Symmetric Encryption (Block Ciphers)

Yan Huang

Credit: David Evans (UVA)

Quiz - Pros and Cons of One-Time Pad

- Pros
  - 1
  - 2
- Cons
  - 1
  - 2

Quiz - Pros and Cons of One-Time Pad

- Pros
  - Perfect security
  - Simple and efficient
- Cons
  - No Integrity
  - Key size no less than message size
  - No security when reusing the key

Problems with One-Time Pad

- Key must be as long as the plaintext
  - Impractical in most realistic scenarios
  - Still used for diplomatic and intelligence traffic
- Does not guarantee integrity
  - One-time pad only guarantees confidentiality
  - Attacker cannot recover plaintext, but can easily change it to something else
- Insecure if keys are reused
  - Attacker can obtain XOR of plaintexts

Reducing Key Size

- What to do when it is infeasible to pre-share huge random keys?
  - Change the security definition to align with some weaker but still useful threat model
  - Use special cryptographic primitives: block ciphers, stream ciphers
    - Single key can be re-used (with some restrictions)

Next lecture…

This lecture…

Ciphers

- Stream Ciphers
  - Encrypts small (bit or byte) units one at a time
- Block Ciphers
  - Operate on a single chunk of plaintext, for example, 64 bits for DES, 128 bits for AES
  - Same key is reused for each block (i.e., keys can be shorter than the messages)
- Without the key, result should look like a random permutation

Block Cipher

- Not impossible to break, just very expensive
  - **Unproven Assumption!** There is no algorithm to break the cipher more efficient than brute-force, i.e., enumerate every possible key
  - Time and cost of breaking the cipher exceed the value and/or useful lifetime of protected information

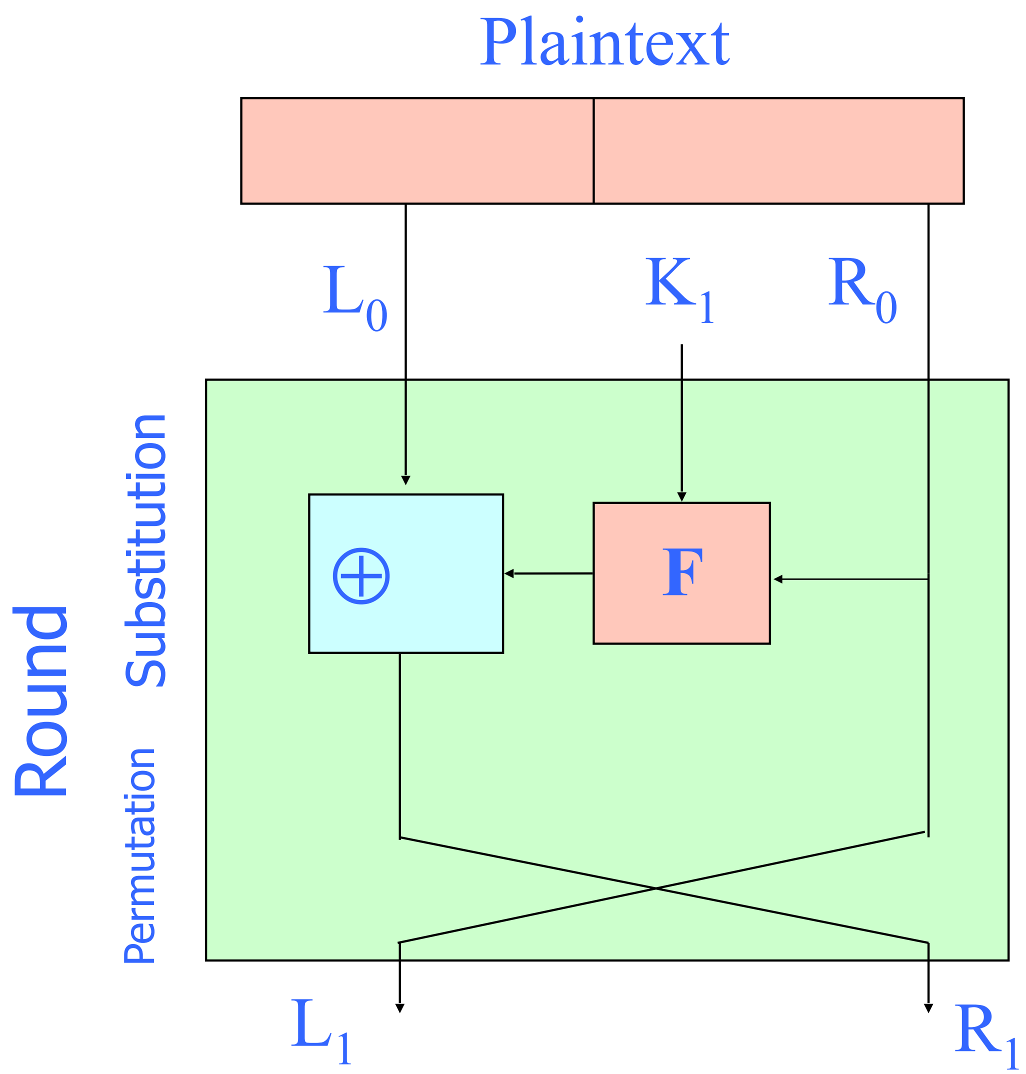
Ideal Block Cipher

- 64 bit blocks
- 2^{64} possible plaintext blocks, must have at least 2^{64} corresponding ciphertext blocks
  - There are $2^{64}!$ possible permutations
- Why not just create a random permutation?
  - Need a log (2^{64} !) bits key(>1.15 x 10^{21} bits)
  - Occupying a $7-billion disc drive
  - Need to distribute new key if compromised
- Approximate ideal random mapping using components controlled by a key

A Bit of Block Cipher History

- Playfair and variants (from 1854 until WWII)
- Feistel structure
  - "Ladder" structure: split input in half, put one half through the round and XOR with the other half
  - After 3 random rounds, ciphertext indistinguishable from a random permutation
- DES: Data Encryption Standard
  - Invented by IBM, issued as federal standard in 1977
  - 64-bit blocks, 56-bit key + 8 bits for parity
  - Very widely used (usually as 3DES) until recently
    - 3DES: DES + inverse DES + DES (with 2 or 3 different keys)

Feistel Cipher Structure



L_0 = left half of plaintext

R_0 = right half of plaintext

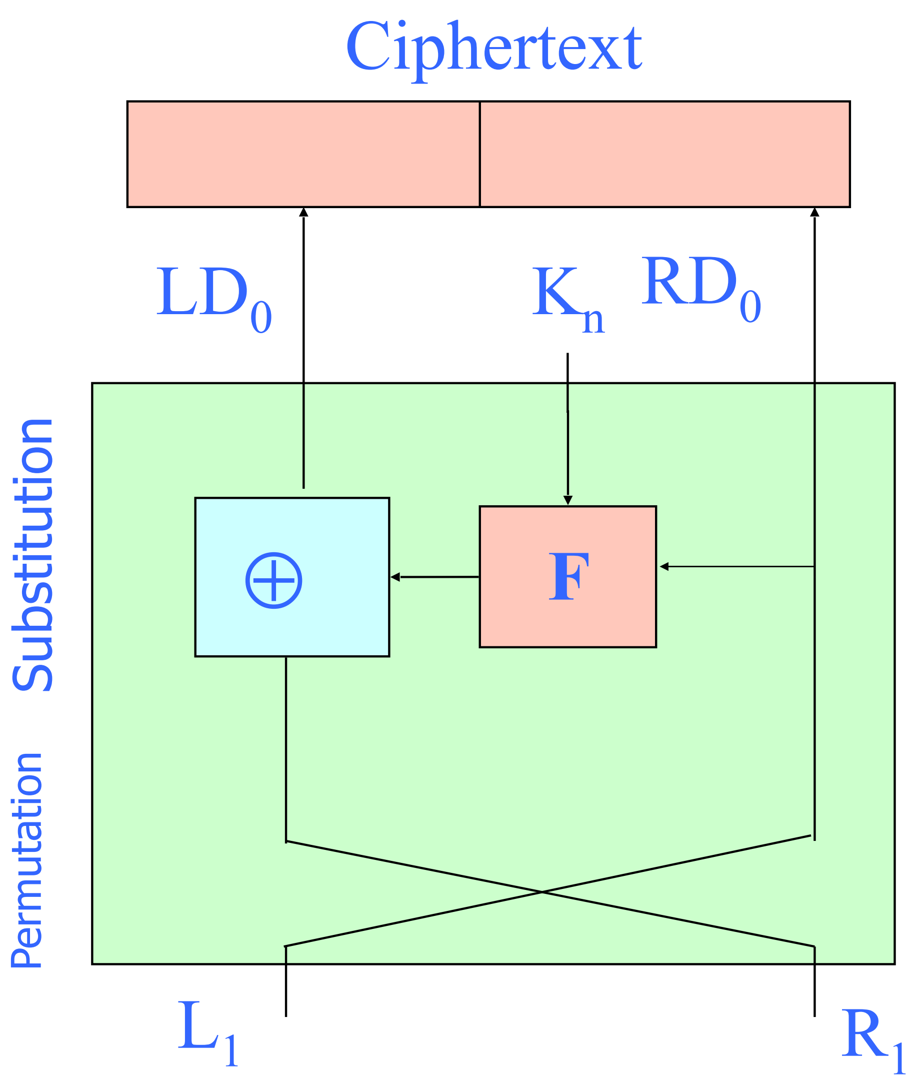
$L_i = R_{i-1}$

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

$C = L_n \| R_n$

n is number of rounds

Decryption



LD_n = left half of ciphertext

RD_n = right half of ciphertext

$$RD_i = LD_{i+1}$$

$$LD_i = LD_{i+1} \oplus F\,(LD_{i+1}, \mathbf{K}_{\mathbf{n-i+1}})$$

$$P = LD_0 \,\|\, RD_0$$

n is number of rounds

F

- What are the requirements on F?
  - For decryption to work: none!
  - For security:
    - Hide patterns in plaintext
    - Hide patterns in key
    - Coming up with a good F is hard

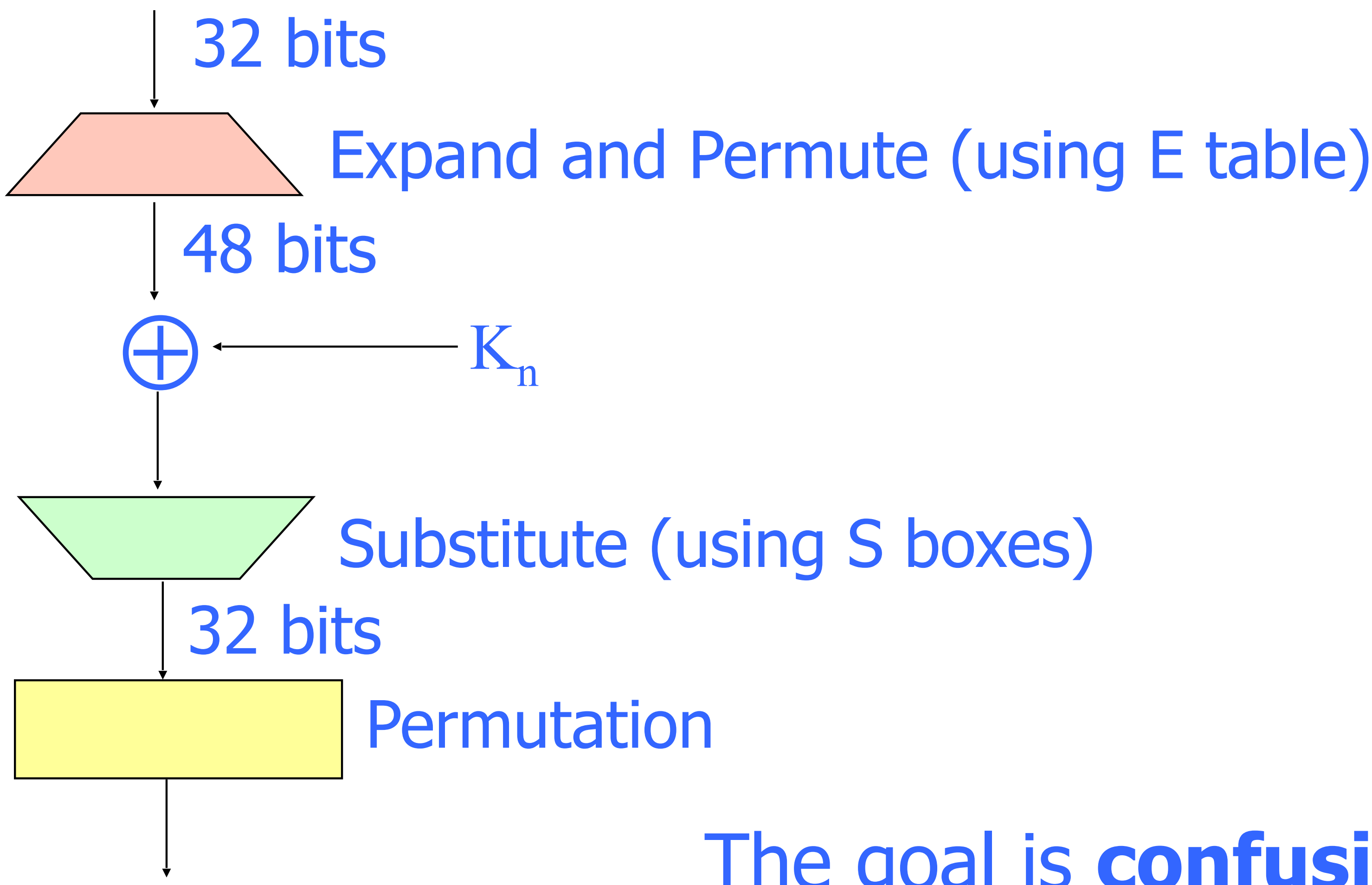
DES

- NIST (then NBS) sought standard for data security (1973)
- IBM's Lucifer only reasonable proposal
- Modified by NSA
  - Changed S-Boxes
  - Reduced key from 128 to 56 bits
- Adopted as standard in 1976

DES Algorithm

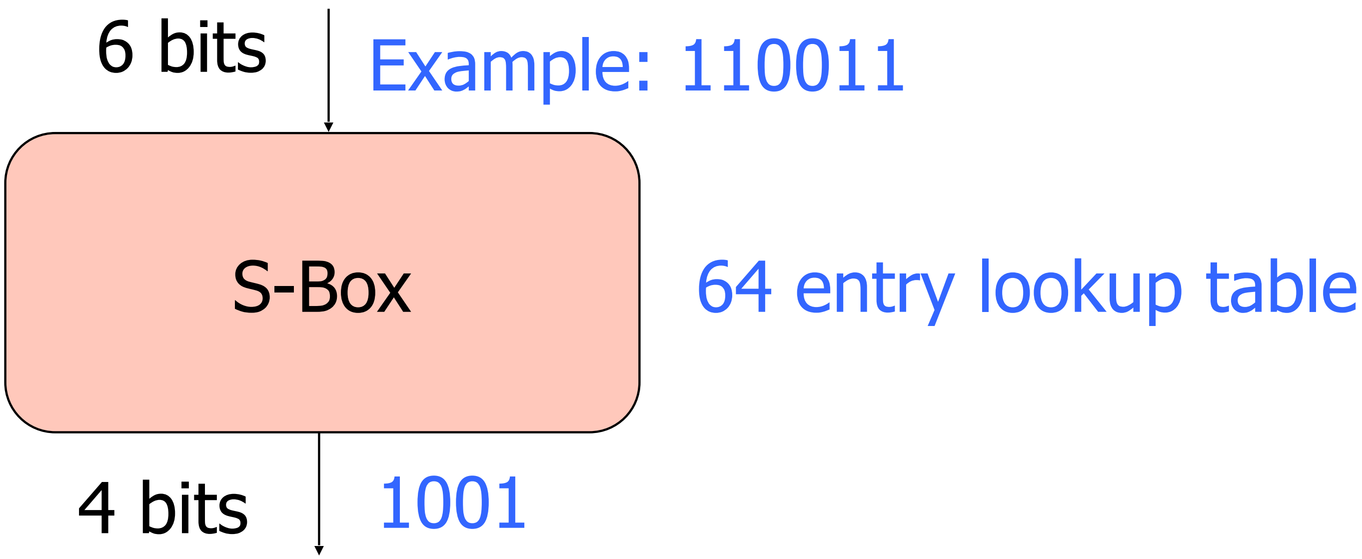
- Feistel cipher with added initial permutation
- Complex choice of F
- 16 rounds
- 56-bit key, shifts and permutations produce 48-bit subkeys for each round

DES's F



The goal is **confusion**!

S-Boxes



Critical to security

NSA changed choice of S-Boxes

Only non-linear step in DES $S(11) \neq S(01) + S(10)$

DES Avalanche

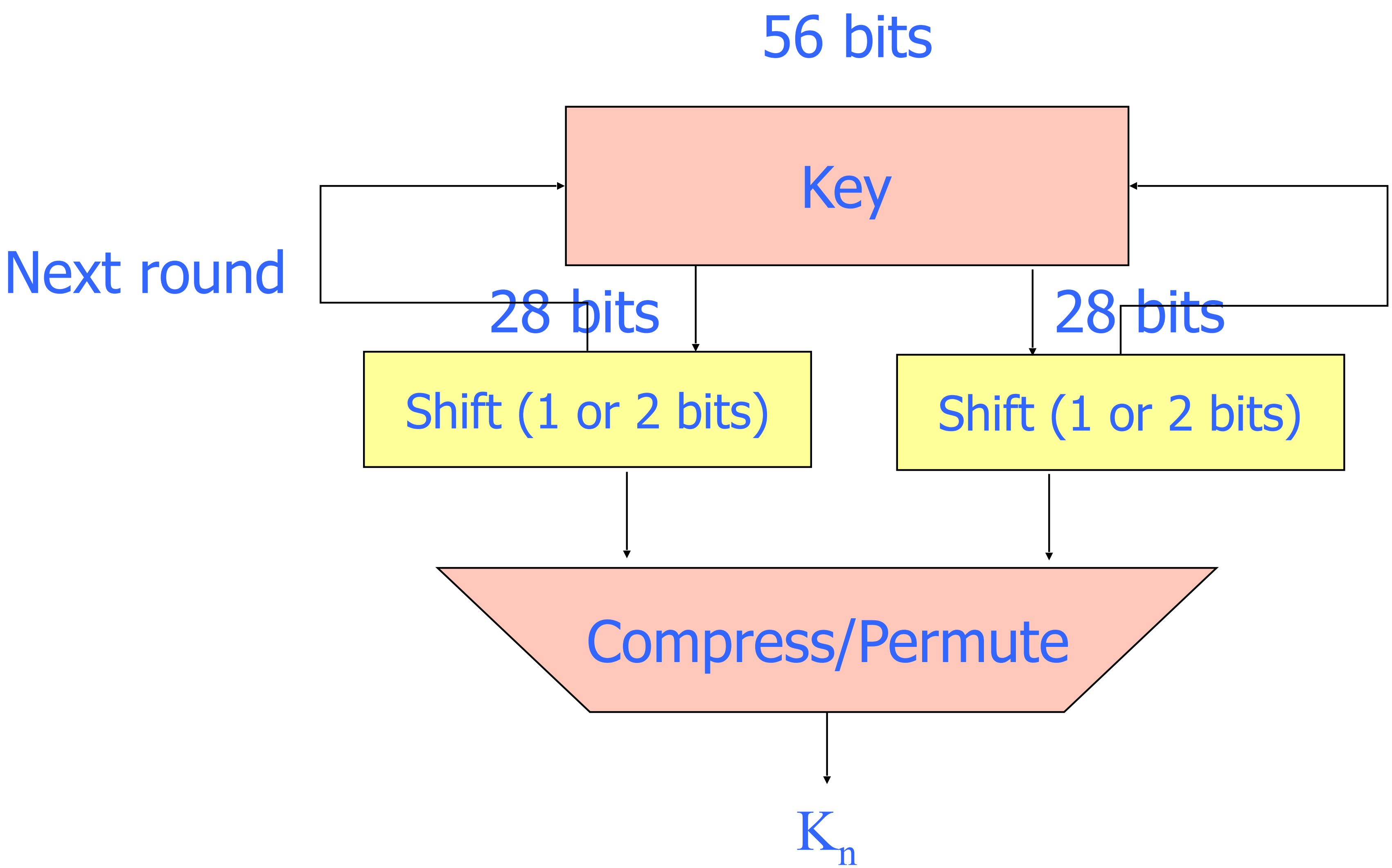
```
Input:     ...............................................................*     1
Permuted:  .......................................*........................     1
Round 1:   .......*........................................................     1
Round 2:   .*..*...*.....*........................*........................     5
Round 3:   .*..*.*.**..*.*.*.*....**.....**.*..*...*.....*.................    18
Round 4:   ..*.*****.*.*****.*.*......*.....*..*.*.**..*.*.*.*....**.....**    28
Round 5:   *...**..*.*...*.*.*.*...*.***..*..*.*****.*.*****.*.*......*....    29
Round 6:   ...*..**.....*.*..**.*.**...*..**...**..*.*...*.*.*.*...*.***..*    26
Round 7:   *****...***....**...*..*.*..*......*..**.....*.*..**.*.**...*..*
Round 8:   *.*.*.*.**.....*.*.*...**.*...*******...***....**...*..*.*..*...
Round 9:   ***.*.***...**.*.****.....**.*..*.*.*.*.**.....*.*.*...**.*...**
Round 10:  *.*..*.*.**.*..*.**.***.**.*...****.*.***...**.*.****.....**.*..
Round 11:  ..******......*..******....*....*.*..*.*.**.*..*.**.***.**.*...*
Round 12:  *..***....*...*.*.*.***...****....******......*..******....*....
Round 13:  **..*....*..******...*........*.*..***....*...*.*.*.***...****..
Round 14:  *.**.*....*.*....**.*...*..**.****..*....*..******...*........*.
Round 15:  **.*....*.*.*...*.**.*..*.*.**.**.**.*....*.*....**.*...*..**.**
Round 16:  .*..*.*..*..*.**....**..*..*..****.*....*.*.*...*.**.*..*.*.**.*
Output:    ..*..**.*.*...*....***..***.**.*...*..*..*.*.*.**.*....*.*.*.**.
```

Source: Willem de Graaf, http://www-groups.dcs.st-and.ac.uk/~wdg/slides/node150.html

Key Schedule

- Need 16 48-bit keys
  - Best security: just use 16 independent keys
  - 768 key bits
- 56-bit key used
  - Represented by 8 bytes (1 parity bit per byte)
  - Produce 48-bit round keys by shifting and permuting

DES Key Schedule



Cracking DES

90B keys per second
Cost < $250K (in 1998)
56 hours to solve RSA DES Challenge

Breaking DES by Brute Force

- RSA DES challenges:
  - 1997: 96 days (using 70,000 machines)
  - Feb 1998: 41 days (distributed.net)
  - July 1998: 56 hours (custom hardware)
  - January 1999: 22 hours (EFF + distributed.net)
    - 245 Billion keys per second
  - May 2005, NIST withdraw DES (FIPS 46-3)
  - Nov 2008, <1 day (FPGA-based RIVYERA machine, $10,000 hardware cost)
- NSA can probably crack DES routinely (but they won't admit it)

Double DES?

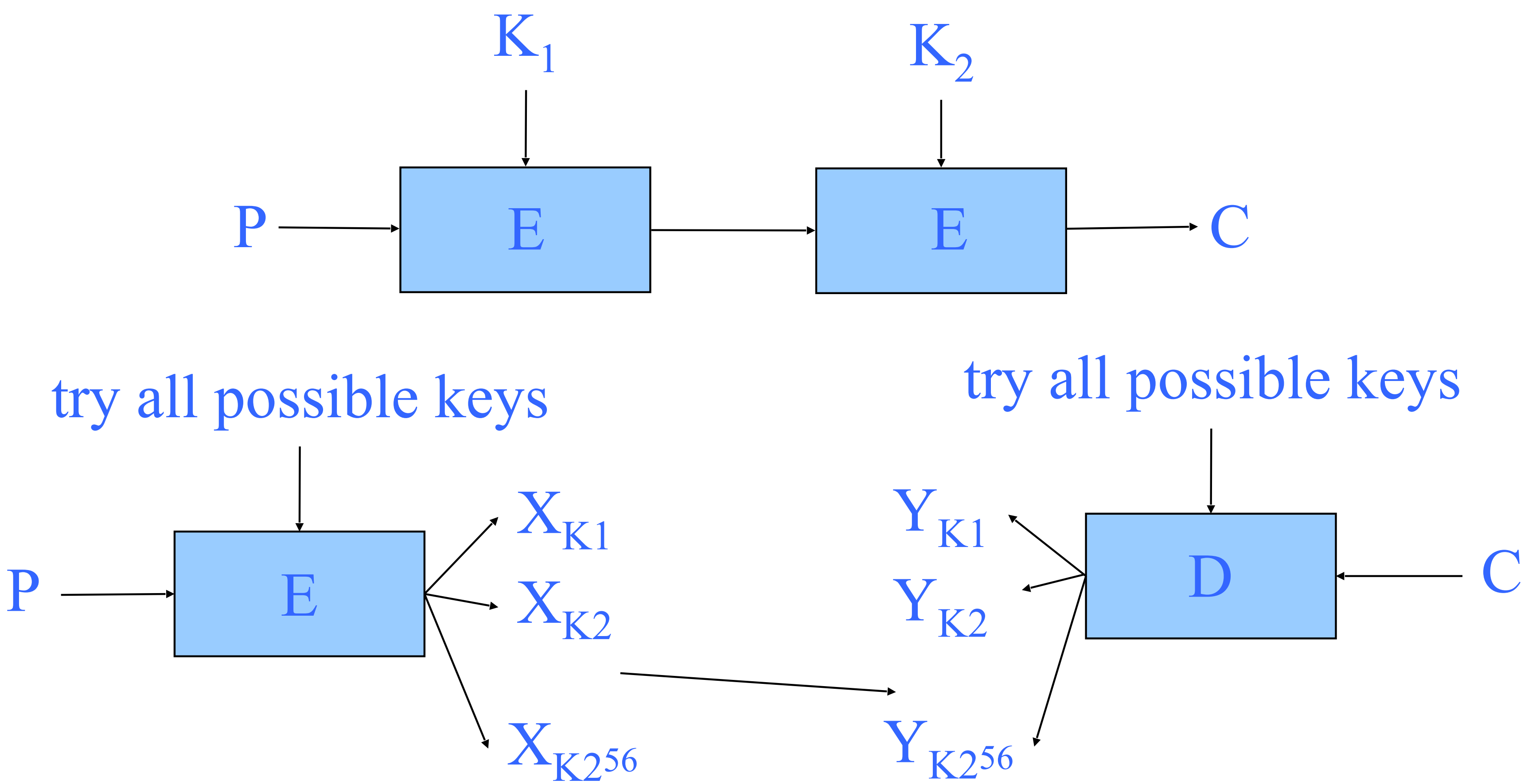
- $C = E_{K2} (E_{K1} (P))$

- Effective key size of Double DES?

  $= 2^{56} * 2^{56} = 2^{112}$

WRONG!

Known Plaintext Attack



One $X_{Ki} = Y_{Kj}$ means $K_1 = K_i$ and $K_2 = K_j$

Meet-in-the-Middle Attack

- $C = E_{K2} (E_{K1} (P))$
- $X = E_{K1} (P) = D_{K2} (C)$
- Brute force attack (given one P/C pair):

  calculate $E_{K1} (P)$ for all keys (2^{56} work)

  calculate $D_{K2} (C)$ for all keys (2^{56} work)

  the match gives the keys
- Total work = $2 * 2^{56} + 56*2^{56} = 58*2^{57}$

Hmmm…maybe thrice?

2-Key Triple DES

- $C = E_{K1} (D_{K2} (E_{K1} (P)))$
- Why D_{K2} not E_{K2}?
  - Backwards compatibility with DES
  - If K1 = K2: $C = E_{K1} (D_{K1} (E_{K1} (P))) = E_{K1} (P)$
- Actual key size = 56 + 56 bits = 112 bits
- Meet-in-the-middle?
  - $X = E_{K1} (P) = D_{K1} (E_{K2} (C))$

2^{56} need to try 2^{112}

How secure is Triple-DES

- Brute force search: 2^{112} keys
  - Best DES attack: 245 B keys/second
  - $\approx 6.7 * 10^{14}$ years (compared to 22 hours)
  - 10^{11} years = total lifetime of universe (closed universe theory)
- Best known attack - reduces to $2^{120-\log_2 n}$
  - n = number of known P-C pairs
  - n = 2^{64}, work is 2^{56}

Realistic?

3-Key Triple DES

- $C = E_{K3} (D_{K2} (E_{K1} (P)))$
- $H(K) = 168$
- Used by PGP, S/MIME
- How much work to brute-force?
  - Meet-in-the-middle:

$$X = D_{K3} (C) = D_{K2} (E_{K1} (P))$$

$$2^{56} \quad + \quad 2^{112}$$