CISC 468: CRYPTOGRAPHY

LESSON 6: BLOCK CIPHERS (CONTINUED)

Furkan Alaca

TODAY, WE WILL LEARN ABOUT...

1. DES Decryption
2. Triple DES
3. Common attack strategies on block ciphers
4. Overview of AES

READINGS

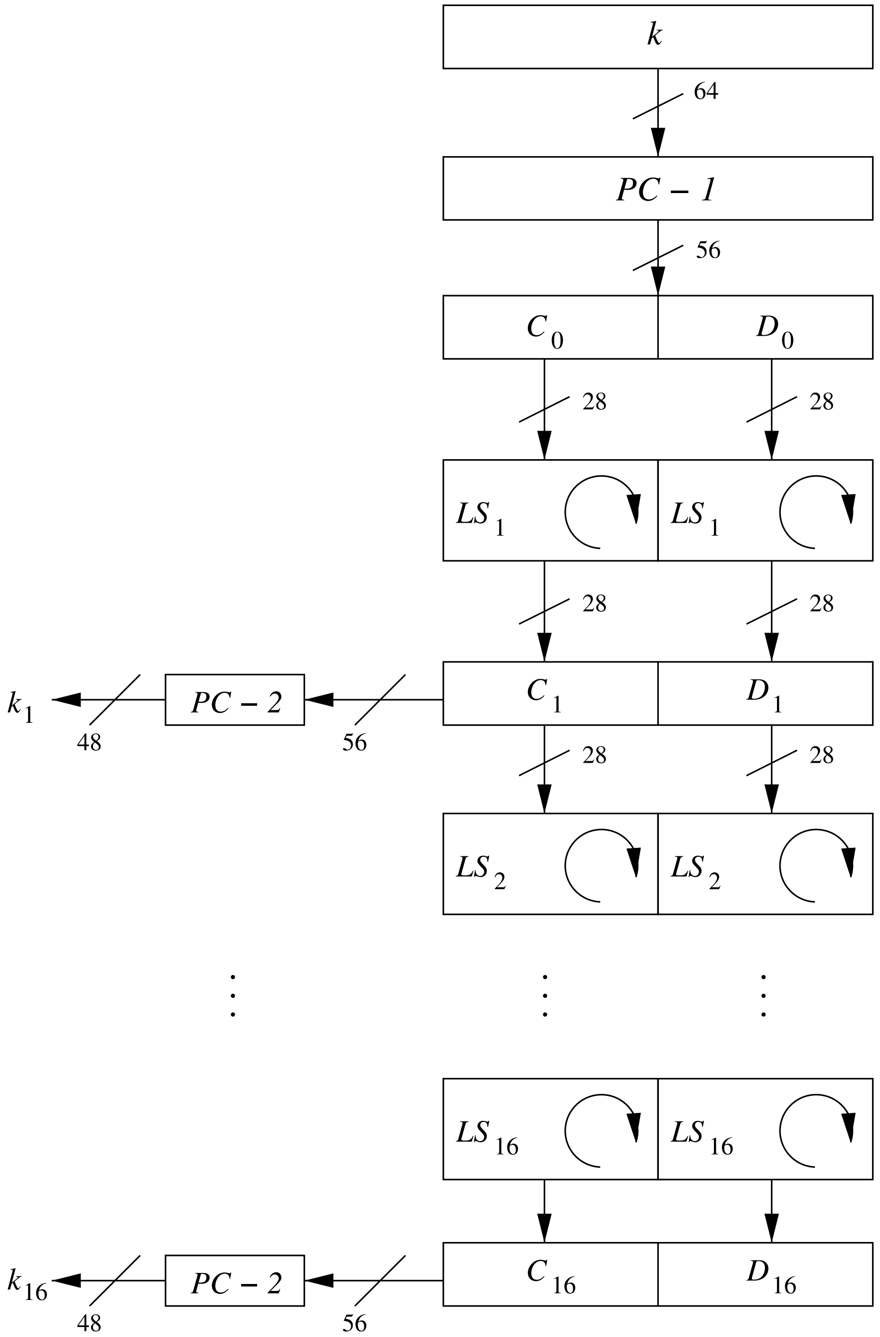
- Section 3.4: DES Decryption, Paar & Pelzl
- Section 3.5: Security of DES, Paar & Pelzl
- Section 3.7.2: Triple DES (3DES) and DESX, Paar & Pelzl
- Section 5.3.1: Double Encryption and Meet-in-the-Middle Attack, Paar & Pelzl
- Section 4.1: Introduction to AES, Paar & Pelzl
- Section 4.2: Overview of the AES Algorithm, Paar & Pelzl

DES: DECRYPTION

- Since DES is a Feistel cipher, decryption and encryption are the same function with the key schedule reversed
  - We will see why this works

DES: DECRYPTION

- Across the 16 rounds, the key schedule shifts the two halves of the 56-bit key to the left by 28 bit positions in total
  - So we already saw that $C_0 = C_{16}$ and $D_0 = D_{16}$
  - If we want to reverse the key schedule, we should be able to derive k_{16} directly from (C_0, D_0)



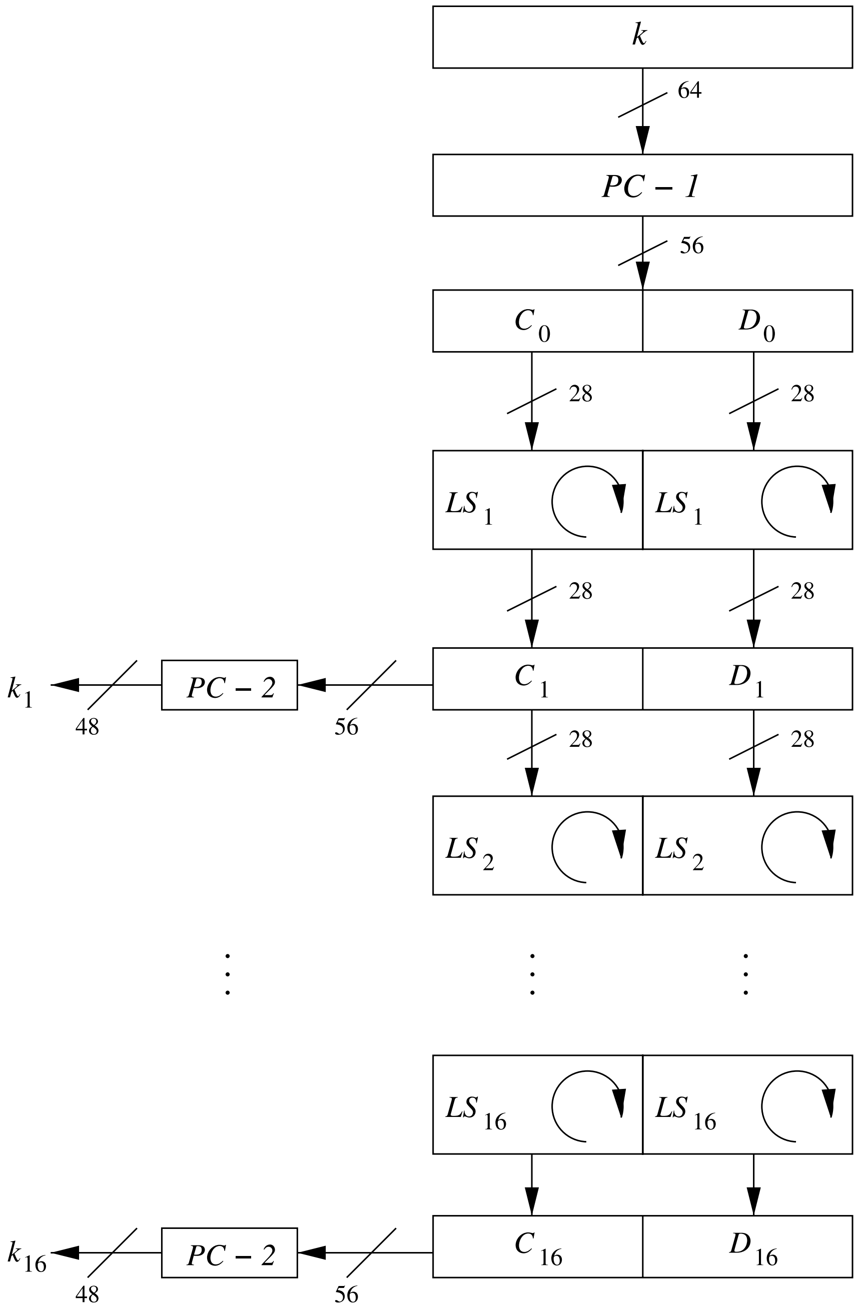
DES: DECRYPTION (2)

To compute k_{16} from k:

$$\begin{aligned} k_{16} &= PC-2(C_{16}, D_{16}) \\ &= PC-2(C_0, D_0) \\ &= PC-2(PC-1(k)) \end{aligned}$$

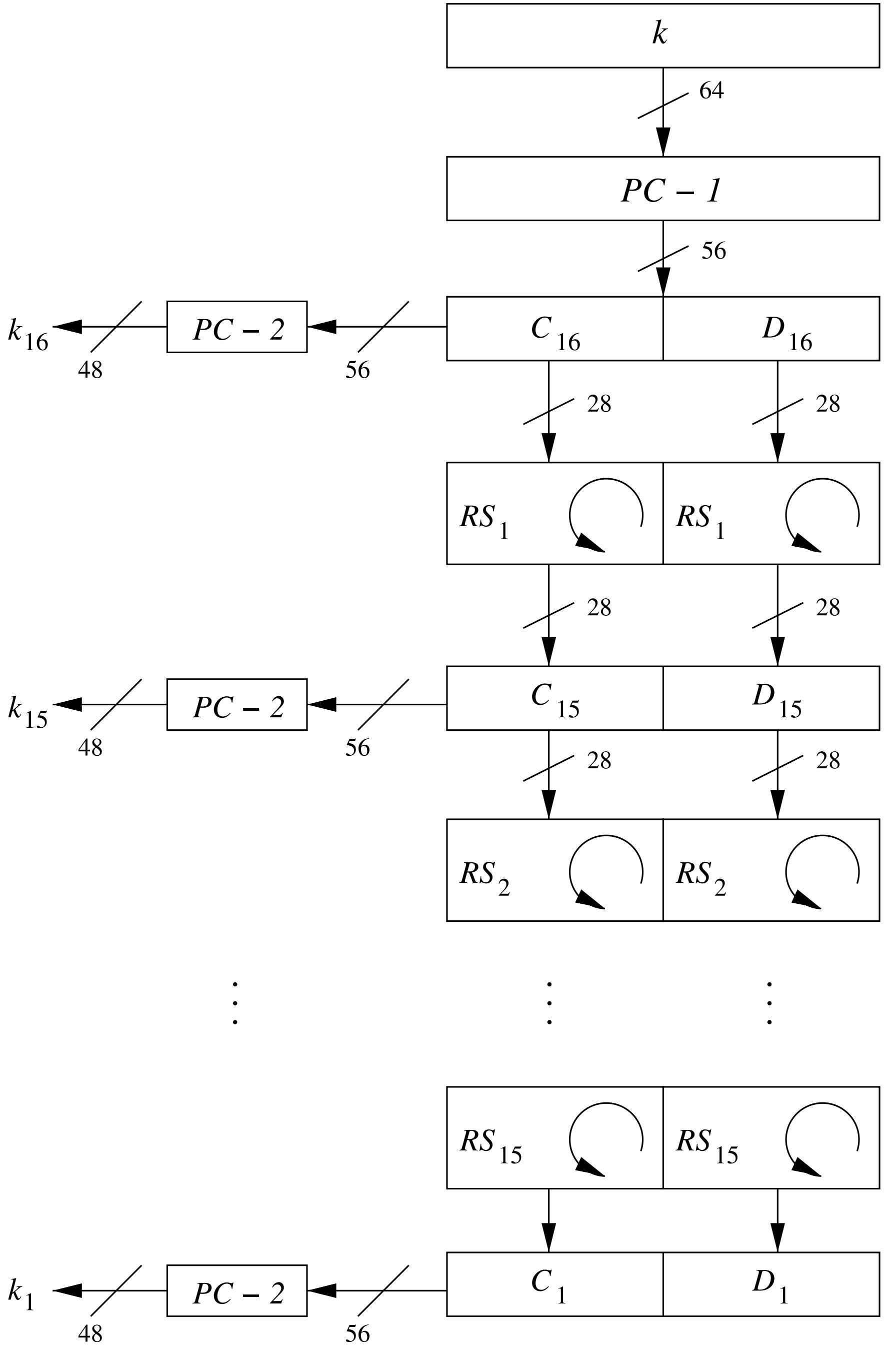
Right shifts revert the left shifts and compute k_{15}:

$$\begin{aligned} k_{15} &= PC-2(C_{15}, D_{15}) \\ &= PC-2(RS_2(C_{16}), RS_2(D_{16})) \\ &= PC-2(RS_2(C_0), RS_2(D_0)) \end{aligned}$$



DES: DECRYPTION (3)

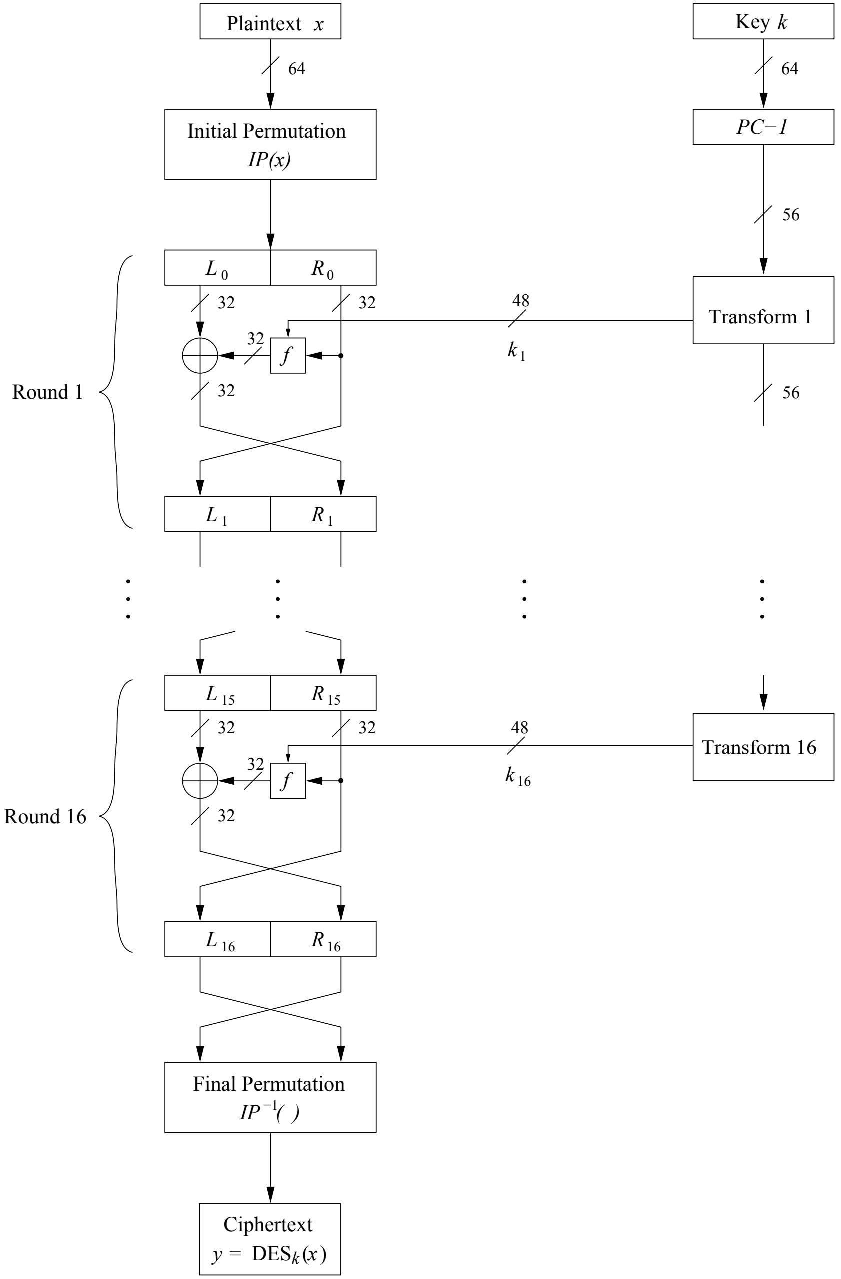
- The remaining subkeys are similarly derived using right shifts
  - For round 1, no shifting is done
  - For rounds 2, 9, and 16, C_x and D_x are shifted by one bit each
  - For remaining rounds, C_x and D_x are shifted by two bits each
- Updated diagram on the right shows the reversed key schedule for decryption

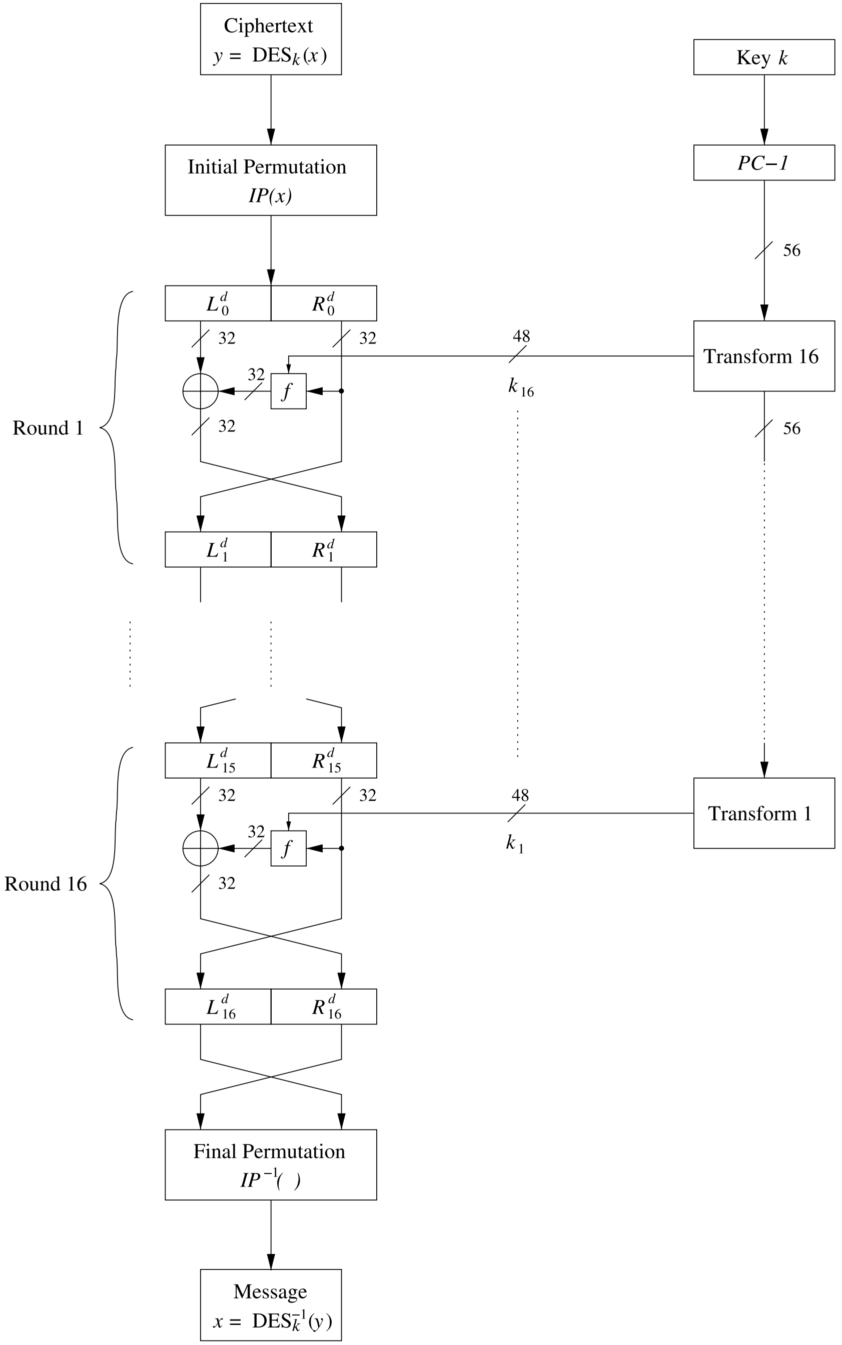


DECRYPTION IN FEISTEL NETWORKS

- The decryption function reverses the encryption function round-by-round, i.e.,
  - Decryption round 1 reverses encryption round 16
  - Decryption round 2 reverses encryption round 15
  - ...
  - Decryption round 16 reverses encryption round 1

DECRYPTION IN FEISTEL NETWORKS (2)





DECRYPTION IN FEISTEL NETWORKS (3)

Using the diagrams on the previous slide, we can see that:

$$\begin{aligned}(L_0^d, R_0^d) &= IP(Y) = IP(IP^{-1}(R_{16}, L_{16})) \\ &= (R_{16}, L_{16}).\end{aligned}$$

To show that decryption round 1 reverses encryption round 16, we show that $(L_1^d, R_1^d) = (R_{15}, L_{15})$:

$$L_1^d = R_0^d = L_{16} = R_{15} \text{ and}$$

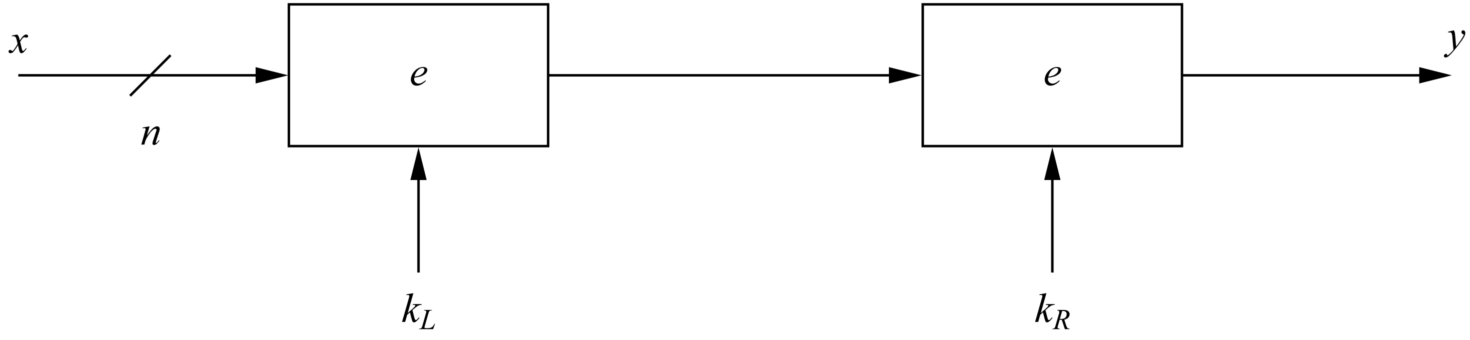
$$\begin{aligned}R_1^d &= L_0^d \oplus f(R_0^d, k_{16}) = R_{16} \oplus f(L_{16}, k_{16}) \\ &= [L_{15} \oplus f(R_{15}, k_{16})] \oplus f(R_{15}, k_{16}) \\ &= L_{15} \oplus [f(R_{15}, k_{16}) \oplus f(R_{15}, k_{16})] = L_{15}.\end{aligned}$$

SECURITY OF DES: BRUTE-FORCE ATTACKS

- DES is vulnerable to brute-force attacks due to its small 56-bit keyspace
  - With modern resources, successful attack requires ~1 day
- Triple-DES (3DES) performs three DES encryptions in series, using three separate keys
  - Total key length is 3x56 = 168 bits
  - *Meet-in-the-middle* attack reduces effective key strength to 112 bits

MEET-IN-THE-MIDDLE ATTACK

- The meet-in-the-middle attack is a lookup-table strategy applicable to encryption schemes that perform multiple encryption operations in sequence
- For double-encryption, observe from the figure below that $e(x, k_L) = d(y, k_R)$
  - Allows elimination of some $k_L - k_R$ pairs from the search
  - Attack is extendable to >2 subsequent encryptions



MEET-IN-THE-MIDDLE ATTACK ON DOUBLE ENCRYPTION

- Consider DES (56-bit key), but can apply to any block cipher
- List the encryptions of x_1 using all possible 2^{56} keys k_L
- List the decryptions of y_1 using all possible 2^{56} keys k_R
- Compare the two lists and eliminate the values not present in both lists
- There may exist multple key pairs that map x_1 to y_1, so we must verify the key pair on several plaintext-ciphertext pairs
- Total computations: $2^{56} + 2^{56} = 2^{57}$
  - Requires more memory, but fewer operations than the $2^{56\times2} = 2^{112}$ required by naive brute-force strategy

SECURITY OF DES: ANALYTICAL ATTACKS

- Vulnerability to analytical attacks was suspected soon after release of DES
    - Many interesting theories, e.g., due to the design criteria of S-boxes having been kept secret
- Some faster-than-brute-force analytical attacks were discovered, but were considered impractical due to the large number (2^{43}) of plaintext-ciphertext pairs needed
    - Strong argument in favour of frequent *rekeying*

SECURITY OF DES: BIRTHDAY ATTACKS

- Block cipher security is often framed in terms of key length
- Block length l matters too, due to *birthday attacks*
  - Based on the *Birthday Paradox*
- Must rekey well before $2^{\frac{l}{2}}$ blocks are encrypted
  - Limit is 256 EB if block size is 128-bit
  - But only 32GB if block size is 64-bit
  - Otherwise, ciphertext collisions will occur and can reveal secrets
- Sweet32 demonstrates attack feasibility on protocols that use 64-bit ciphers (e.g., DES, Blowfish) and repeatedly send same data (e.g., authentication cookies in HTTPS)

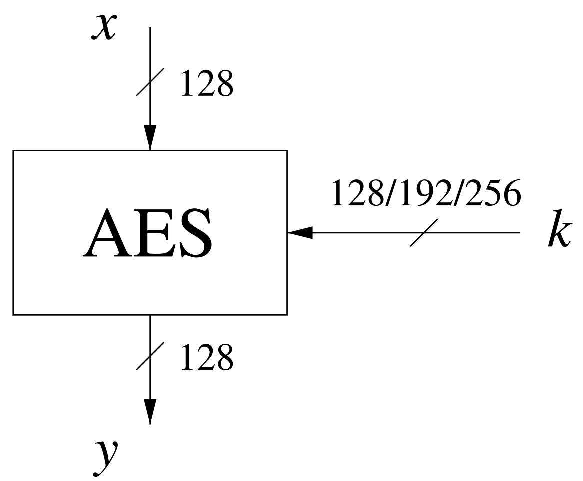
THE ADVANCED ENCRYPTION STANDARD (AES)

- AES was the result of an open competition
- At the last stage of the selection process, there were five finalists: Rijndael, Mars, RC6, Serpent, and Twofish
  - Mars, RC6, and Twofish are Feistel ciphers
- Rijndael was selected as the winner

OVERVIEW OF AES PARAMETERS

- Block size is 128-bit
- Key size can be 128, 192, or 256 bits
  - US government allows 128-bit for SECRET level data, 192- and 256-bit for TOP SECRET level data
- Number of rounds determined by key size
  - Not a Feistel Cipher, so each round encrypts all bits

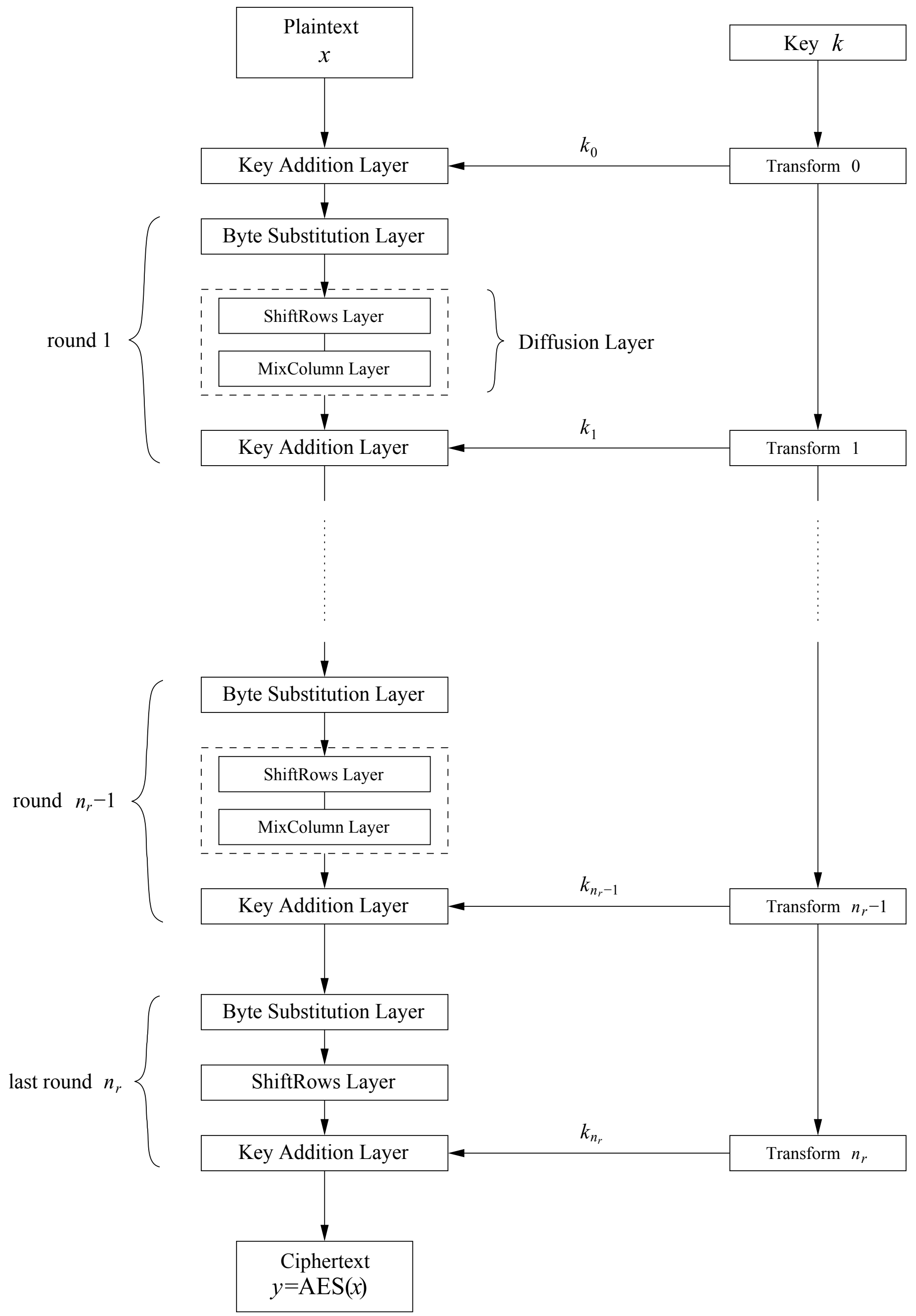
Key length	Num. Rounds
128 bit	10
192 bit	12
256 bit	14



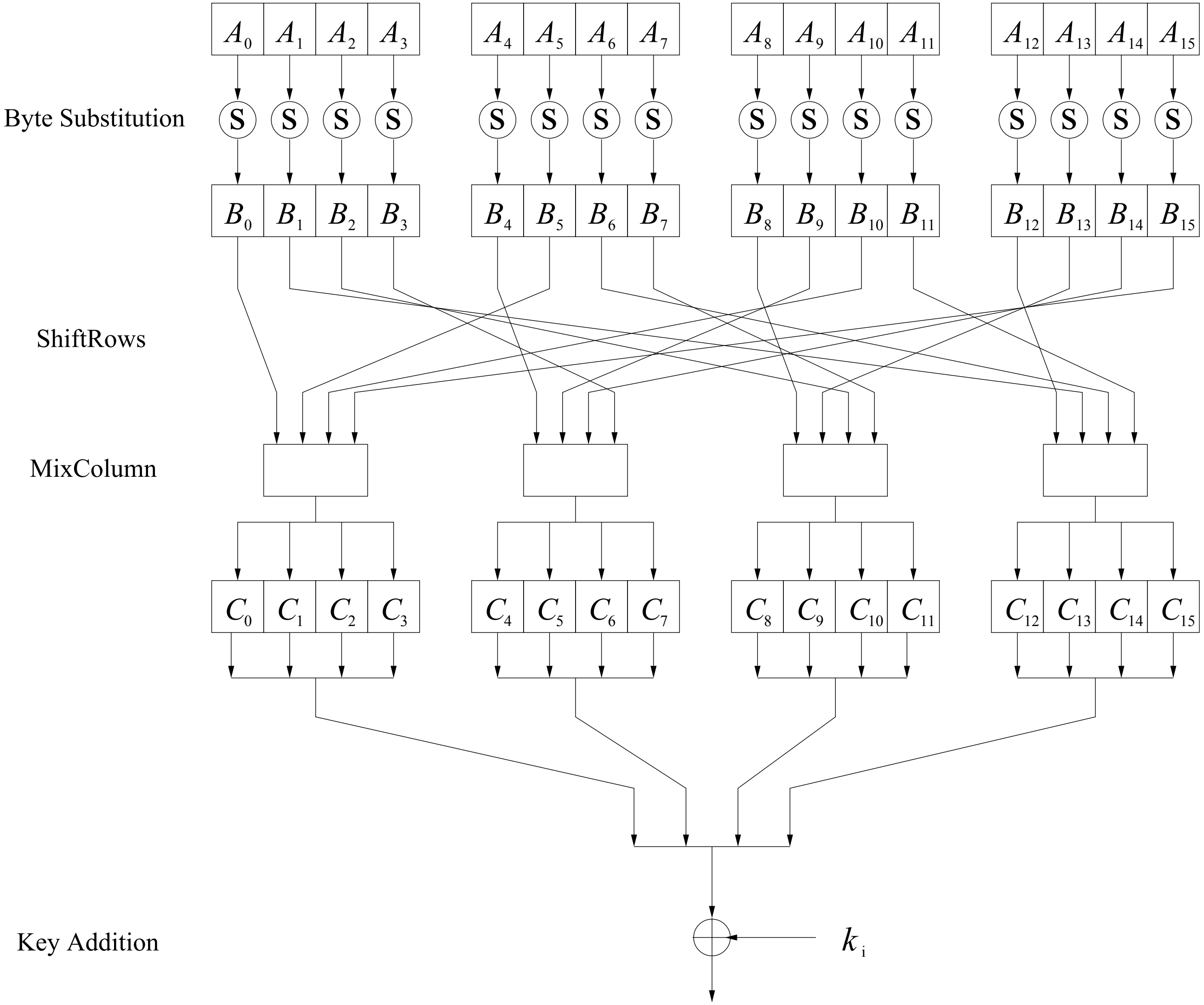
OVERVIEW OF AES LAYERS

- Each round encrypts consists of three layers:
  - *Key Addition layer* XORs the internal state with a subkey derived by the key schedule
  - *Byte Substitution layer* performs a nonlinear transformation using S-Boxes with special mathematiacl properties
  - *Diffusion layer* performs linear operations using *ShiftRows* and *MixColumn*
- Uses Galois field arithmetic

AES BLOCK DIAGRAM



AES ROUND FUNCTION BLOCK DIAGRAM



A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15}
Byte Substitution
S S S S S S S S S S S S S S S S
B_0 B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15}
ShiftRows
MixColumn
C_0 C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 C_9 C_{10} C_{11} C_{12} C_{13} C_{14} C_{15}
Key Addition
k_i

AES BYTE SUBSTITUTION LAYER

- AES is a byte-oriented cipher (as opposed to DES, which is bit-oriented)
- The 16-byte (128-bit) input is fed into 16 identical S-Boxes
- Each S-Box substitutes the input byte A_i by another byte B_i
- The only non-linear element of AES, i.e., `ByteSub(A) + ByteSub(B) != ByteSub(A+B)`
- The S-Box substitution is a bijective mapping, so it can be reversed for decryption

AES DIFFUSION LAYER

The ShiftRows operation, treating the 16-byte state as a 4x4 matrix, cyclically shifts the rows as follows:

$$\begin{array}{|c|c|c|c|}\hline B_0 & B_4 & B_8 & B_{12} \\ \hline B_1 & B_5 & B_9 & B_{13} \\ \hline B_2 & B_6 & B_{10} & B_{14} \\ \hline B_3 & B_7 & B_{11} & B_{15} \\ \hline \end{array} \longrightarrow \begin{array}{|c|c|c|c|}\hline B_0 & B_4 & B_8 & B_{12} \\ \hline B_5 & B_9 & B_{13} & B_1 \\ \hline B_{10} & B_{14} & B_2 & B_6 \\ \hline B_{15} & B_3 & B_7 & B_{11} \\ \hline \end{array} \begin{array}{l} \text{no shift} \\ \longrightarrow \text{three positions right shift} \\ \longrightarrow \text{two positions right shift} \\ \longrightarrow \text{one position right shift} \end{array}$$

Then, MixColumn performs a matrix multiplication on each column of the resulting matrix so that after three rounds each byte in the state matrix depends on all 16 plaintext bytes—all arithmetic is performed in the Galois field $GF(2^8)$.

> *AES is as secure as a block cipher can be, and it will never be broken. Fundamentally, AES is secure because all output bits depend on all input bits in some complex, pseudorandom way. To achieve this, the designers of AES carefully chose each component for a particular reason—MixColumns for its maximal diffusion properties and SubBytes for its optimal non-linearity —and they have shown that this composition protects AES against whole classes of cryptanalytic attacks.*

—JP Aumasson, Serious Cryptography: A Practical Introduction to Modern Encryption

ATTACKS ON AES

- Faster-than-brute-force attacks exist, but are computationally infeasible
  - e.g., AES-128 key can be recovered by performing 2^{126} operations
- No practical attacks exist against the AES cipher
  - But the strength of AES doesn't prevent its misuse, which can be exploited (as we will see)
- Attacks exist on weakened versions of AES with fewer rounds
  - e.g., 7/10 rounds for AES-128, 8/12 rounds for AES-192
  - Common practice to try to break weakened versions first

CONCLUDING REMARKS

- Standard DES can be easily broken by brute-force
  - 3DES mitigates this by increasing the key strength
- DES (and therefore 3DES) still seems resilient to analytic attacks, so legacy 3DES applications are thought to be secure **if they rekey frequently** to defend against birthday attacks
  - In 2017, NIST announced that 3DES using a three-key bundle should only be used to encrypt up to 2^{20} blocks
  - In 2019, NIST announced that 3DES was being deprecated and would be disallowed for use after 2023
- Use AES instead, and make sure you use it correctly (more on this next week)