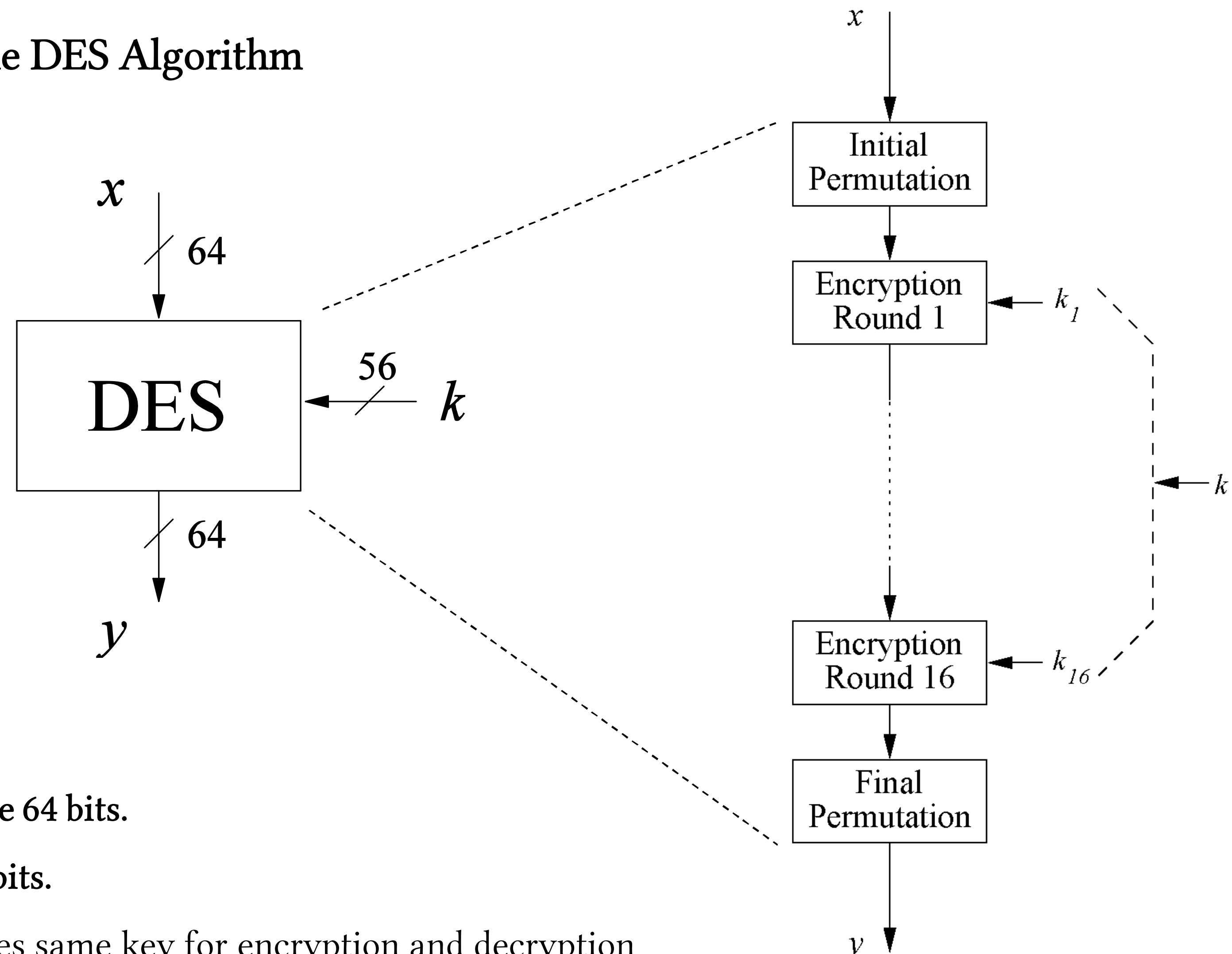
بسم الله الرحمن الرحیم

# Network Security

Ali Shakiba
Vali-e-Asr University of Rafsanjan
ali.shakiba@vru.ac.ir
www.1ali.ir

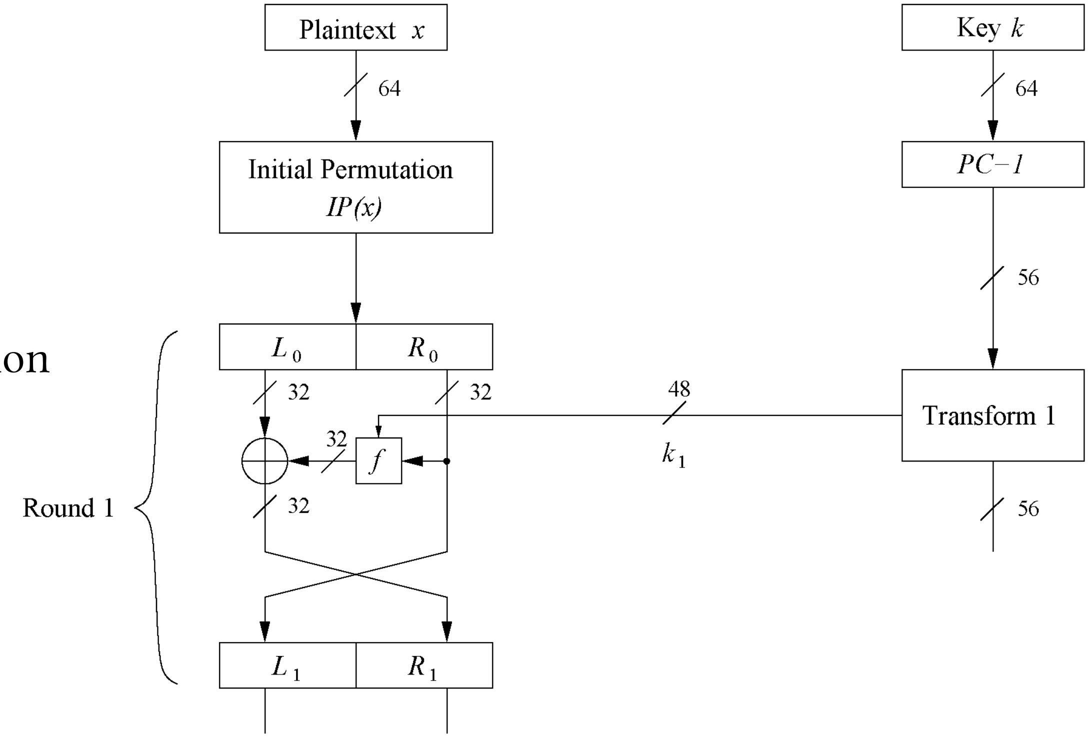
## Overview of the DES Algorithm




- **Encrypts blocks of size 64 bits.**
- **Uses a key of size 56 bits.**
- Symmetric cipher: uses same key for encryption and decryption
- Uses 16 rounds which all perform the identical operation
- Different subkey in each round derived from main key

## The DES Feistel Network (1)

- DES structure is a *Feistel network*
- Advantage: encryption and decryption differ only in keyschedule

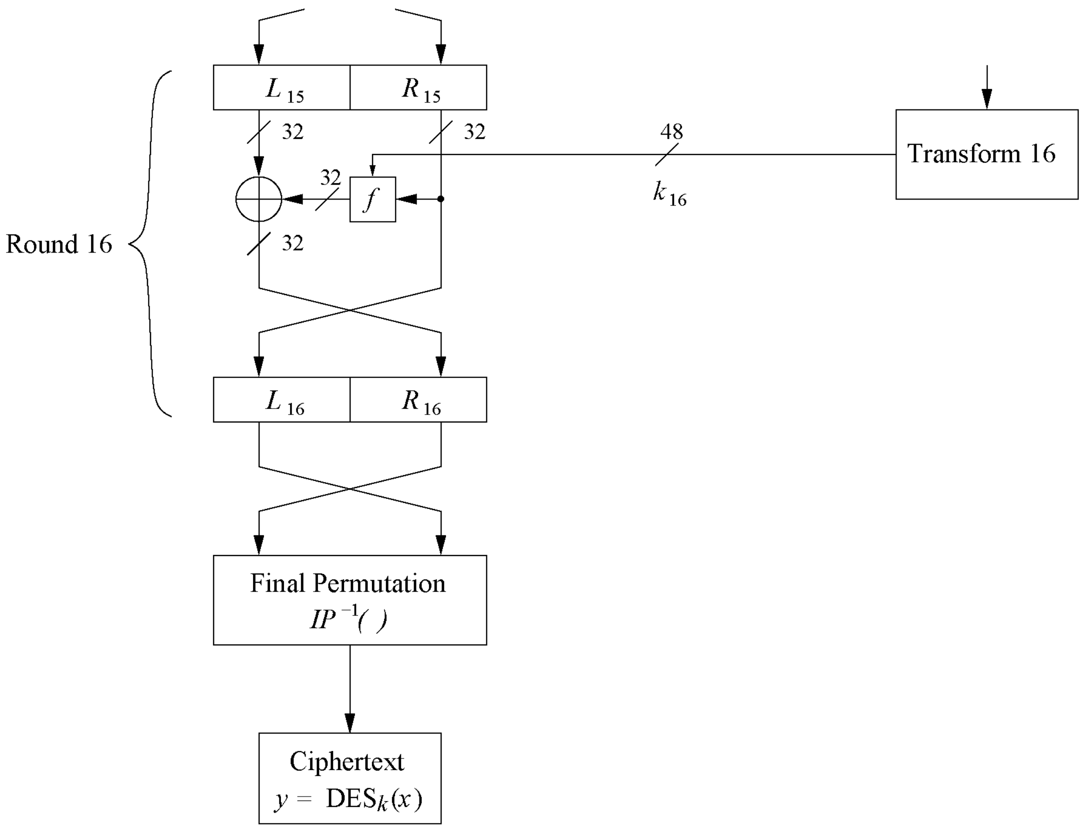



- Bitwise initial permutation, then 16 rounds
  1. Plaintext is split into 32-bit halves $L_i$ and $R_i$
  2. $R_i$ is fed into the function $f$, the output of which is then XORed with $L_i$
  3. Left and right half are swapped
- Rounds can be expressed as:

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

## The DES Feistel Network (2)

- L and R swapped again at the end of the cipher, i.e., after round 16 followed by a final permutation



# Content of this Chapter

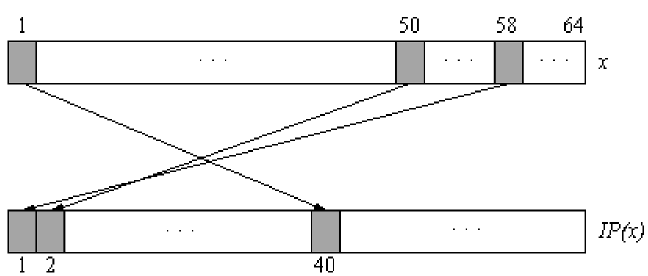
- Introduction to DES
- Overview of the DES Algorithm
- **Internal Structure of DES**
- Decryption
- Security of DES

## Initial and Final Permutation

- Bitwise Permutations.
- Inverse operations.
- Described by tables *IP* and $IP^{-1}$.

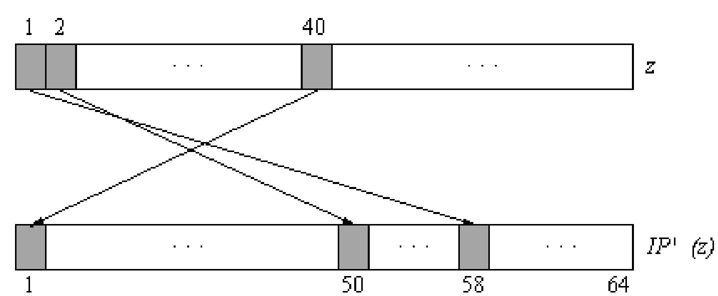
### Initial Permutation

| IP | | | | | | | |
|---|---|---|---|---|---|---|---|
| 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2 |
| 60 | 52 | 44 | 36 | 28 | 20 | 12 | 4 |
| 62 | 54 | 46 | 38 | 30 | 22 | 14 | 6 |
| 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8 |
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 |
| 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 |
| 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 |



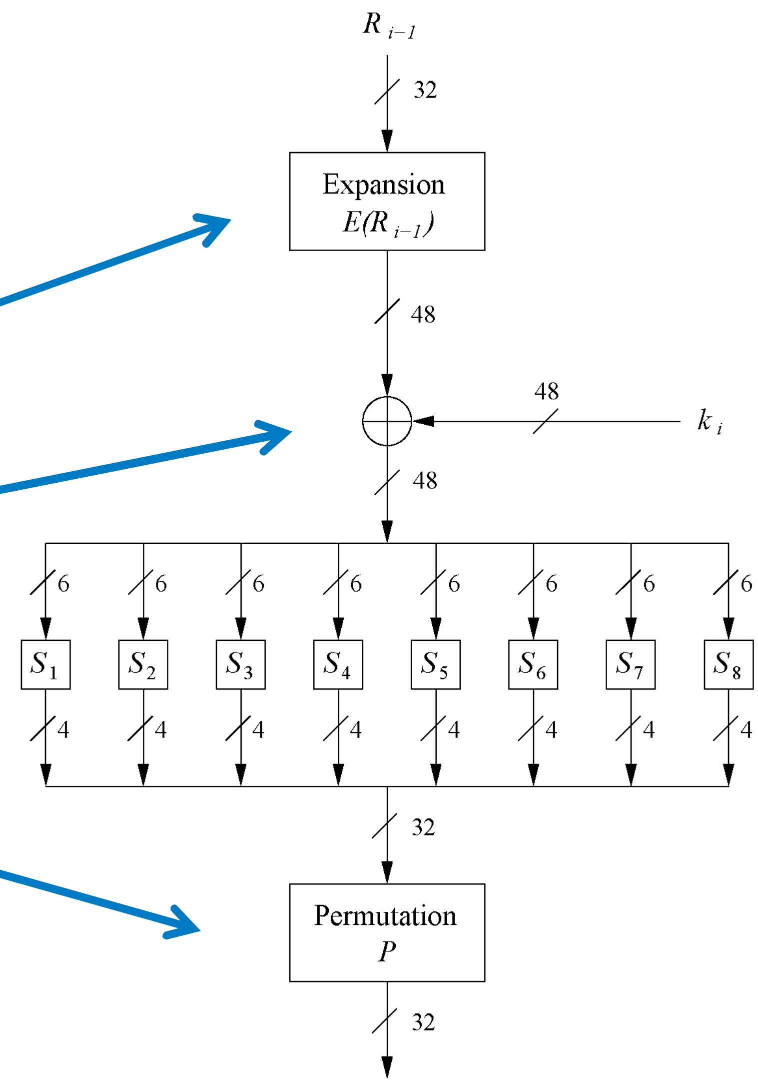

### Final Permutation

| $IP^{-1}$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| 40 | 8 | 48 | 16 | 56 | 24 | 64 | 32 |
| 39 | 7 | 47 | 15 | 55 | 23 | 63 | 31 |
| 38 | 6 | 46 | 14 | 54 | 22 | 62 | 30 |
| 37 | 5 | 45 | 13 | 53 | 21 | 61 | 29 |
| 36 | 4 | 44 | 12 | 52 | 20 | 60 | 28 |
| 35 | 3 | 43 | 11 | 51 | 19 | 59 | 27 |
| 34 | 2 | 42 | 10 | 50 | 18 | 58 | 26 |
| 33 | 1 | 41 | 9 | 49 | 17 | 57 | 25 |



## The f-Function

- main operation of DES
- *f*-Function inputs: $R_{i-1}$ and round key $k_i$
- 4 Steps:
  1. Expansion $E$
  2. XOR with round key
  3. S-box substitution
  4. Permutation

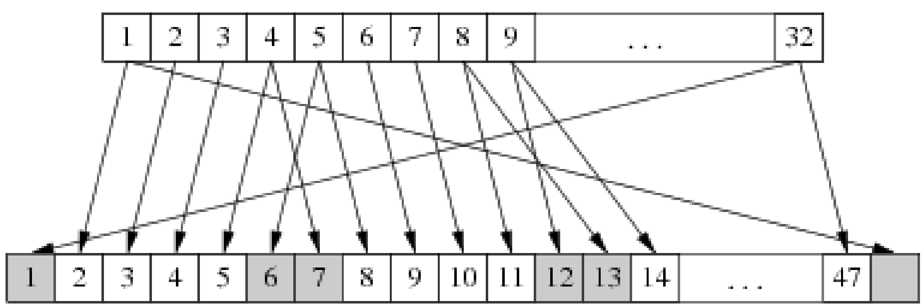


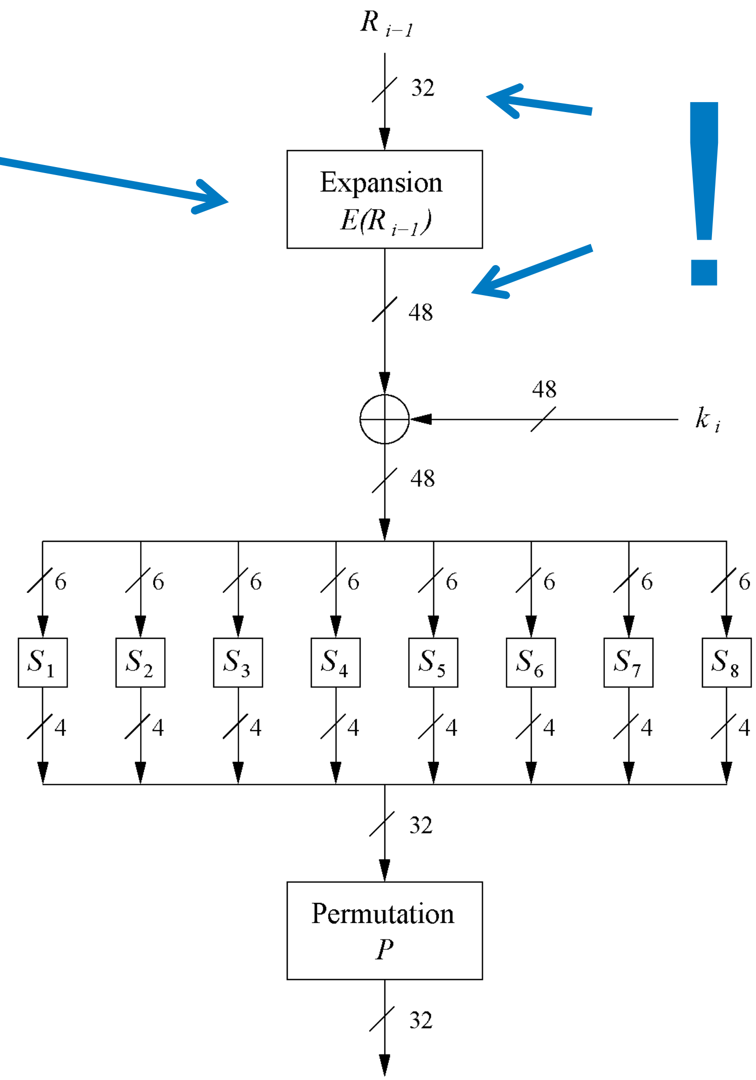
## The Expansion Function E

1. Expansion *E*

- main purpose: increases diffusion

| *E* | | | | | |
|---|---|---|---|---|---|
| 32 | 1 | 2 | 3 | 4 | 5 |
| 4 | 5 | 6 | 7 | 8 | 9 |
| 8 | 9 | 10 | 11 | 12 | 13 |
| 12 | 13 | 14 | 15 | 16 | 17 |
| 16 | 17 | 18 | 19 | 20 | 21 |
| 20 | 21 | 22 | 23 | 24 | 25 |
| 24 | 25 | 26 | 27 | 28 | 29 |
| 28 | 29 | 30 | 31 | 32 | 1 |



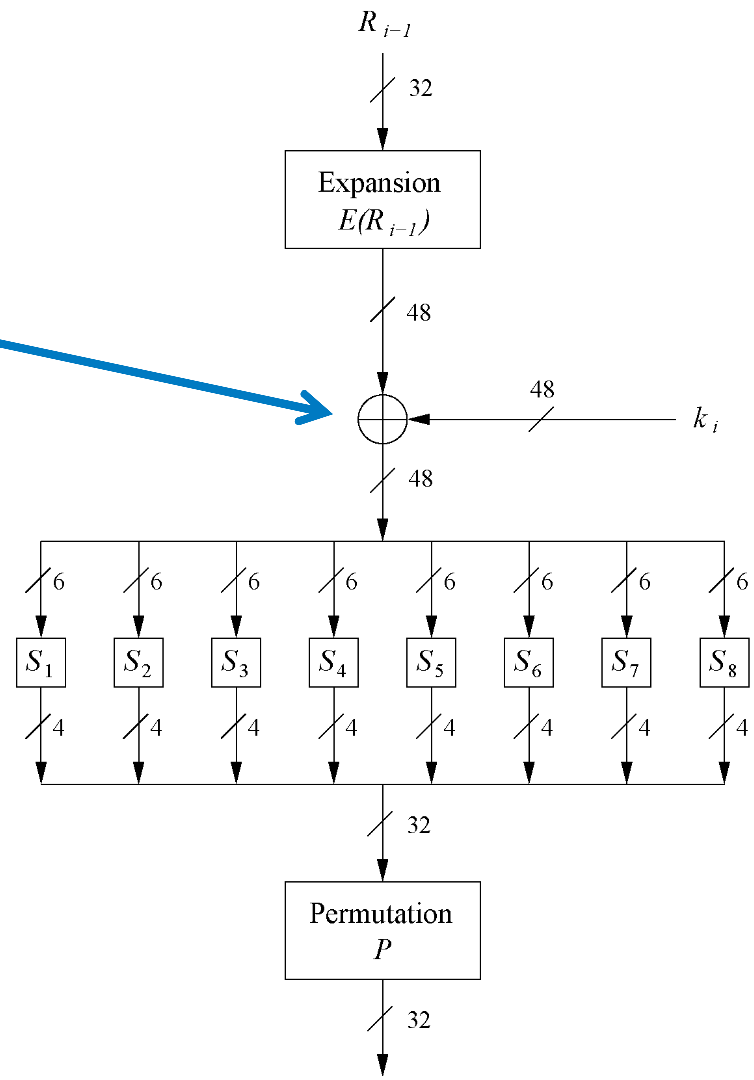



## Add Round Key

### 2. XOR Round Key

- Bitwise XOR of the round key and the output of the expansion function *E*

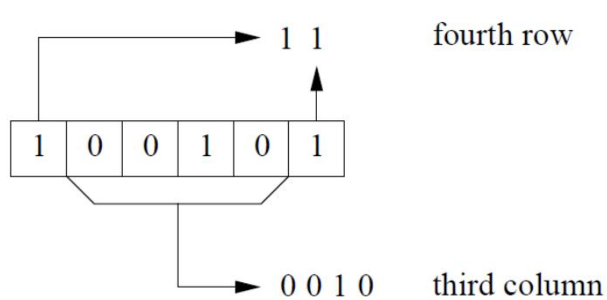
- Round keys are derived from the main key in the DES keyschedule (in a few slides)

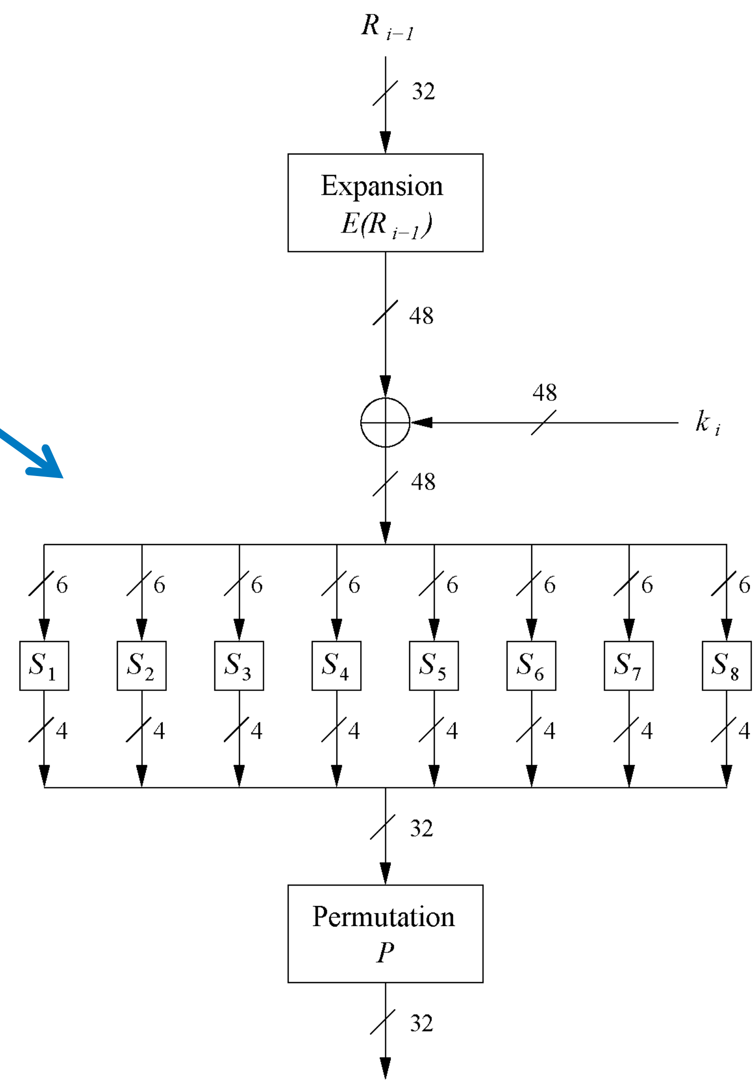


## The DES S-Boxes

### 3. S-Box substitution

- Eight substitution tables.
- 6 bits of input, 4 bits of output.
- Non-linear and resistant to differential cryptanalysis.
- Crucial element for DES security!
- Find all S-Box tables and S-Box design criteria in *Understanding Cryptography* Chapter 3.




| $S_1$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 14 | 04 | 13 | 01 | 02 | 15 | 11 | 08 | 03 | 10 | 06 | 12 | 05 | 09 | 00 | 07 |
| 1 | 00 | 15 | 07 | 04 | 14 | 02 | 13 | 01 | 10 | 06 | 12 | 11 | 09 | 05 | 03 | 08 |
| 2 | 04 | 01 | 14 | 08 | 13 | 06 | 02 | 11 | 15 | 12 | 09 | 07 | 03 | 10 | 05 | 00 |
| 3 | 15 | 12 | 08 | 02 | 04 | 09 | 01 | 07 | 05 | 11 | 03 | 14 | 10 | 00 | 06 | 13 |



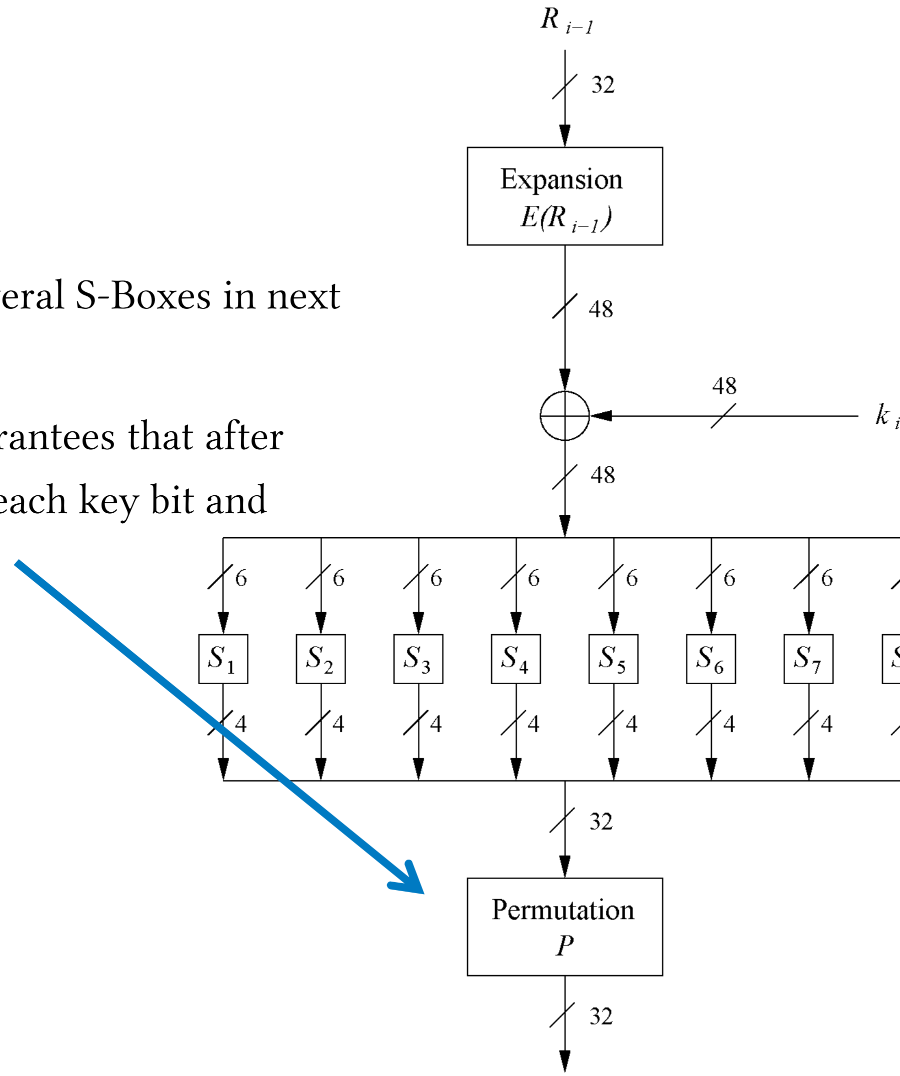
Chapter 3 of *Understanding Cryptography* by Christof Paar and Jan Pelzl

## The Permutation P

### 4. Permutation P

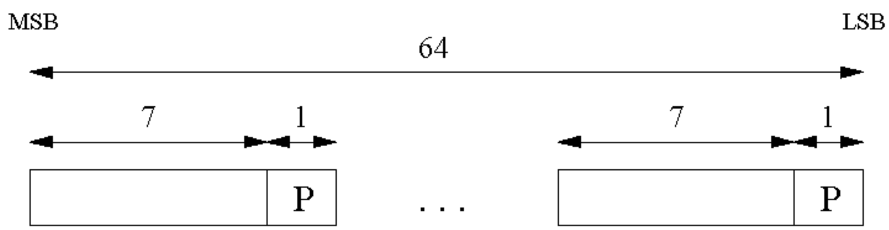
- Bitwise permutation.
- Introduces diffusion.
- Output bits of one S-Box effect several S-Boxes in next round
- Diffusion by E, S-Boxes and P guarantees that after Round 5 every bit is a function of each key bit and each plaintext bit.

| P | | | | | | | |
|---|---|---|---|---|---|---|---|
| 16 | 7 | 20 | 21 | 29 | 12 | 28 | 17 |
| 1 | 15 | 23 | 26 | 5 | 18 | 31 | 10 |
| 2 | 8 | 24 | 14 | 32 | 27 | 3 | 9 |
| 19 | 13 | 30 | 6 | 22 | 11 | 4 | 25 |



## Key Schedule (1)

- Derives 16 round keys (or *subkeys*) $k_i$ of 48 bits each from the original 56 bit key.
- The input key size of the DES is 64 bit: **56 bit key** and 8 bit parity: !




P = parity bit

- **Parity bits are removed** in a first **permuted choice** *PC-1*:
  (note that the bits 8, 16, 24, 32, 40, 48, 56 and 64 are not used at all)

| $PC-1$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 |
| 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2 |
| 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 |
| 60 | 52 | 44 | 36 | 63 | 55 | 47 | 39 |
| 31 | 23 | 15 | 7 | 62 | 54 | 46 | 38 |
| 30 | 22 | 14 | 6 | 61 | 53 | 45 | 37 |
| 29 | 21 | 13 | 5 | 28 | 20 | 12 | 4 |

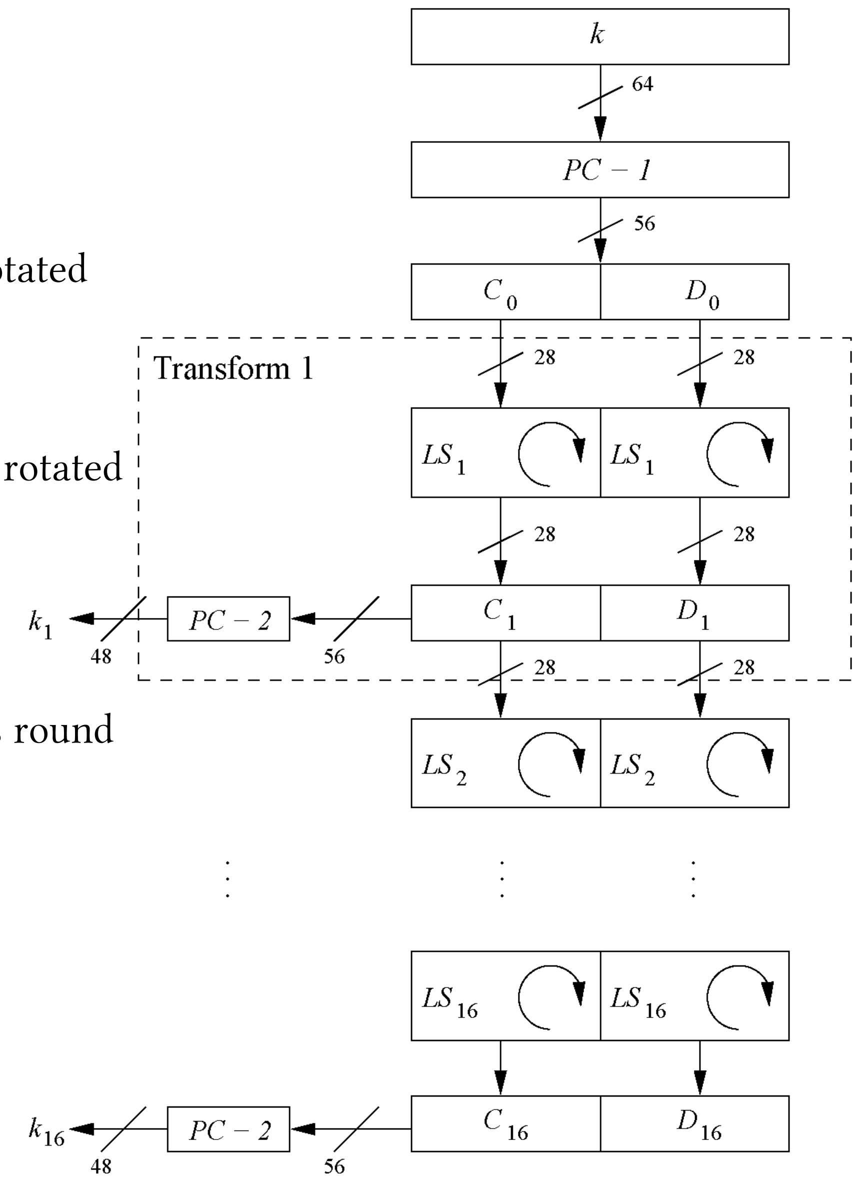
## Key Schedule (2)

- Split key into 28-bit halves $C_0$ and $D_0$.
- In **rounds $i$ = 1, 2, 9 ,16,** the two halves are each rotated left by **one bit**.
- In **all other rounds** where the two halves are each rotated left by **two bits**.
- *In each round i permuted choice **PC-2*** selects a permuted subset of 48 bits of $C_i$ and $D_i$ as round key $k_i$, i.e. **each $k_i$ is a permutation of $k$**!

| PC − 2 | | | | | | | |
|---|---|---|---|---|---|---|---|
| 14 | 17 | 11 | 24 | 1 | 5 | 3 | 28 |
| 15 | 6 | 21 | 10 | 23 | 19 | 12 | 4 |
| 26 | 8 | 16 | 7 | 27 | 20 | 13 | 2 |
| 41 | 52 | 31 | 37 | 47 | 55 | 30 | 40 |
| 51 | 45 | 33 | 48 | 44 | 49 | 39 | 56 |
| 34 | 53 | 46 | 42 | 50 | 36 | 29 | 32 |

- **Note:** The total number of rotations:

  4 x 1 + 12 x 2 = 28 $\Rightarrow D_0 = D_{16}$ and $C_0 = C_{16}$!

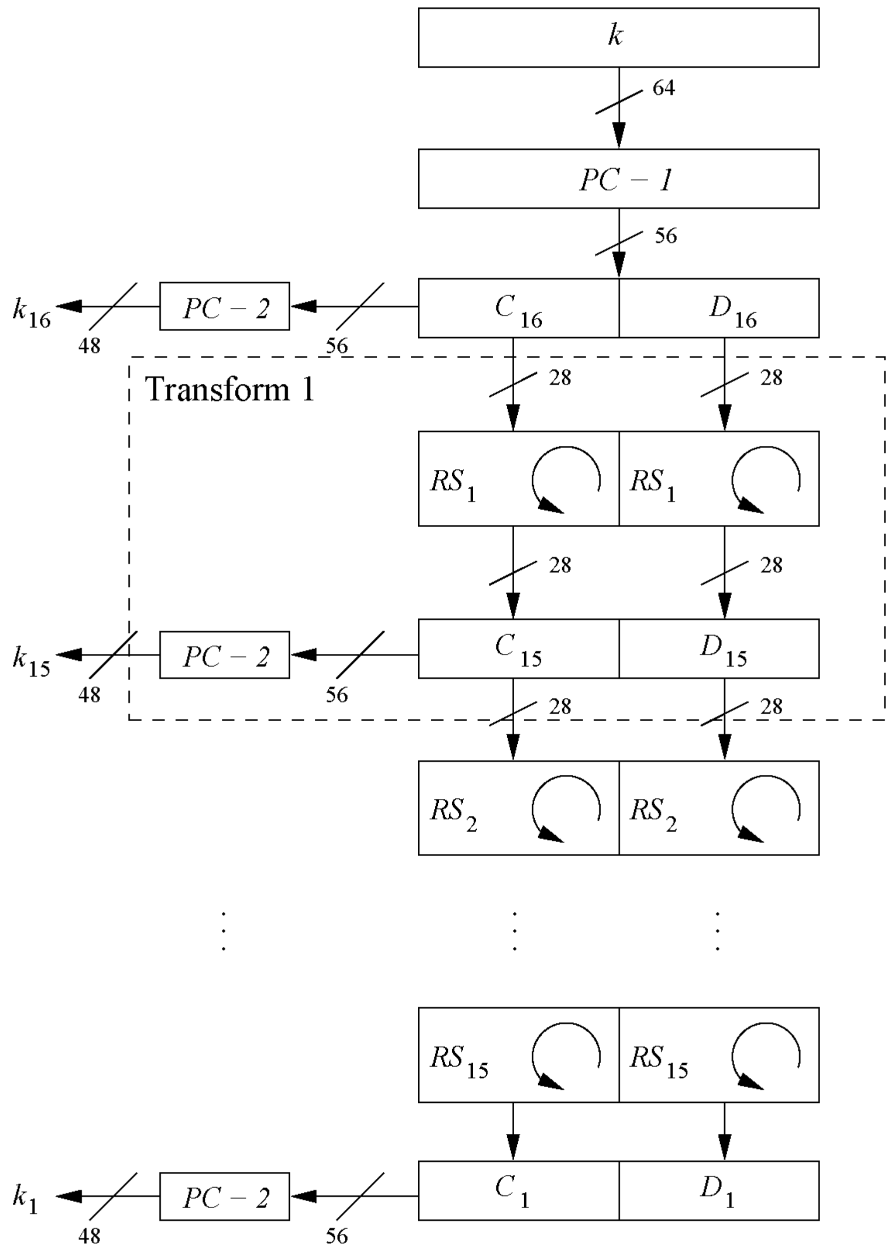


# Content of this Chapter

- Introduction to DES
- Overview of the DES Algorithm
- Internal Structure of DES
- **Decryption**
- Security of DES

## Decryption

- In **Feistel ciphers** only the keyschedule has to be modified for decryption.
- Generate the same 16 round keys in reverse order.
  (for a detailed discussion on why this works see *Understanding Crptography* Chapter 3)
- **Reversed key schedule:**
  As $D_0=D_{16}$ and $C_0=C_{16}$ the first round key can be generated by applying *PC-2* right after *PC-1* (no rotation here!).
  All other rotations of $C$ and $D$ can be reversed to reproduce the other round keys resulting in:
  - No rotation in round 1.
  - One bit rotation **to the right** in rounds 2, 9 and 16.
  - Two bit rotations **to the right** in all other rounds.






Chapter 3 of *Understanding Cryptography* by Christof Paar and Jan Pelzl

# Content of this Chapter

- Introduction to DES
- Overview of the DES Algorithm
- Internal Structure of DES
- Decryption
- **Security of DES**

## Security of DES

- **After proposal of DES two major criticisms arose:**
  1. Key space is too small ($2^{56}$ keys)
  2. S-box design criteria have been kept secret: Are there any hidden analytical attacks (*backdoors*), only known to the NSA?

- **Analytical Attacks:** DES is highly resistent to both *differential* and *linear cryptanalysis*, which have been published years later than the DES. This means IBM and NSA had been aware of these attacks for 15 years!
  So far there is no known analytical attack which breaks DES in realistic scenarios.

- **Exhaustive key search:** For a given pair of plaintext-ciphertext ($x$, $y$) test all $2^{56}$ keys until the condition $DES_k^{-1}(x)=y$ is fulfilled.
  $\Rightarrow$ Relatively easy given today's computer technology!

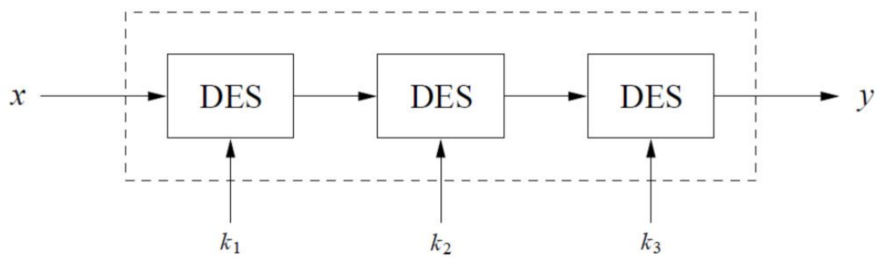
## History of Attacks on DES

[redacted]

# Triple DES – 3DES

- Triple encryption using DES is often used in practice to extend the effective key length of DES to 112. For more info on multiple encryption and effective key lengths see Chapter 5 of *Understanding Cryptography.*

$$y = DES_{k_3}(DES_{k_2}(DES_{k_1}(x)))$$




- Alternative version of *3DES:* $y = DES_{k_3}(DES^{-1}_{k_2}(DES_{k_1}(x))).$

  Advantage: choosing $k_1=k_2=k_3$ performs single DES encryption.
- No practical attack known today.
- Used in many legacy applications, i.e., in banking systems.

## Alternatives to DES

| Algorithm | I/O Bit | key lengths | remarks |
|---|---|---|---|
| AES / Rijndael | 128 | 128/192/256 | DES "replacement",<br>worldwide used standard |
| Triple DES | 64 | 112 (effective) | conservative choice |
| Mars | 128 | 128/192/256 | AES finalist |
| RC6 | 128 | 128/192/256 | AES finalist |
| Serpent | 128 | 128/192/256 | AES finalist |
| Twofish | 128 | 128/192/256 | AES finalist |
| IDEA | 64 | 128 | (Patented till 2011) |

## Lessons Learned

- DES was the dominant symmetric encryption algorithm from the mid-1970s to the mid-1990s. Since 56-bit keys are no longer secure, the Advanced Encryption Standard (AES) was created.
- Standard DES with 56-bit key length can be broken relatively easily nowadays through an exhaustive key search.
- DES is quite robust against known analytical attacks: In practice it is very difficult to break the cipher with differential or linear cryptanalysis.
- By encrypting with DES three times in a row, triple DES (3DES) is created, against which no practical attack is currently known.
- The "default" symmetric cipher is nowadays often AES. In addition, the other four AES finalist ciphers all seem very secure and efficient.

# Content of this Chapter

- **Overview of the AES algorithm**
- Internal structure of AES
  - Byte Substitution layer
  - Diffusion layer
  - Key Addition layer
  - Key schedule
- Decryption
- Practical issues

## Some Basic Facts

- AES is the most widely used symmetric cipher today

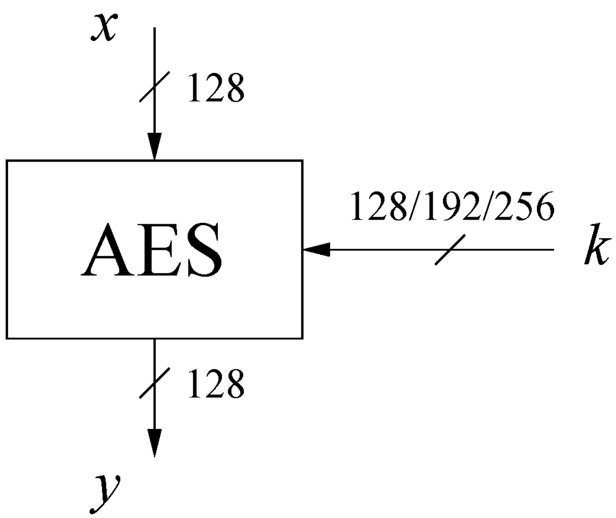
- The algorithm for AES was chosen by the US *National Institute of Standards and Technology* (NIST) in a multi-year selection process

- The requirements for all AES candidate submissions were:
  - Block cipher with **128-bit block size**
  - **Three supported key lengths**: 128, 192 and 256 bit
  - Security relative to other submitted algorithms
  - **Efficiency** in software and hardware

## Chronology of the AES Selection

- The need for a new block cipher announced by NIST in January, 1997
- 15 candidates algorithms accepted in August, 1998
- 5 finalists announced in August, 1999:
  - *Mars* – IBM Corporation
  - *RC6* – *RSA* Laboratories
  - *Rijndael* – J. Daemen & V. Rijmen
  - *Serpent* – Eli Biham et al.
  - *Twofish* – B. Schneier et al.
- In October 2000, *Rijndael* was chosen as the AES
- AES was formally approved as a US federal standard in November 2001

## AES: Overview

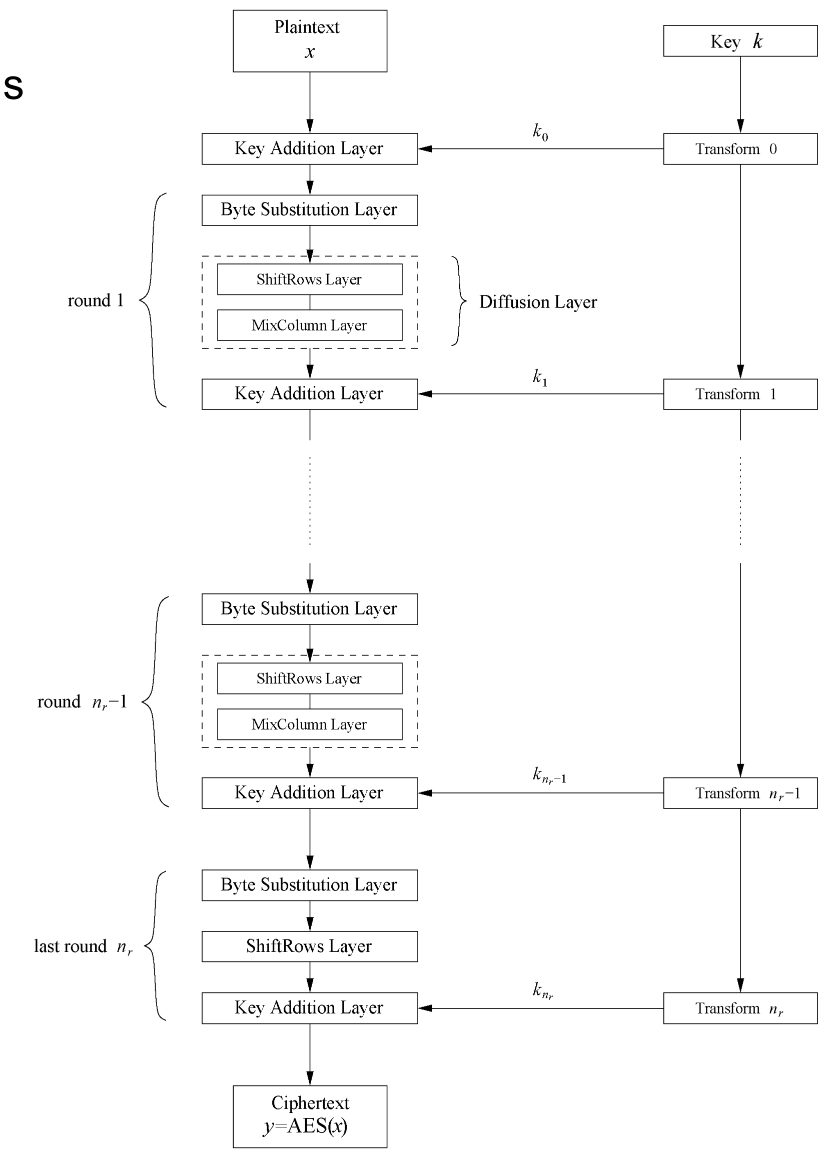



The number of rounds depends on the chosen key length:

| Key length (bits) | Number of rounds |
|---|---|
| 128 | 10 |
| 192 | 12 |
| 256 | 14 |

## AES: Overview

- Iterated cipher with 10/12/14 rounds
- Each round consists of "Layers"



# Content of this Chapter

- Overview of the AES algorithm
- **Internal structure of AES**
  - Byte Substitution layer
  - Diffusion layer
  - Key Addition layer
  - Key schedule
- Decryption
- Practical issues

## Internal Structure of AES

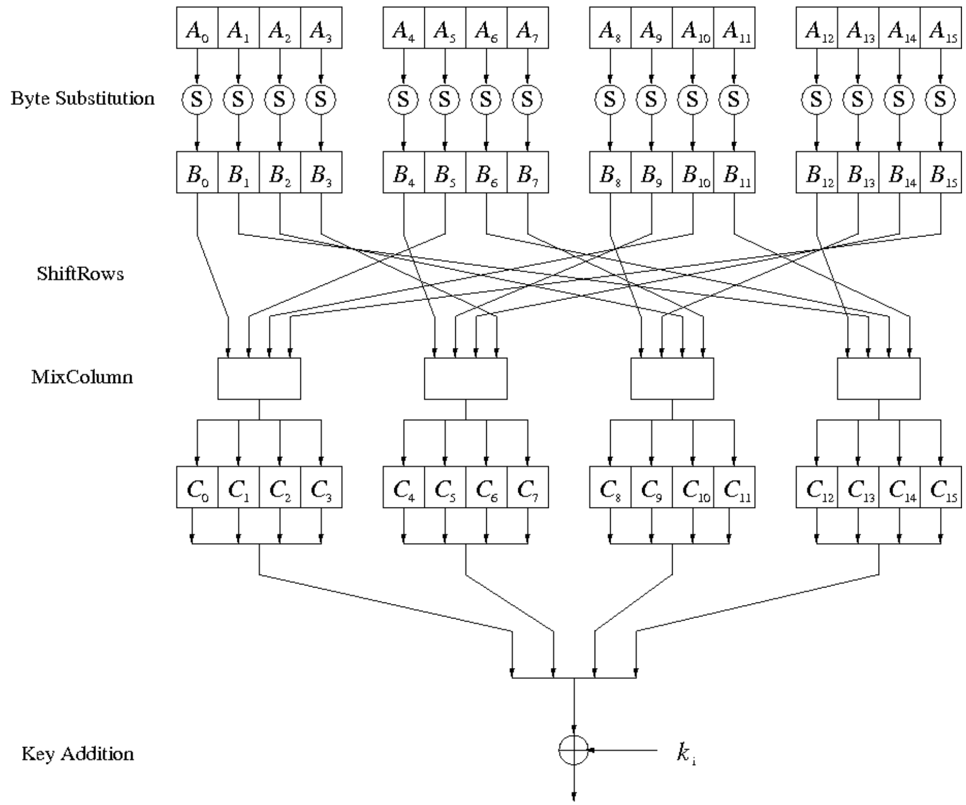
- AES is a byte-oriented cipher
- The state $A$ (i.e., the 128-bit data path) can be arranged in a 4x4 matrix:

| | | | |
|---|---|---|---|
| $A_0$ | $A_4$ | $A_8$ | $A_{12}$ |
| $A_1$ | $A_5$ | $A_9$ | $A_{13}$ |
| $A_2$ | $A_6$ | $A_{10}$ | $A_{14}$ |
| $A_3$ | $A_7$ | $A_{11}$ | $A_{15}$ |

with $A_0, \ldots, A_{15}$ denoting the 16-byte input of AES

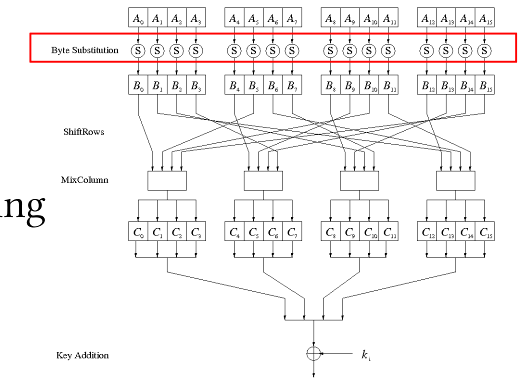
# Internal Structure of AES

- Round function for rounds 1,2,…,$n_{r-1}$:




- Note: In the last round, the MixColumn tansformation is omitted

## Byte Substitution Layer



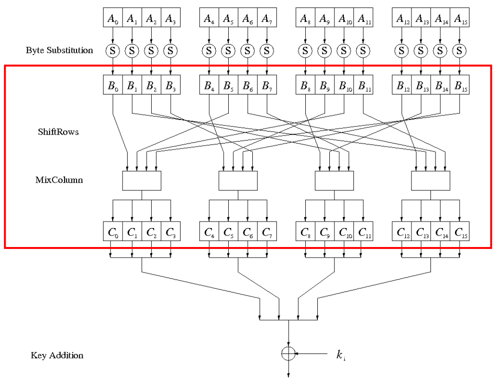

- The Byte Substitution layer consists of 16 **S-Boxes** with the following properties:

  The S-Boxes are

  - **identical**
  - the only **nonlinear** elements of AES, i.e.,
    $\text{ByteSub}(A_i) + \text{ByteSub}(A_j) \neq \text{ByteSub}(A_i + A_j)$, for $i,j = 0,\ldots,15$
  - **bijective**, i.e., there exists a one-to-one mapping of input and output bytes
    $\Rightarrow$ S-Box can be uniquely reversed

- In software implementations, the S-Box is usually realized as a lookup table

# Diffusion Layer




The Diffusion layer

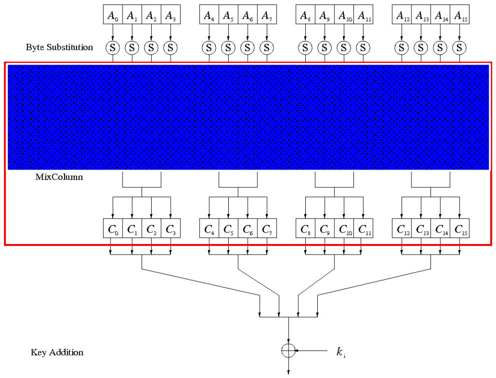
- provides diffusion over all input state bits

- consists of two sublayers:
  - **ShiftRows Sublayer**: Permutation of the data on a byte level
  - **MixColumn Sublayer**: Matrix operation which combines ("mixes") blocks of four bytes

- performs a linear operation on state matrices $A$, $B$, i.e.,

$$\mathrm{DIFF}(A) + \mathrm{DIFF}(B) = \mathrm{DIFF}(A + B)$$

## ShiftRows Sublayer




- Rows of the state matrix are shifted cyclically:

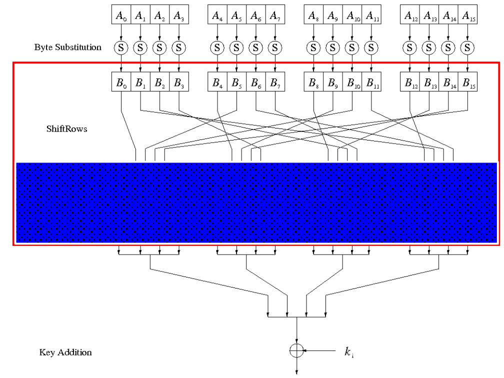
Input matrix

| | | | |
|---|---|---|---|
| $B_0$ | $B_4$ | $B_8$ | $B_{12}$ |
| $B_1$ | $B_5$ | $B_9$ | $B_{13}$ |
| $B_2$ | $B_6$ | $B_{10}$ | $B_{14}$ |
| $B_3$ | $B_7$ | $B_{11}$ | $B_{15}$ |

Output matrix

| | | | | |
|---|---|---|---|---|
| $B_0$ | $B_4$ | $B_8$ | $B_{12}$ | no shift |
| $B_5$ | $B_9$ | $B_{13}$ | $B_1$ | ← one position left shift |
| $B_{10}$ | $B_{14}$ | $B_2$ | $B_6$ | ← two positions left shift |
| $B_{15}$ | $B_3$ | $B_7$ | $B_{11}$ | ← three positions left shift |

## MixColumn Sublayer




- Linear transformation which mixes each column of the state matrix

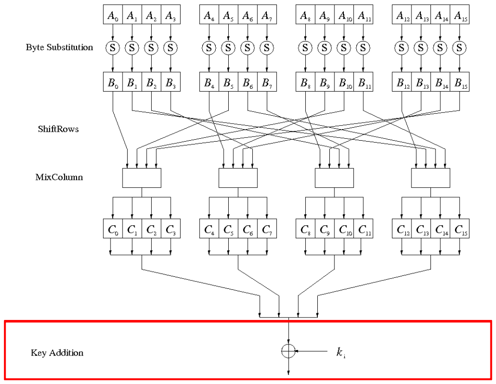
- Each 4-byte column is considered as a vector and multiplied by a fixed 4x4 matrix, e.g.,

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{pmatrix}$$

  where 01, 02 and 03 are given in hexadecimal notation

- All arithmetic is done in the Galois field $GF(2^8)$ (for more information see Chapter 4.3 in *Understanding Cryptography*)

## Key Addition Layer




- Inputs:
    - 16-byte state matrix $C$
    - 16-byte subkey $k_i$

- Output: $C \oplus k_i$

- The subkeys are generated in the key schedule

## Key Schedule

- Subkeys are derived recursively from the original 128/192/256-bit input key

- Each round has 1 subkey, plus 1 subkey at the beginning of AES

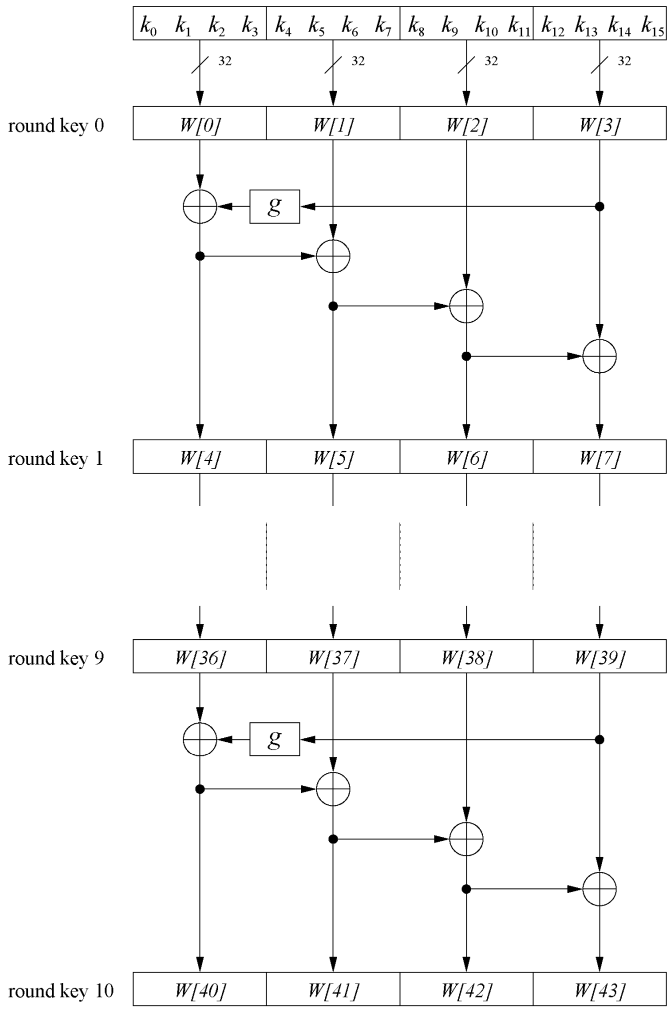
| Key length (bits) | Number of subkeys |
|---|---|
| 128 | 11 |
| 192 | 13 |
| 256 | 15 |

- Key whitening: Subkey is used both at the input and output of AES
  $\Rightarrow$ # subkeys = # rounds + 1

- There are different key schedules for the different key sizes

## Key Schedule

Example: Key schedule for 128-bit key AES




- Word-oriented: 1 word = 32 bits
- 11 subkeys are stored in *W[0]…W[3], W[4]…W[7], … , W[40]…W[43]*
- First subkey *W[0]…W[3]* is the original AES key

$$i = 1, \dots, 10, j = 1,2,3$$

$$W[4i] = W[4(i-1)] + g(W[4i-1])$$

$$W[4i+j] = W[4i+j-1] + W[4(i-1)+j]$$

## Key Schedule

- Function $g$ rotates its four input bytes and performs a bytewise S-Box substitution
  $\Rightarrow$ nonlinearity

- The round coefficient $RC$ is only added to the leftmost byte and varies from round to round:

$$RC[1] = x^0 = (00000001)_2$$
$$RC[2] = x^1 = (00000010)_2$$
$$RC[3] = x^2 = (00000100)_2$$
$$...$$
$$RC[10] = x^9 = (00110110)_2$$

- $x^i$ represents an element in a Galois field
  (again, cf. Chapter 4.3 of *Understanding Cryptography*)