Key Encryption

Setup

- Alice and Bob meet beforehand to agree on secret key k
- Both use same key for encryption and decryption
- Eve (eavesdropper) tries to learn message from ciphertext

Three Components

A symmetric encryption scheme consists of:

1. **Key Generation Gen**(θ): Outputs key k
2. **Encryption Enc**(k, m): Takes key k and message m, outputs ciphertext c
3. **Decryption Dec**(k, c): Takes key k and ciphertext c, recovers message m

All three algorithms can be probabilistic (use randomness).

One-Time Pad

Construction

- **Key Generation:** $k \leftarrow \{0,1\}^\ell$ (random ℓ-bit string)
- **Encryption:** $c = m \oplus k$ (XOR message with key)
- **Decryption:** $m = c \oplus k$ (XOR ciphertext with key)

Why One-Time Pad is Perfectly Secret

Theorem: One-time pad achieves perfect secrecy.
Proof idea: For any message m and ciphertext c of length ℓ:

$$\Pr_{k \sim \{0,1\}^\ell}[\text{Enc}(k,m) = c] = \frac{1}{2^\ell}$$

- There's exactly one key $k = m \oplus c$ that maps m to c
- Since k is chosen uniformly at random from 2^ℓ possibilities
- Probability is $1/2^\ell$ for ANY message m
- Therefore: $\Pr[\text{Enc}(k,m) = c] = \Pr[\text{Enc}(k,m') = c]$ for any m, m'
- Ciphertext reveals nothing about which message was encrypted

Why Key Reuse Breaks Security

Two-Time Pad Attack

Theorem: Reusing one-time pad key does not achieve perfect indistinguishability.
Proof: We show probabilities differ for different message pairs.
Choose:

- $m_0 = m_1 = m$
- $m'_0 \neq m'_1$
- $c_0 = c_1 = c$ (observing same ciphertext)

Then:

$$\Pr[\text{Enc}(k,m_0) = c \text{ and } \text{Enc}(k,m_1) = c] = \frac{1}{2^\ell}$$
$$\Pr[\text{Enc}(k,m'_0) = c \text{ and } \text{Enc}(k,m'_1) = c] = 0$$

Why second probability is 0:

- If $m'_0 \oplus k = c$, then $k = m'_0 \oplus c$
- Then $m'_1 \oplus k = m'_1 \oplus m'_0 \oplus c$
- Since $m'_0 \neq m'_1$, this cannot equal c
- So both encryptions can't produce same ciphertext

Since probabilities differ, perfect indistinguishability fails.

Fundamental Limitation of Perfect Secrecy

A Shorter Key?

Theorem: For any perfectly secure encryption scheme with key space $\mathcal{K}$ and message space $\mathcal{M}$:

$$|\mathcal{K}| \geq |\mathcal{M}|$$

Note: Key must be at least as long as message.

Probabilistic Polynomial Time (PPT)

Definition: A PPT algorithm satisfies:

- Runs in time polynomial in input length: $\text{poly}(\ell)$
- Can use randomness during execution
- Time bound holds for all possible random coin tosses

Intuition: PPT captures "realistic" adversaries with bounded computational resources. Assumes adversaries can't run for exponential time like 2^{100} years.

Negligible Functions

Definition: Function $\mu : \mathbb{N} \to \mathbb{R}$ is negligible if for every polynomial $p(\cdot)$, there exists n_0 such that for all $n \geq n_0$:

$$\mu(n) < \frac{1}{p(n)}$$

Notation: $\mu(n) = o(1/p(n))$ for any polynomial p.
Intuition: Negligible means "smaller than any inverse polynomial" - vanishingly small.

Computational Indistinguishability

COMPUTATIONAL INDISTINGUISHABILITY

$\mu(n)$ is necessary because if $\mu(n)$, even when we relax it to P.P.T, still will be complete secrecy

INITIALIZATION VECTOR & ONE-TIME PADDING will often be used in homeworks

Initialization Vector

PROBLEM: deterministic encryption (same input always gives the same ouput)
SOLUTION: Initialization Vector (IV)

IV is a random piece of data that we use to randomize each message BEFORE encrypting it.

In IV encryption, the IV acts as a **one time pad** for each message.