ECE568 笔记汇总

- Cryptography - Block Ciphers
- Cryptography - Ciphers
- Cryptography - Hashes, MACs, and Digital-Signitures
- Cryptography - Public-Key Cryptography
- Cryptography - Stream Ciphers

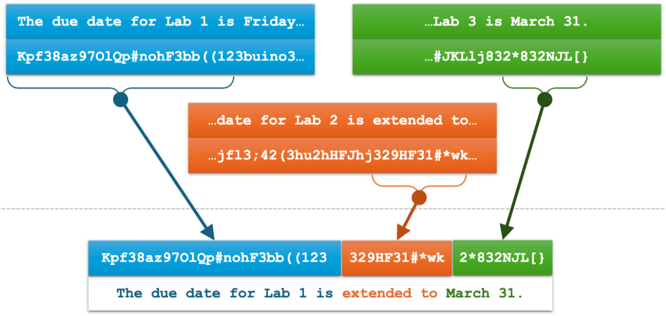
Table of Contents

Block Cipher Design

- *Communication Theory of Secrecy Systems* (Claude Shannon, 1949) foundation for modern cryptography
- 2 goals for good crypto-system
  - **Confusion** - obscuring of the relationship between PT and CT
    - make statistical analysis difficult, even if attacker has large PT-CT pairs
    - encoding should be **non-linear** to avoid **extension attacks**
    - Each character of the CT should depend on the **entire key**
  - **Diffusion** - spreading the influence of individual PT char over much of the CT
    - each output bit should result from a combination of **many input bits**
      - flipping 1 bit of key/PT should change >50% of output bit
      - any repetitive patterns in PT are spread over entire CT, hiding statistical info

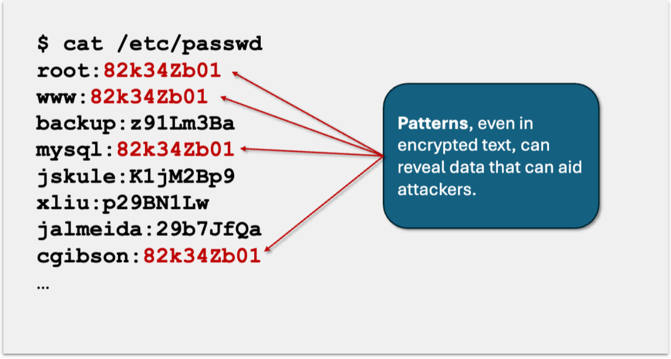
Extension Attack

Insert CT in the middle of a CT to alter meaning



The due date for Lab 1 is Friday…
Kpf38az9701Qp#nohF3bb((123buino3…
…Lab 3 is March 31.
…#JKL1j832*832NJL[}
…date for Lab 2 is extended to…
…jf13;42(3hu2hHFJhj329HF31#*wk…
Kpf38az9701Qp#nohF3bb((123
329HF31#*wk
2*832NJL[}
The due date for Lab 1 is extended to March 31.

Pattern-Hiding



```
$ cat /etc/passwd
root:82k34Zb01
www:82k34Zb01
backup:z91Lm3Ba
mysql:82k34Zb01
jskule:K1jM2Bp9
xliu:p29BN1Lw
jalmeida:29b7JfQa
cgibson:82k34Zb01
…
```

Patterns, even in encrypted text, can reveal data that can aid attackers.

Design

- 2 simple weak ciphers are combined to design secure block ciphers
  - **substitution cipher** replaces c in PT with c from same L, with 1-to-1 mapping (confusion)
  - **permutation cipher** transposes the PT (diffusion)
- **iterated block cipher** repeatedly applies these 2 ciphers in different combinations

Substitution-Permutation Network (SPN)

- combine several rounds of substitution and permutation
- **round keys** - derived from primary key, typically applied by *XORing key with the output of each round of the encryption*

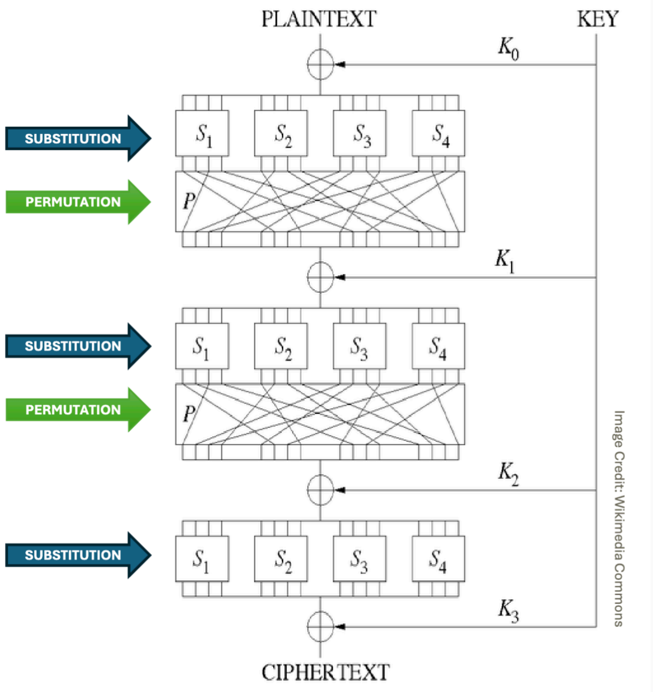


Image Credit: Wikimedia Commons

DES, AES

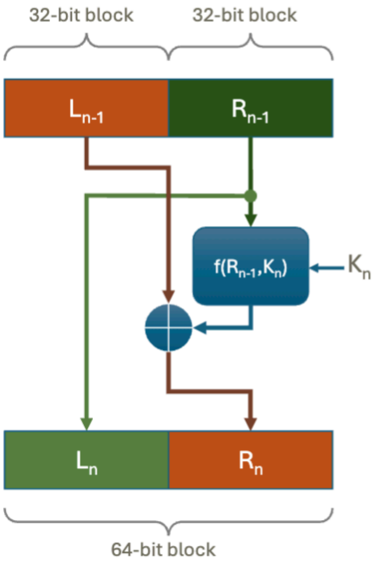
- Certified by US National Institute of Standards and Technology (NIST), 2 common iterated block ciphers
  - **DES** Data Encryption Standard - 56 bit key, a block length of 64 bits
  - **AES** Advanced Encryption Standard -

DES Timeline

- NIST wants to standardize encryption with a reliably strong, well-studied cipher
  - IBM introduced a one based on Lucifer Cipher, which won the DES competition in 1976
  - Pronounced secure by NSA, invoked much distrust

DES Architecture

- **Feistel Network**, consists of **16** rounds
- In each round, the input is split into **L** and **R**
- Halves are swapped, *substitution function* modifies half the input bits
  - Each round uses a **subkey** K_n derived from the master encryption key

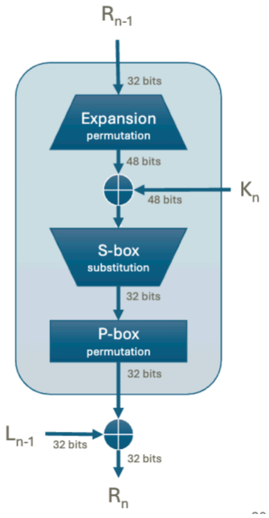


DES Subkey Generation

- 56 bit encryption key undergoes a key schedule to create 16 **sub-keys** K_n
  - Split into 2 28-bit halves
  - Each half is shifted left by 1 or 2 bits (depending on the round)
  - 24-bits are selected from each 28-bit halves to make 48-bit sub-key

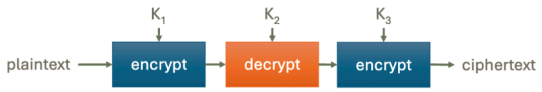
DES Substitution

- Each $f(R_{n-1}, K_n)$ substitution box contains
  - Expansion permutation of 32-bit input into 48-bits
  - XOR with 48-bit K_n
  - S-box substitution that compress 48-bit into 32-bit output **non-linear element**
  - P-box permutation of 32-bit output

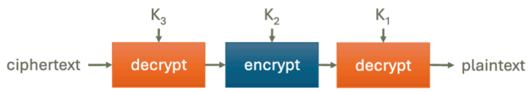


3DES

- 56-bit key has become inadequate (brute forcing 2^{56} key comb in less than a day)
  - one solution is to chain 3 DES, splitting 168-bit key into 3 56-bit parts



$$\text{ciphertext} = E_{K3}(D_{K2}(E_{K1}(\text{plaintext})))$$



$$\text{plaintext} = D_{K1}(E_{K2}(D_{K3}(\text{ciphertext})))$$

AES Timeline

- 1997, NIST ran a competition to replace DES
  - Symmetric BC
  - Increasing key-len possible
  - easily implemented in hardware/software

AES Architecture

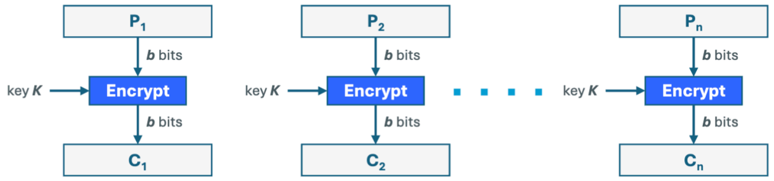
- Supports **variable key and block lengths**
  - 128-, 192-, 256 bit keys
  - 128, 192, 256 bit blocks
  - any combo is possible
  - extensions to 160, 224 possible
- Based on rounds; number of rounds is based on key length and block size (10-14)

Block Cipher Encryption Modes

- Most secure way to encrypt multiple blocks?
- Block ciphers encrypt a block at a time and use **modes** to improve security
  - **Security** - Is algorithm effective in hiding patterns in PT?
  - **Error Propagation** - Impact of CT bit flip?
  - **Error Recovery** - Error affect all blocks? Recoverable? How much data to retransmit?
  - **Performance** - What throughput is supported?

Electronic Codebook (ECB)

- Simplest mode
  - message is broken into block-sized chunks
  - padding added to last block
  - each chunk is encrypted independently



- **Error Propagation** - errors only affect one PT block, changed completely randomly
- **Error Recovery** - only retransmit affected blocks; during decryption just skip bad blocks
- **Performance** - highly parallelizable
- **Security** - POOR
  - Adversary can add, delete, or re-order blocks
  - PT block always encrypt to same CT block
  - CT block can reveal macro-structure of PT data (since divided into blocks)

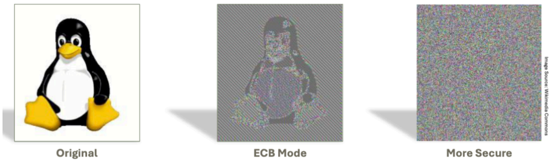
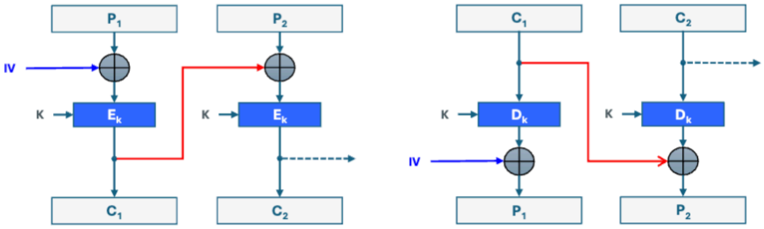


Image Source: Wikimedia Commons

Cipher Block Chaining (CBC)

- Makes every block's input dependent on the CT output of previous block
  - The **Initialization Vector (IV)** doesn't have to be secret
    - normally sent in PT along CT
    - IV should NOT be re-used

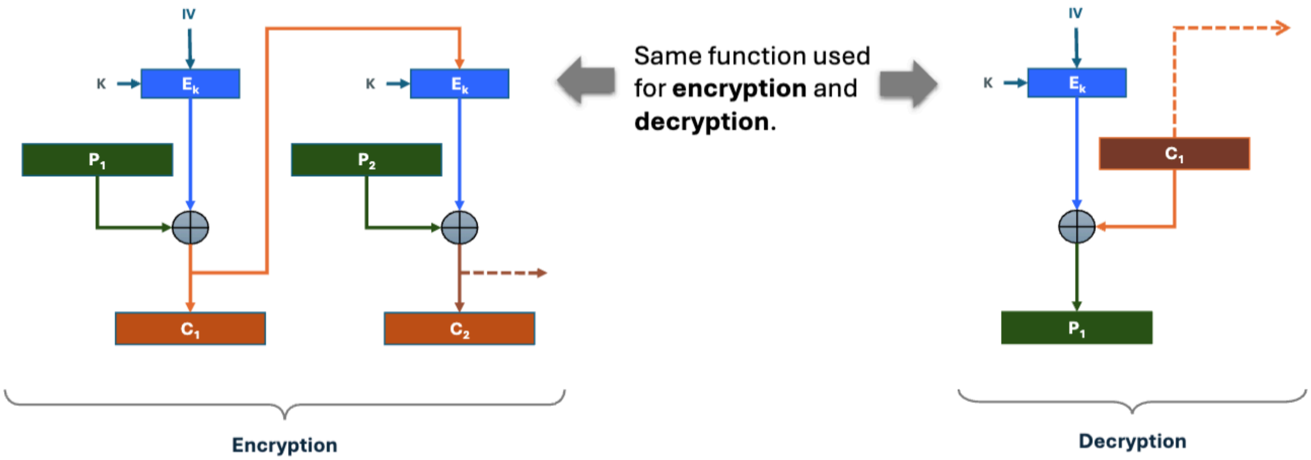


- **Security** - Any change in PT affects all later blocks, modification to CT affects at most 2 blocks during decryption
- **Performance** - encryption POOR, decryption can be parallelized
- **Error Propagation** - only affect current block and following block
- **Error Recovery** - can drop affected blocks and continue decryption

Other modes

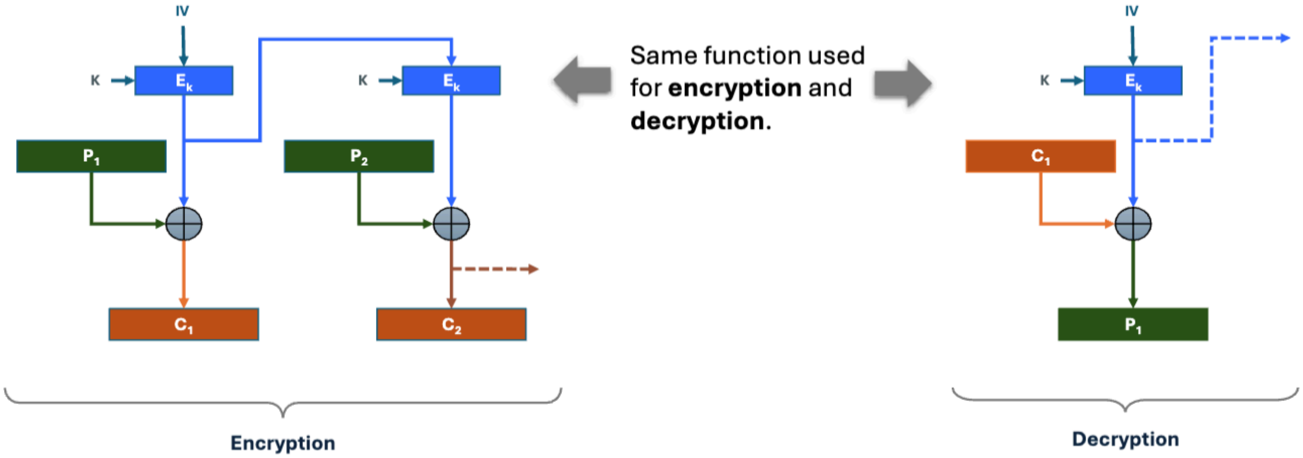
- CFP (Cipher Feedback)
- OFB (Output Feedback)
  - allow encryption and decryption in units of less than a full block at a time
  - convert block ciphers into stream ciphers
    - **security, error propagation, recovery** similar to stream ciphers
    - more effective since no padding is necessary
    - CFB - pipelining is possible
    - OFB - key stream is independent of PT; allows performing CT ops in advance and supports ECC

CFB (Cipher Feedback)



Same function used for **encryption** and **decryption**.

OFB (Output Feedback)



Same function used for **encryption** and **decryption**.