Speaker's Notes: Symmetric Cryptography (15-Minute Oral Exam)

1. Introduction & Cryptosystem Definition (0-2:00)

[BOARD] Write: (G, E, D) and the three sets: $\mathscr{K}, \mathscr{P}, \mathscr{C}$

- Start: "Symmetric cryptography is built on cryptosystems"
- **Definition**: Triple (G, E, D) — three algorithms
- Emphasize: **Definition says nothing about security** (just syntax)
- **Correctness requirement**: For any K from G, any $x \in \mathscr{P}$:

$$x = D_K(E_K(x))$$

- Symmetric = same key for encryption and decryption

Three Algorithms:

G (Key Generation):

- Probabilistic, no input
- Outputs $K \in \mathscr{K}$ (usually uniform)

E (Encryption):

- Input: K and $x \in \mathscr{P}$
- Output: $E_K(x) \in \mathscr{C}$
- **May be probabilistic** — same (K, x) can give different ciphertexts

D (Decryption):

- Input: K and $y \in \mathscr{C}$
- Output: $D_K(y) \in \mathscr{P}$
- Usually deterministic

[TRANSITION] "This is the abstract definition. In practice, we often use *block ciphers*—a specific type of cryptosystem with fixed-length blocks."

---

2. Block Ciphers & Feistel Structure (2:00-4:30)

[BOARD] Write: **Block cipher: fixed n-bit input $\rightarrow$ fixed n-bit output**

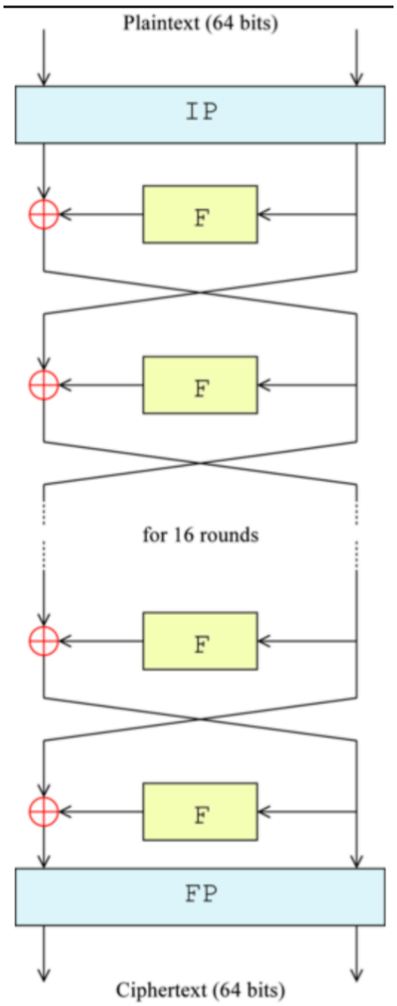
Block Cipher:

- Key = bitstring of fixed length (uniform)
- Fixed-length input $\to$ same-length output
- Example: DES (64-bit blocks), AES (128-bit blocks)

[TRANSITION] "A powerful design pattern for block ciphers is the *Feistel cipher*."

Feistel Cipher Structure:

[BOARD] Draw: Split block into $L_0 || R_0$, show one round with XOR



- Named after Horst Feistel (IBM, DES team)
- Structure: n rounds of substitution/permutation
- **Key Schedule**: $K \to K_1, K_2, \ldots, K_n$ (round keys)

Round computation (for $i = 1 \ldots n-1$):

$$L_i = R_{i-1}, \quad R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

Last round (different):

$$R_n = R_{n-1}, \quad L_n = L_{n-1} \oplus f(R_{n-1}, K_n)$$

Output: $C = (L_n, R_n)$

Key insight:

- Decryption = **reverse the rounds** (use keys $K_n, \ldots, K_1$)
- Works for **any** function f (doesn't need to be invertible!)
- Design f for maximum security without other constraints

[TRANSITION] "The most famous Feistel cipher is DES—let me briefly describe it."

---

3. DES Overview (4:30-6:00)

[BOARD] Write: **DES: 64-bit blocks, 56-bit key, 16 Feistel rounds**

DES (Data Encryption Standard):

- 1970s standard, deprecated 2001 (replaced by AES)
- 64-bit plaintext, 56-bit key $\rightarrow$ 64-bit ciphertext
- **Structure**: 16-round Feistel network

High-level process: 0. Key Scheduler generates 16 subkeys from K_0

1. Initial permutation, split into L_0, R_0 (32 bits each)
2. **Round function**: $f(R, K_i) = P(S(K_i \oplus E(R)))$
    - $E(R)$: expand 32 bits $\rightarrow$ 48 bits
    - XOR with 48-bit subkey
    - $S(\ldots)$: S-boxes (nonlinear substitution, 48 $\rightarrow$ 32 bits, LUT)
    - $P(\ldots)$: permutation
3. After 16 rounds: swap halves, final permutation

Decryption: Same network, **reverse subkey order**

Weakness: Still unbroken - but 56-bit key too small by modern standards

[TRANSITION] "But what does it mean that DES is unbroken? What is 'secure'?"

---

4. PRF Security (6:00-9:00)

[BOARD] Write: **PRF Security: Can adversary distinguish f_K from random function?**

Intuition:

- Good cryptosystem: encryption looks **random**
- Adversary can't tell $f_K(x)$ from $R(x)$ (truly random function)

Setup: Function f_K where $K \in \{0,1\}^k$, $f_K : \{0,1\}^n \rightarrow \{0,1\}^m$

The Game (Two Worlds):

[BOARD] Draw two boxes: O_{Real} and O_{Ideal}

Ideal World (O_{Ideal}):

- Choose random mapping $R : \{0,1\}^n \to \{0,1\}^m$ (uniform over all mappings)
- On query x: return $R(x)$

Real World (O_{Real}):

- Choose $K \to \{0,1\}^k$ (fixed for entire game)
- On query x: return $f_K(x)$

Adversary A:

- Makes queries, gets responses
- Outputs 1 bit: guess which world

[BOARD] Write advantage formula:

$$\text{Adv}_A(O_{\text{Real}}, O_{\text{Ideal}}) = |p(A,0) - p(A,1)|$$

where $p(A,b)$ = probability A outputs b

Definition: f_K is (t, q, ε)**-PRF-secure** if:

- Any adversary running time $\leq t$
- Making $\leq q$ oracle queries
- Has advantage $\leq \varepsilon$

If Adv cannot even solve this, it won't "break" our system for sure.

[TRANSITION] "PRF security is nice, but has a problem: deterministic encryption leaks information! If I encrypt the same message twice, I get the same ciphertext. We need something stronger: CPA security."

---

5. CPA Security (9:00-11:30)

[BOARD] Write: **CPA** (Chosen Plaintext Attack)

The Problem with Deterministic Encryption:

- Encrypt m today $\to$ ciphertext c
- Encrypt m tomorrow $\to$ same c
- Adversary learns: "same message encrypted twice"
- Too much information leaked!

CPA Security Intuition:

- Adversary **chooses** plaintexts x to encrypt

- Can't distinguish $E_K(x)$ from $E_K(r)$ where r is random (same length)

The CPA Game (Two Worlds):

[BOARD] Draw: O_{Real} vs O_{Ideal}

Real World (O_{Real}):

- Fix $K \leftarrow G$
- On query x: return $E_K(x)$

Ideal World (O_{Ideal}):

- Fix $K \leftarrow G$
- On query x: choose random r (same length as x), return $E_K(r)$

Example why DES fails CPA:

- Adversary submits x multiple times
- If same ciphertext $\rightarrow$ Real world
- If different ciphertexts $\rightarrow$ Ideal world
- Deterministic encryption breaks CPA!

[BOARD] Write definition: (G, E, D) is (t, q, μ, ε)-CPA-secure

For any adversary:

- Time $\leq t$
- Queries $\leq q$
- Total plaintext bits $\leq \mu$

Then: $\text{Adv}_A(O_{\text{Real}}, O_{\text{Ideal}}) \leq \varepsilon$

[TRANSITION] "So how do we build CPA-secure encryption from PRF-secure block ciphers? Answer: *modes of operation* like CTR mode."

---

6. CTR Mode (11:30-14:00)

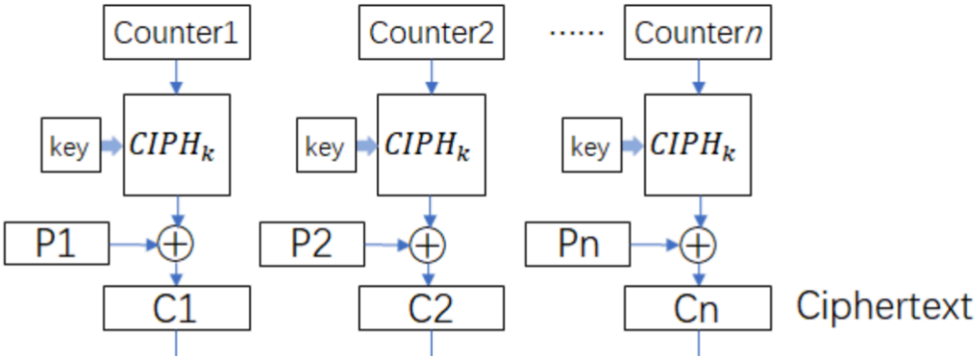
[BOARD] Write: **CTR = Counter Mode** and draw the encryption scheme

The Idea:

- Take PRF-secure block cipher $\rightarrow$ make it CPA-secure
- Handles **arbitrary length** messages (not just fixed blocks)

CTR Encryption:

[BOARD] Draw: $IV \rightarrow [E_K] \rightarrow \oplus m_1$, $(IV + 1) \rightarrow [E_K] \rightarrow \oplus m_2$, etc.



1. Generate random IV (initialization vector / nonce), length n (block size)
2. Split message into n-bit blocks: $m_1, m_2, \ldots, m_\ell$
3. For block i:
    - Compute $E_K(IV + i - 1)$
    - XOR with m_i
4. **Ciphertext**:

$$C = [IV, m_1 \oplus E_K(IV), m_2 \oplus E_K(IV+1), \ldots]$$

Decryption:

- Parse IV from ciphertext
- Recompute $E_K(IV), E_K(IV+1), \ldots$
- XOR with ciphertext blocks
- Trivial!

Why CPA-secure?

- Random IV each time $\to$ different ciphertext even for same message
- Encrypting same x twice: different IV $\to$ completely different output

[BOARD] Write theorem:

Theorem: If E'_K is (t', q', ε')-PRF-secure, then CTR mode is (t, q, μ, ε)-CPA-secure where:\Z

$$t \leq t', \quad \frac{\mu}{n} \leq q', \quad \varepsilon = \varepsilon' + \left(\frac{\mu}{n}\right)^2 \cdot \frac{1}{2^n}$$

Conclusion (14:00-14:30)

[BOARD] Point back to (G, E, D) and the security notions

Summary:

- Started with abstract cryptosystem definition
- Concrete example: DES (Feistel structure)
- Security notions: PRF (pseudorandomness) and CPA (chosen plaintext)
- Practical construction: CTR mode bridges PRF $\to$ CPA security

- Bonus: gives us stream ciphers