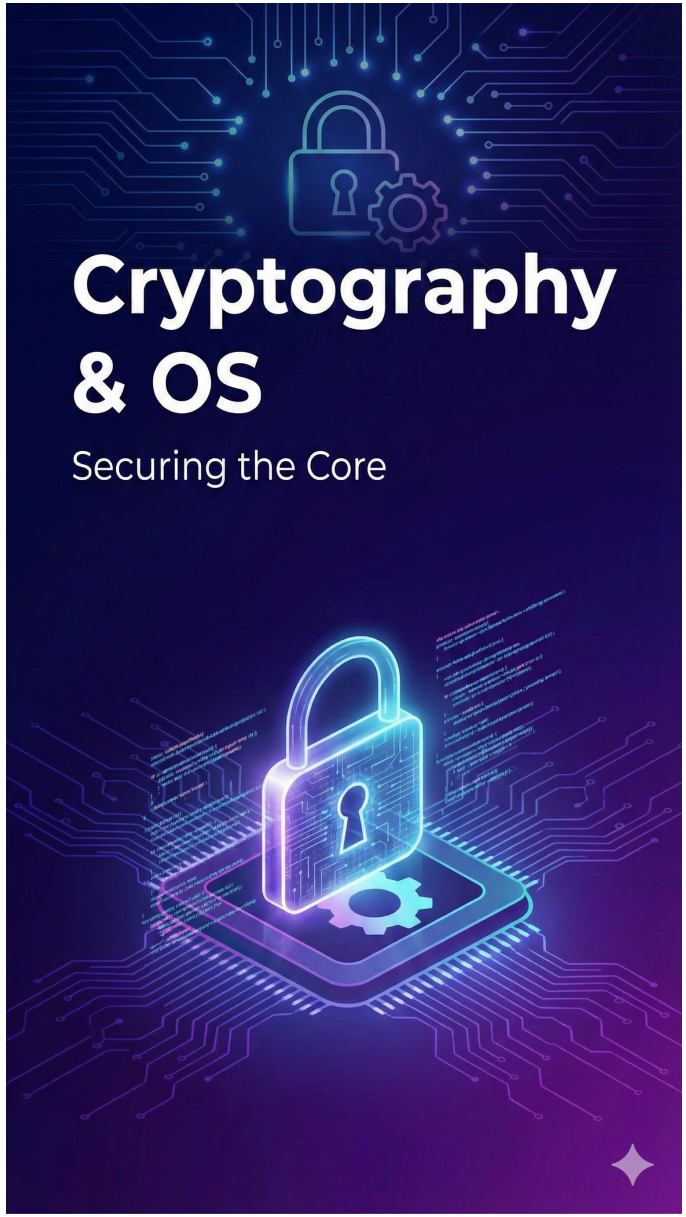


Image Source: Gemini

CSL 301

OPERATING SYSTEMS

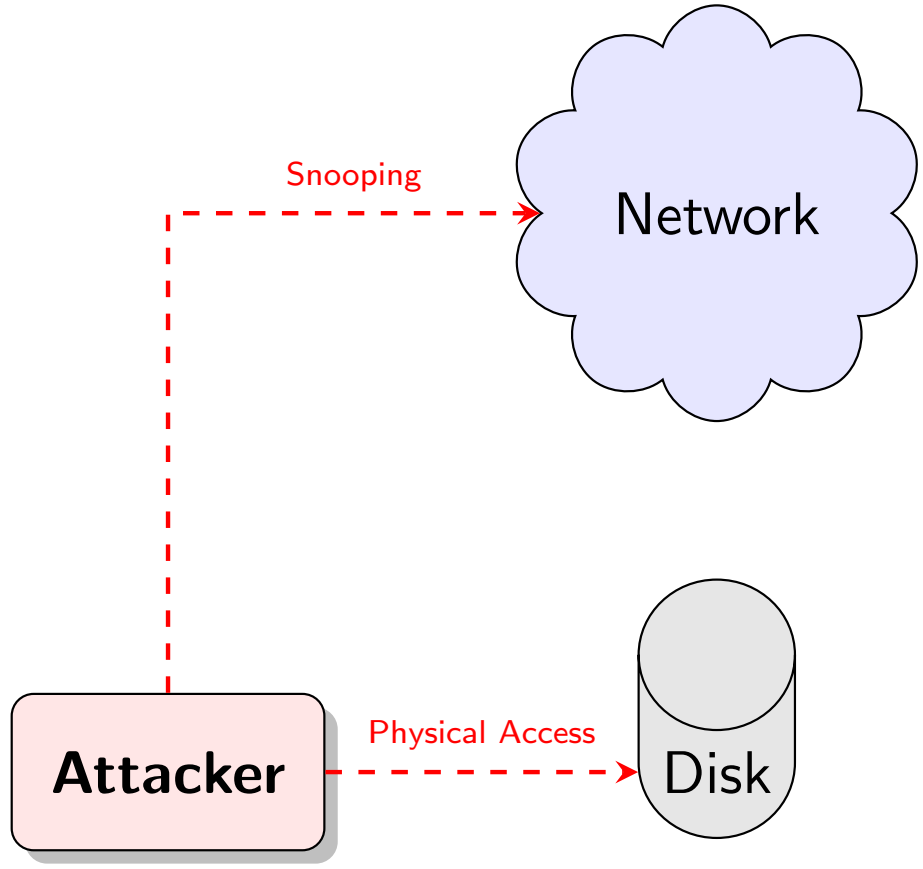
Lecture 30

Cryptography & OS

Instructor
Dr. Dhiman Saha

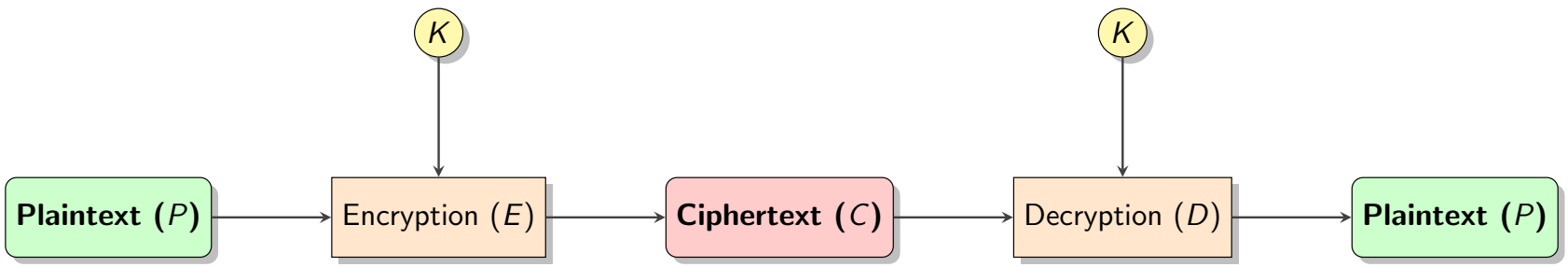
Why Cryptography in OS?

- **The OS is powerful** but has limits.
- It controls hardware interfaces but not:
  - Other machines on the network.
  - Physical access to storage devices (if bypassed).
  - Data in transit.
- **Goal**: Protect information *outside* the OS's direct control.
- **Solution**: Transform data so it is unusable to unauthorized parties.



The Core Concept

- **Plaintext (P)**: The original data.
- **Ciphertext (C)**: The scrambled data.
- **Key (K)**: Secret information used for transformation.
- **Encryption (E)**: $C = E(P, K)$
- **Decryption (D)**: $P = D(C, K)$



Kerckhoffs's Principle

Important Rule

The security of the cryptosystem must NOT depend on keeping the algorithm secret. It must depend **only on the secrecy of the key**.

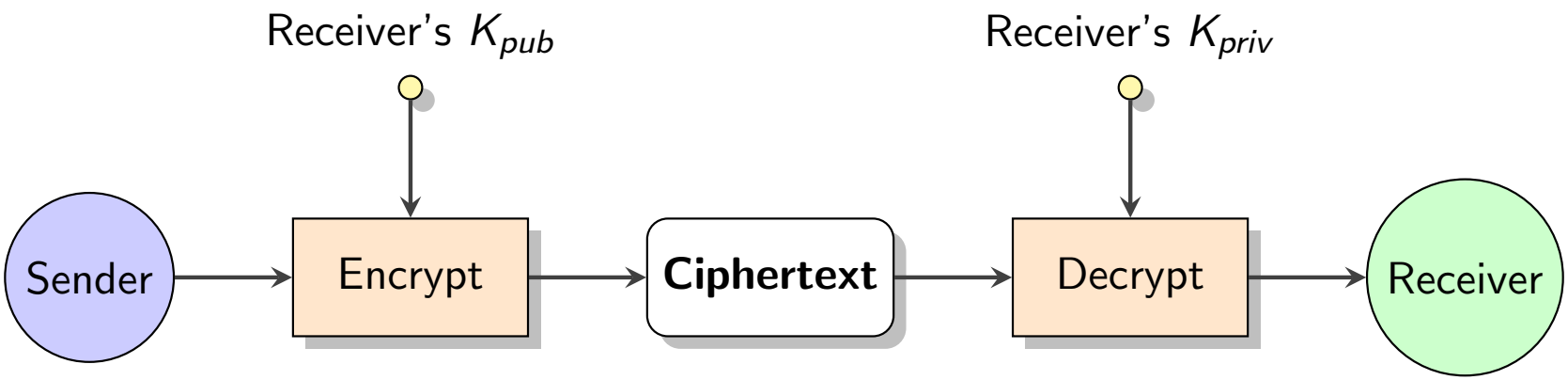
- **Don't roll your own crypto!**
- Use standard, vetted algorithms (e.g., AES).
- Proprietary algorithms are often flawed and easily broken by experts.

Symmetric Cryptography

- Same key used for Encryption and Decryption.
- **Pros**: Fast, efficient (good for bulk data).
- **Cons**: Key distribution is difficult (How do we share K securely?).
- **Example**: AES (Advanced Encryption Standard).

Public Key Cryptography (Asymmetric)

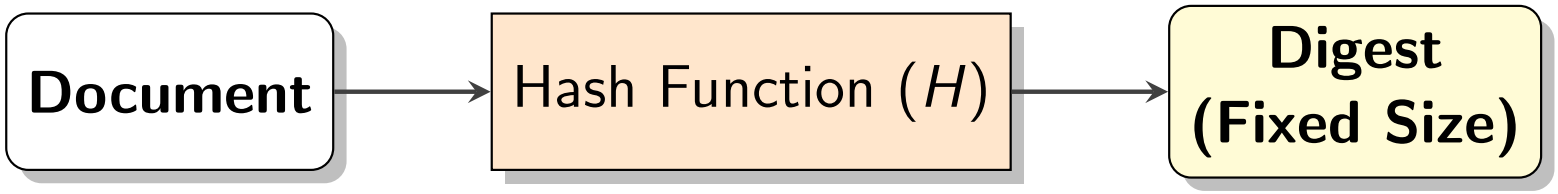
- Two keys: **Public Key (K_{pub})** and **Private Key (K_{priv})**.
- **Secrecy**: Encrypt with K_{pub}, Decrypt with K_{priv}.
- **Authentication**: Encrypt with K_{priv}, Decrypt with K_{pub}.



Scenario: Secrecy

Cryptographic Hashes

- One-way function: $S = H(P)$.
- **Integrity**: Detects tampering.
- **Properties**:
  1. Collision resistance (Hard to find P_1, P_2 such that $H(P_1) = H(P_2)$).
  2. Avalanche effect (Small change in $P \rightarrow$ Huge change in S).
  3. Irreversible.



e.g., SHA-256
a591a6d40bf42040...

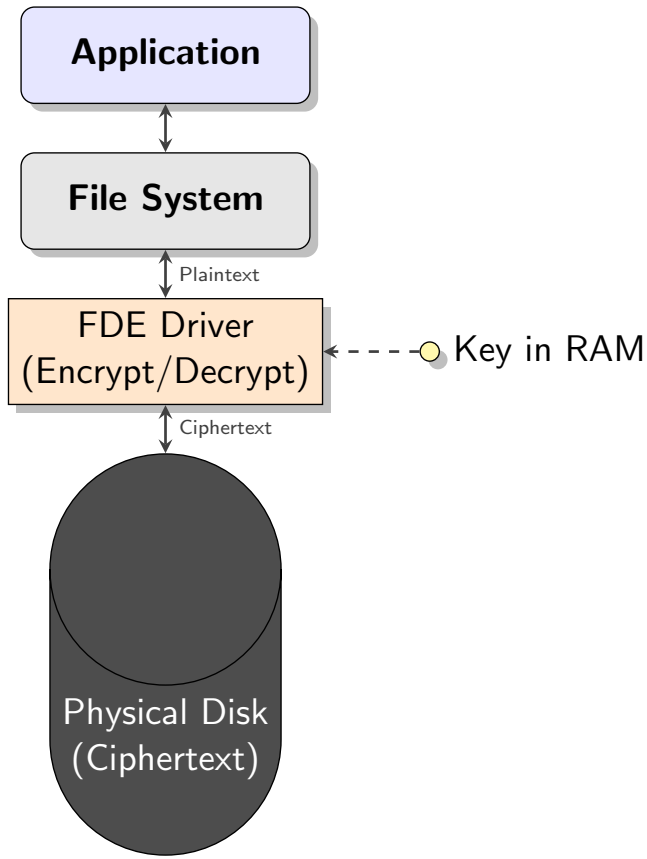
The Hardest Part: Key Management

"The cryptography's benefit relies entirely on the secrecy of the key."

- **Key Selection**: Must be random (high entropy).
- **Bad Practice**: Using time of day, process ID, or short passwords as seeds.
- **Storage**: Don't hardcode keys in software/firmware!
- **Example Failure**: Embedded devices sharing the same private key.

Full Disk Encryption (FDE)

- Protects data when the device is powered off or stolen.
- Transparent to applications.
- **Boot Process**: User enters passphrase → Key derived → OS boots.



Summary

- **Cryptography** allows us to protect data even when we lose physical control.
- **Primitives**: Symmetric (AES), Asymmetric (RSA/ECC), Hashes (SHA).
- **Implementation**:
  - Never invent your own cipher.
  - Manage keys securely (Entropy, Storage).
- **OS Role**:
- Provides entropy (/dev/random).
- Implements Full Disk Encryption.
- Must be trusted to handle keys in memory.