CP 460 - Applied Cryptography

The Data Encryption Standard (DES)

Department of Physics and Computer Science
Faculty of Science, Waterloo

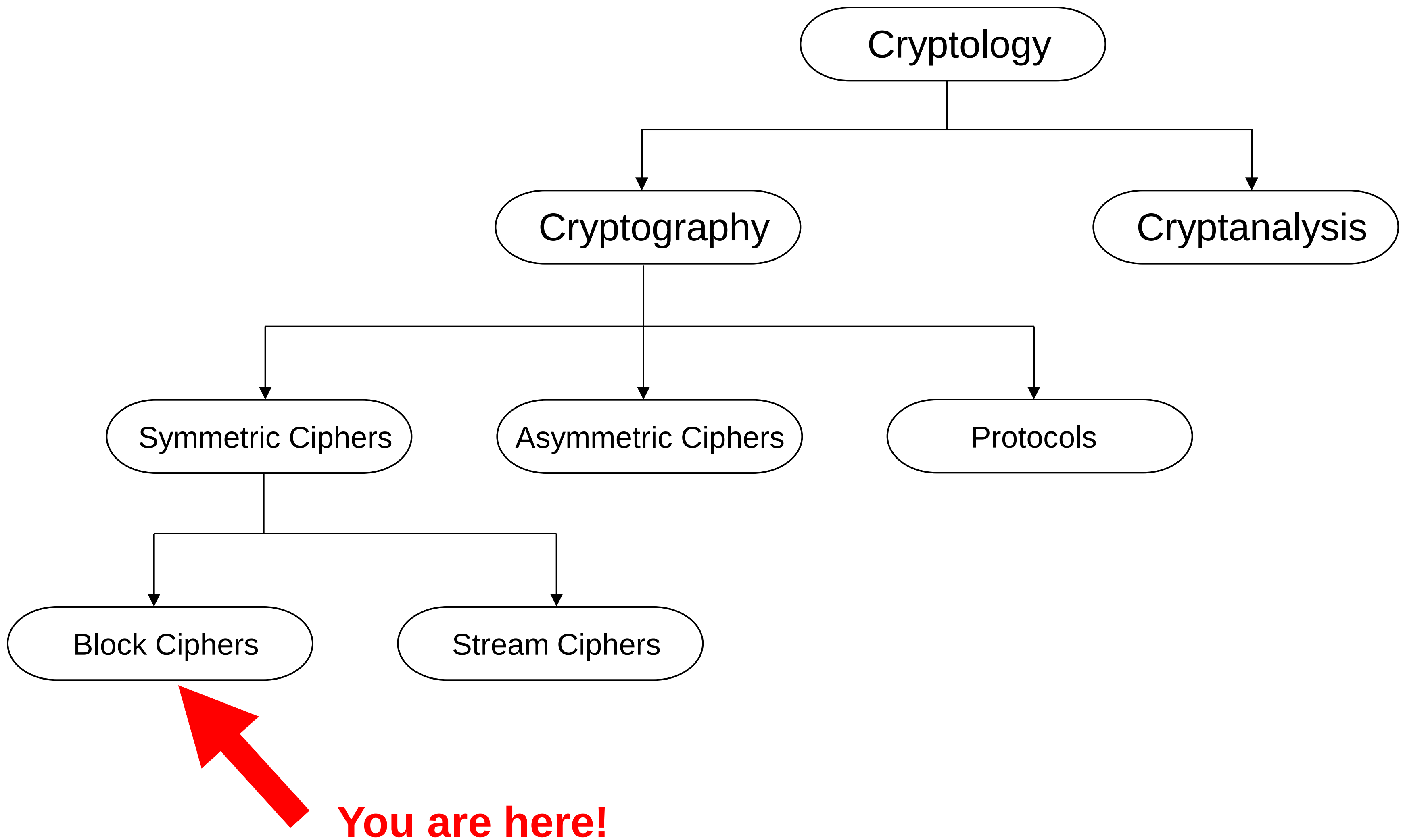
Abbas Yazdinejad, Ph.D.

Fall 2024

Content of this Chapter

Content of this Chapter

Classification of DES in the Field of Cryptology



You are here!

DES Facts

- Data Encryption Standard (DES) encrypts **blocks of size 64 bit**.
- Developed by **IBM** based on the cipher *Lucifer* under influence of the *National Security Agency* (NSA), the design criteria for DES have not been published
- **Standardized 1977** by the **National Bureau of Standards** (NBS) today called *National Institute of Standards and Technology* (NIST)
- Most popular **block cipher** for most of the last 30 years.
- By far best studied symmetric algorithm.
- Nowadays considered insecure due to the small **key length of 56 bit.**
- **But: 3DES yields very secure cipher**, still widely used today.
- Replaced by the *Advanced Encryption Standard* (**AES**) in 2000

- For a more detailed history see Chapter 3.1 in *Understanding Cryptography*

Block Cipher Primitives: Confusion and Diffusion

- Claude Shannon: There are two primitive operations with which strong encryption algorithms can be built:

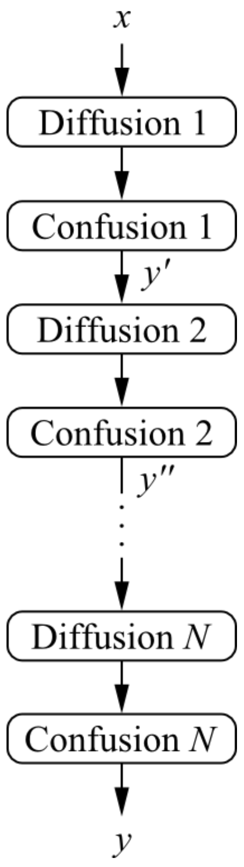
  1. **Confusion:** An encryption operation where the **relationship between key and ciphertext is obscured**. Today, a common element for achieving confusion is **substitution**, which is found in both AES and DES.
     Example: If "A" in plaintext is replaced by "Q" in ciphertext, it's hard to directly relate "A" to "Q" without knowing the key.
  2. **Diffusion:** An encryption operation where the **influence of one plaintext symbol is spread over many ciphertext symbols** with the goal of hiding statistical properties of the plaintext. A simple diffusion element is the **bit permutation**, which is frequently used within DES.

     Example: Changing one letter in the plaintext (e.g., "A" to "B") might change several letters in the ciphertext, making it harder to detect patterns.

- Both operations by themselves cannot provide security. The idea is to concatenate confusion and diffusion elements to build so called *product ciphers*.

Product Ciphers

is an encryption method that combines both confusion and diffusion techniques to create a more secure encryption algorithm. The idea is to repeatedly apply confusion (such as substitution) and diffusion (such as permutation or transposition) to make the cipher stronger.

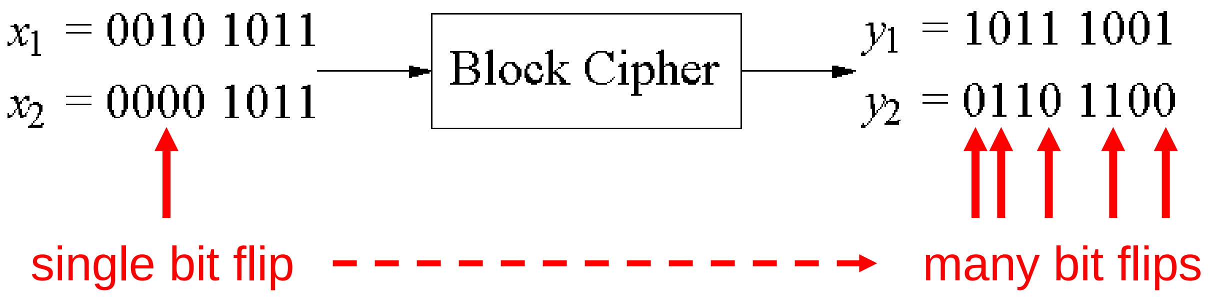


N round product cipher

- Most of today‘s block ciphers are *product ciphers* as they consist of rounds which are applied repeatedly to the data.

- Can reach excellent diffusion: **changing of one bit of plaintext results** *on average* in the **change of half the output bits**.

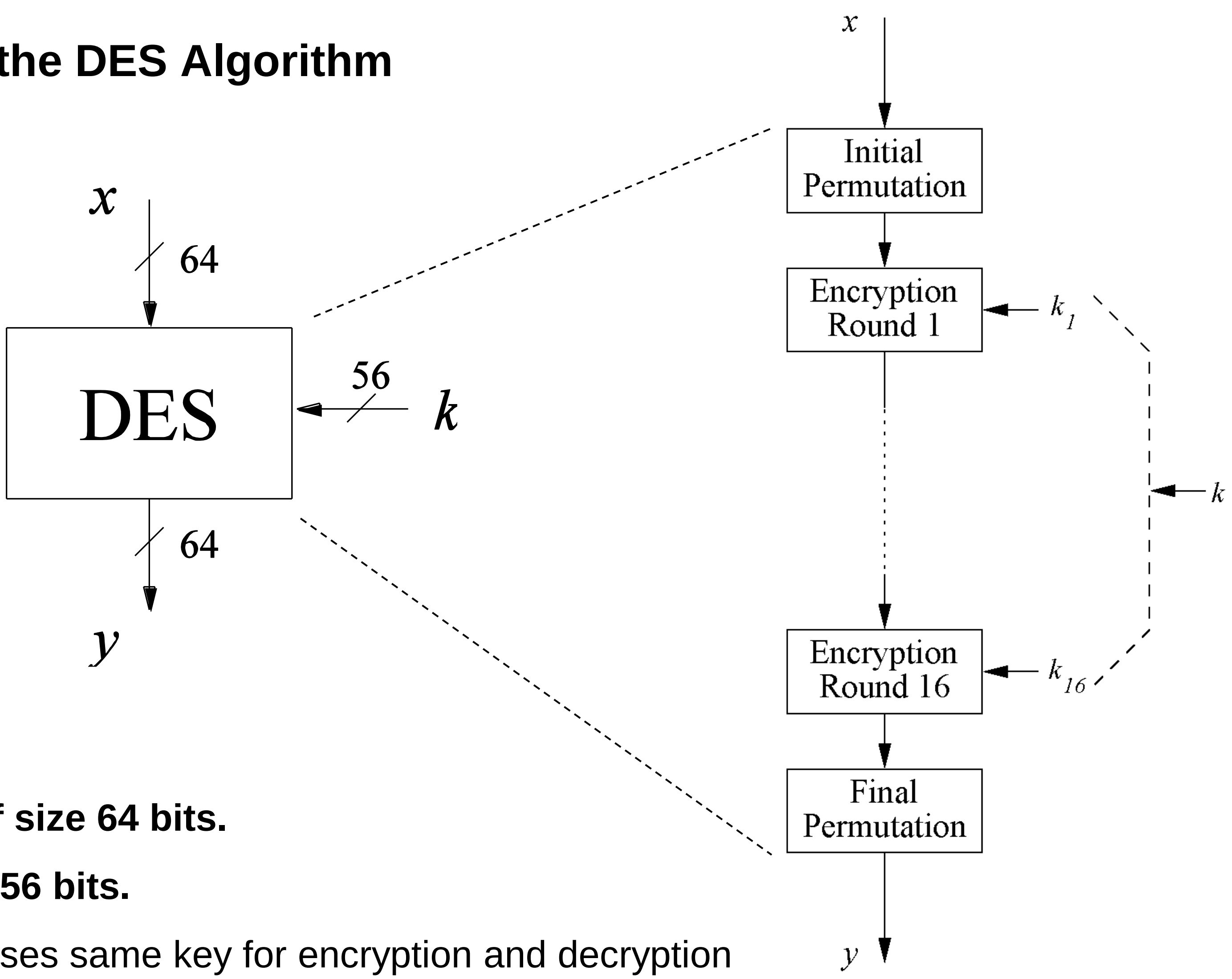
Example:



Content of this Chapter

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

Overview of the DES Algorithm

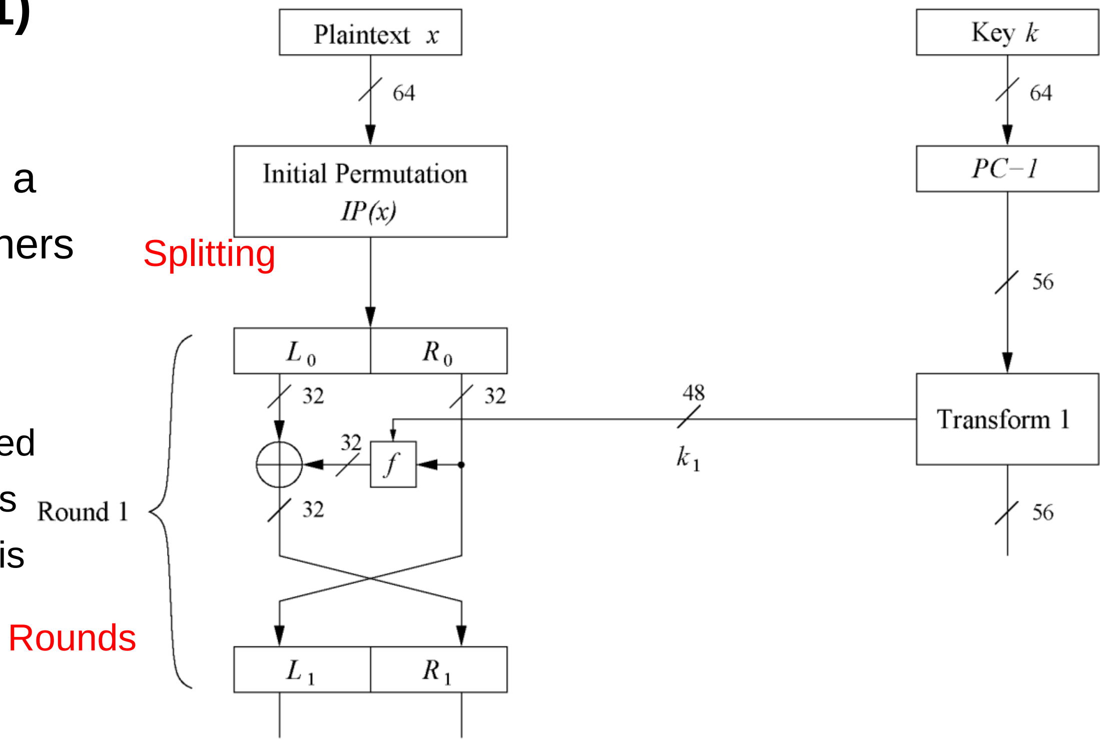


- **Encrypts blocks of size 64 bits.**
- **Uses a key of size 56 bits.**
- Symmetric cipher: uses same key for encryption and decryption
- Uses 16 rounds which all perform the identical operation
- Different subkey in each round derived from main key

The DES Feistel Network (1)

- DES structure is a *Feistel network,* is a symmetric structure for building block ciphers

- The Feistel Network splits the plaintext into two halves, then applies a series of operations (called rounds) on one half, while the other half remains unchanged, and then swaps the two halves. This process is repeated for multiple rounds.



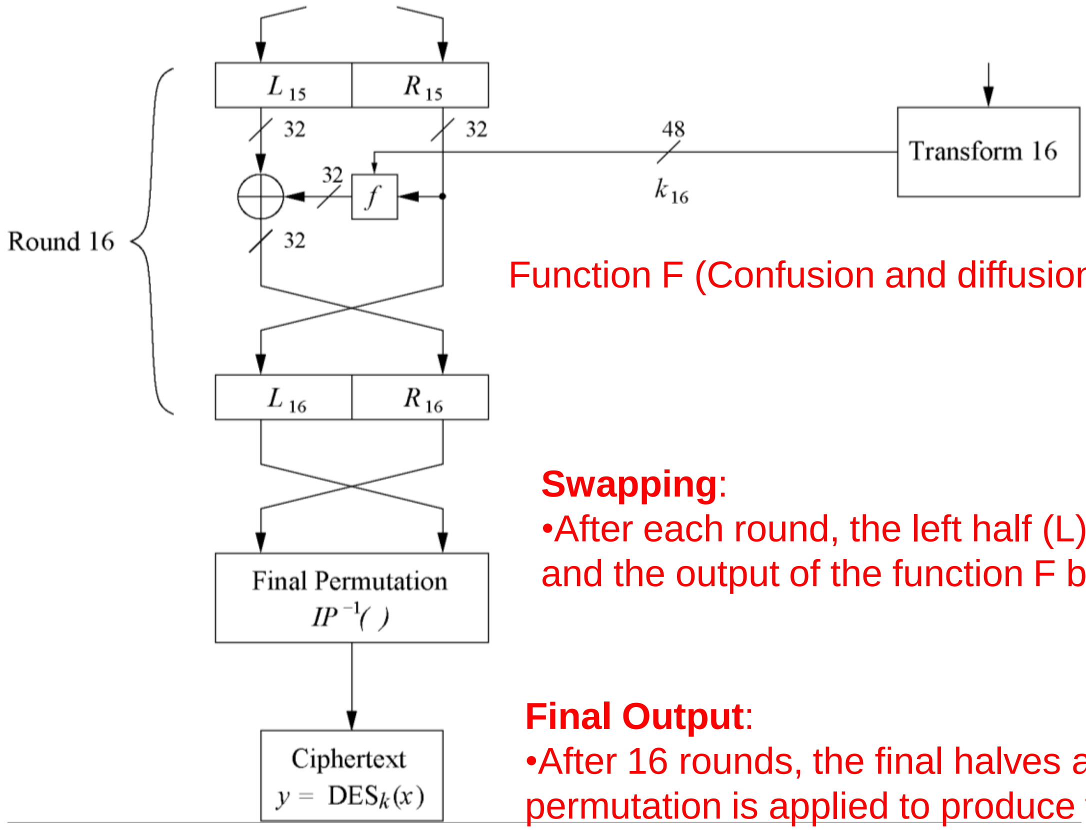
- Bitwise initial permutation, then 16 rounds
  1. Plaintext is split into 32-bit halves L_i and R_i
  2. R_i is fed into the function *f*, the output of which is then XORed with L_i
  3. Left and right half are swapped
- Rounds can be expressed as:

$$L_i = R_{i-1},$$

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

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



Function F (Confusion and diffusion take place)

Swapping:
•After each round, the left half (L) becomes the new right half (R), and the output of the function F becomes the new left half (L).

Final Output:
•After 16 rounds, the final halves are combined, and a final permutation is applied to produce the 64-bit ciphertext.

Short Example:

1. Initial Split: Plaintext 64-bit: L0 (32 bits) and R0 (32 bits).

2. First Round:L1 = R0R1 = L0 XOR F(R0, K1) (where K1 is the subkey for round 1).

3. Repeat for 16 Rounds: Each round swaps the halves and applies the function F with a new subkey.

4. Final Step: After 16 rounds, the halves are joined and a final permutation gives the ciphertext.

This Feistel structure allows encryption and decryption to be very similar, which is why DES can be easily reversed for decryption by simply reversing the order of the subkeys.

Content of this Chapter

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

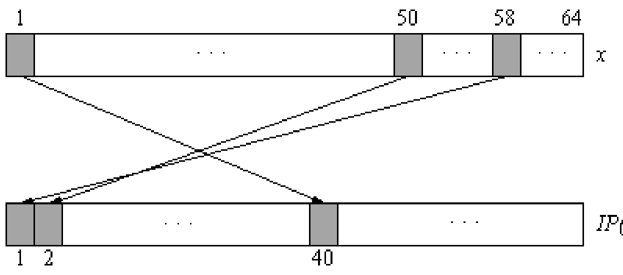
Initial and Final Permutation

The Initial Permutation (IP) takes the 64-bit plaintext and reorders the bits based on a fixed table. Each bit in the original plaintext is moved to a new position in the permuted output.For example, the bit at position 58 in the plaintext is moved to position 1 in the permuted output, the bit at position 50 is moved to position 2, and so on.

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

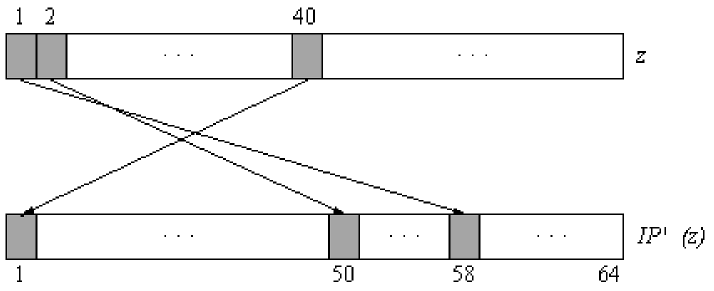
Initial Permutation

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

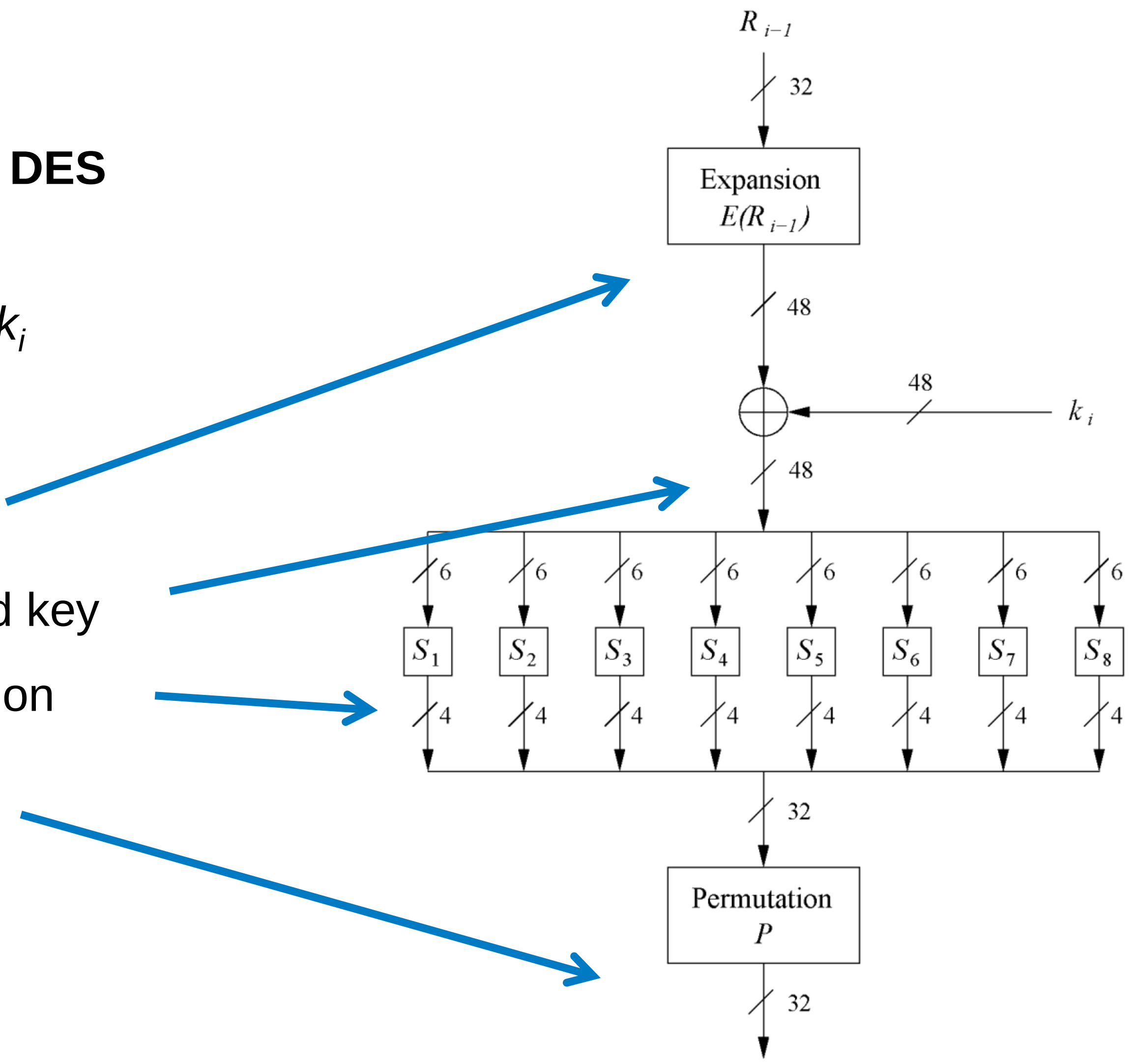
IP^{-1}							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25



In short, **IP** shuffles the bits to prepare the plaintext for the Feistel rounds in DES, while **IP⁻¹** restores the bit order after the encryption process.

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



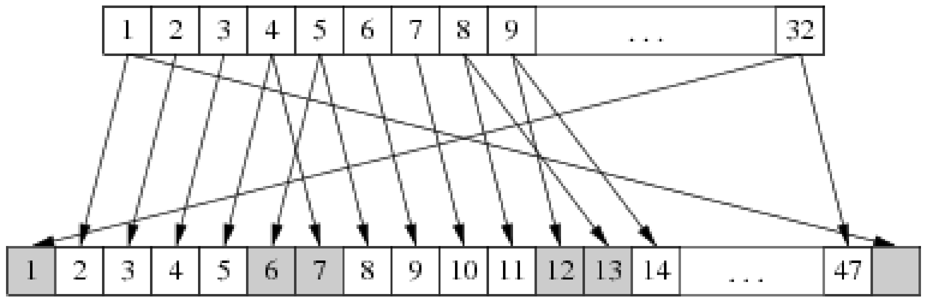
The **f-function** (or **Feistel function**) is a key component of the DES encryption process and plays a central role in ensuring both **confusion** and **diffusion** during each round of the Feistel network. The f-function takes a 32-bit input (the right half of the data) and a 48-bit round-specific subkey, processes them, and outputs a 32-bit result. This result is then XORed with the left half of the data in each DES round.

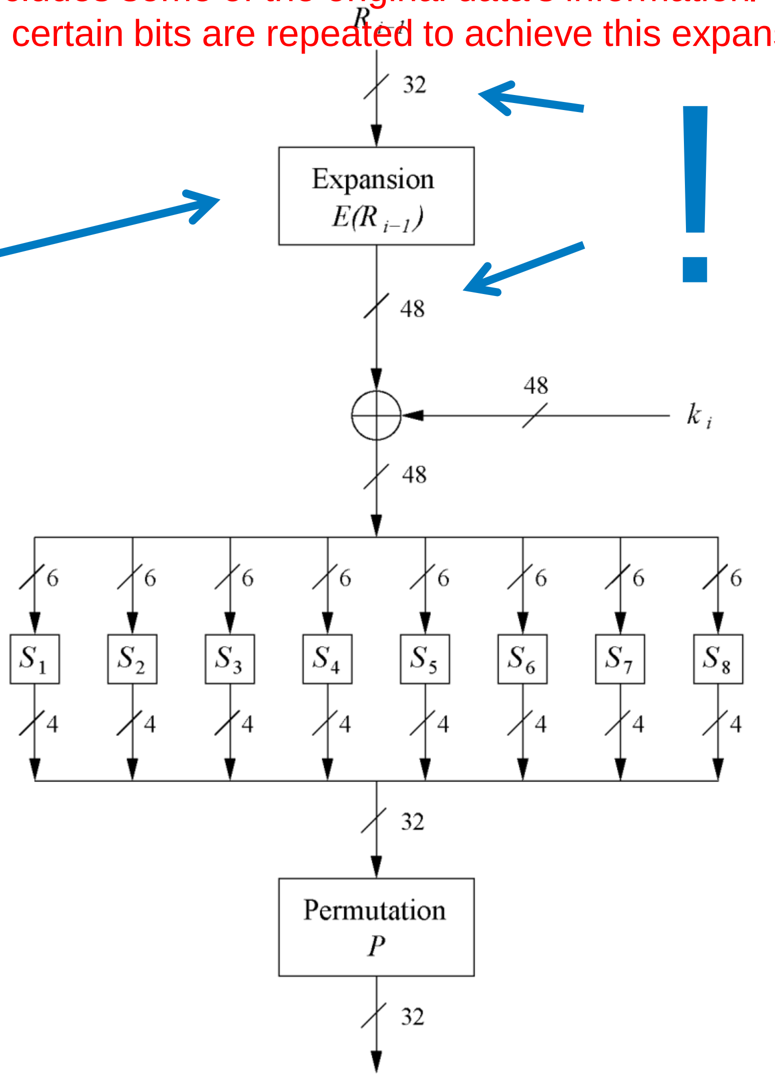
The Expansion Function E

The 32-bit right half of the data is expanded to 48 bits. This is done by duplicating some of the bits to increase the size from 32 to 48, ensuring that the expanded version includes some of the original data's information. The expansion follows a fixed pattern known as the E-box, where certain bits are repeated to achieve this expansion.

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

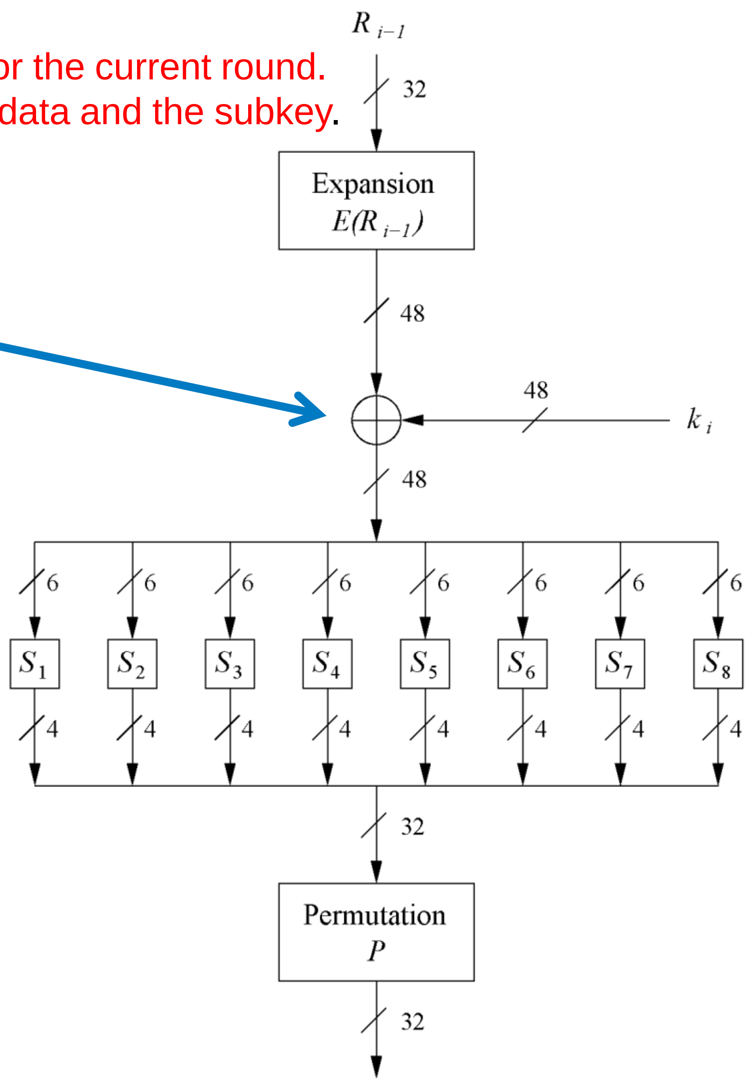
Key Mixing (XOR with Subkey):

•The expanded 48-bit data is XORed with the 48-bit subkey for the current round.

•This adds confusion, as the result now depends on both the data and the subkey.

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)**

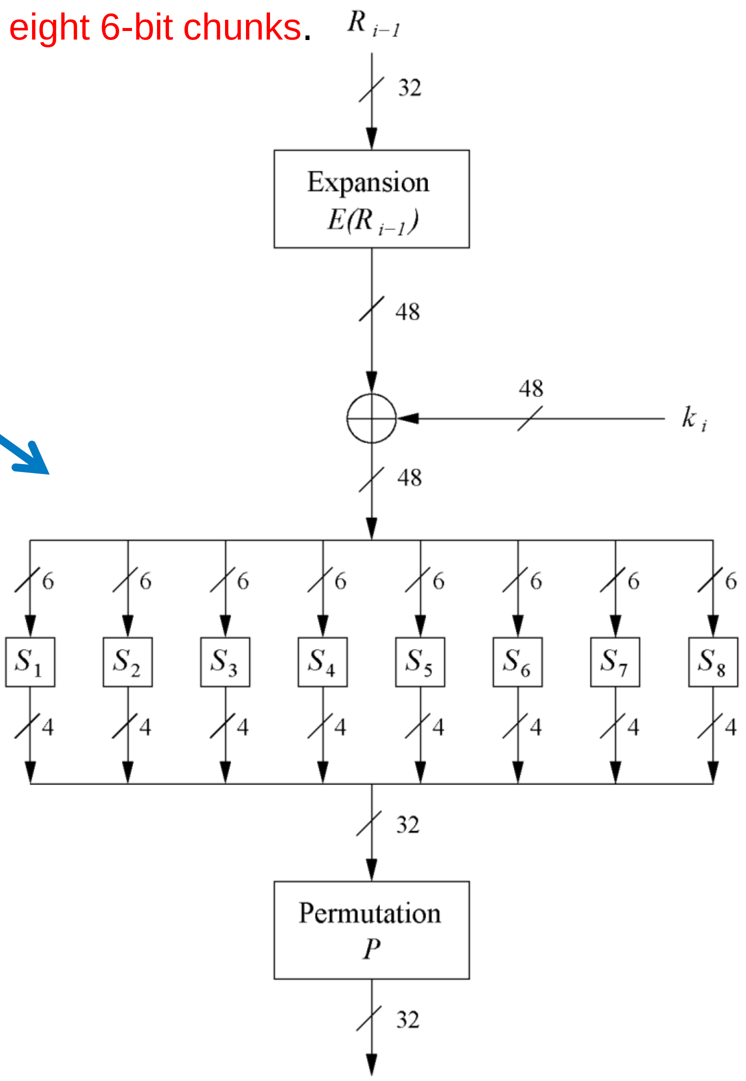


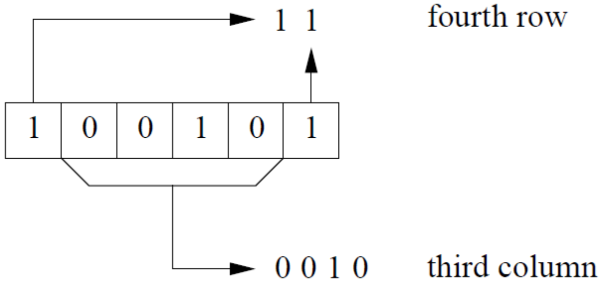
The DES S-Boxes

The 48-bit result from the XOR operation is then divided into eight 6-bit chunks.

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

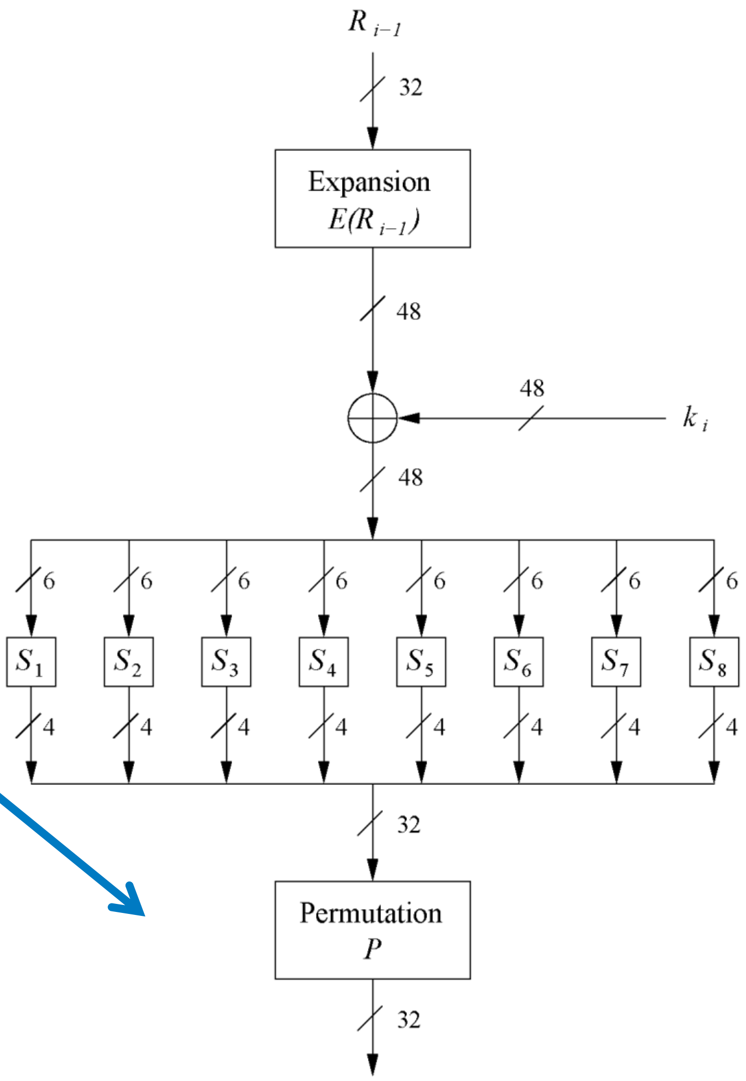
The Permutation P

After the substitution step, the 32-bit output undergoes a permutation (P-box). This step rearranges the bits according to a predefined pattern, further diffusing the data. The purpose of this is to spread the influence of each bit over multiple positions, contributing to the diffusion property.

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.

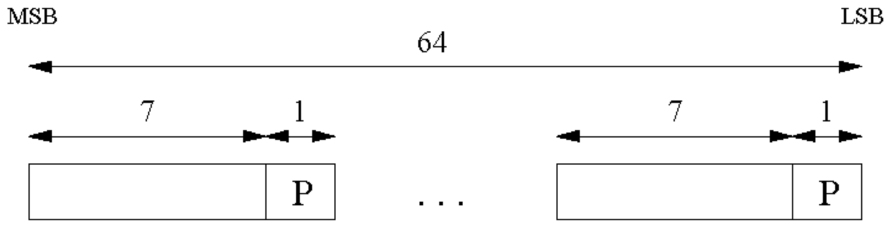
P							
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25



Key Schedule (1)

the **Key Schedule** refers to the process by which the initial 56-bit key is transformed into **16 different subkeys**, one for each round of the encryption process. Each subkey is 48 bits long and is derived from the original key using specific permutations and shifts. This key schedule ensures that different parts of the key are used in each round, making the cipher more secure.

- 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)

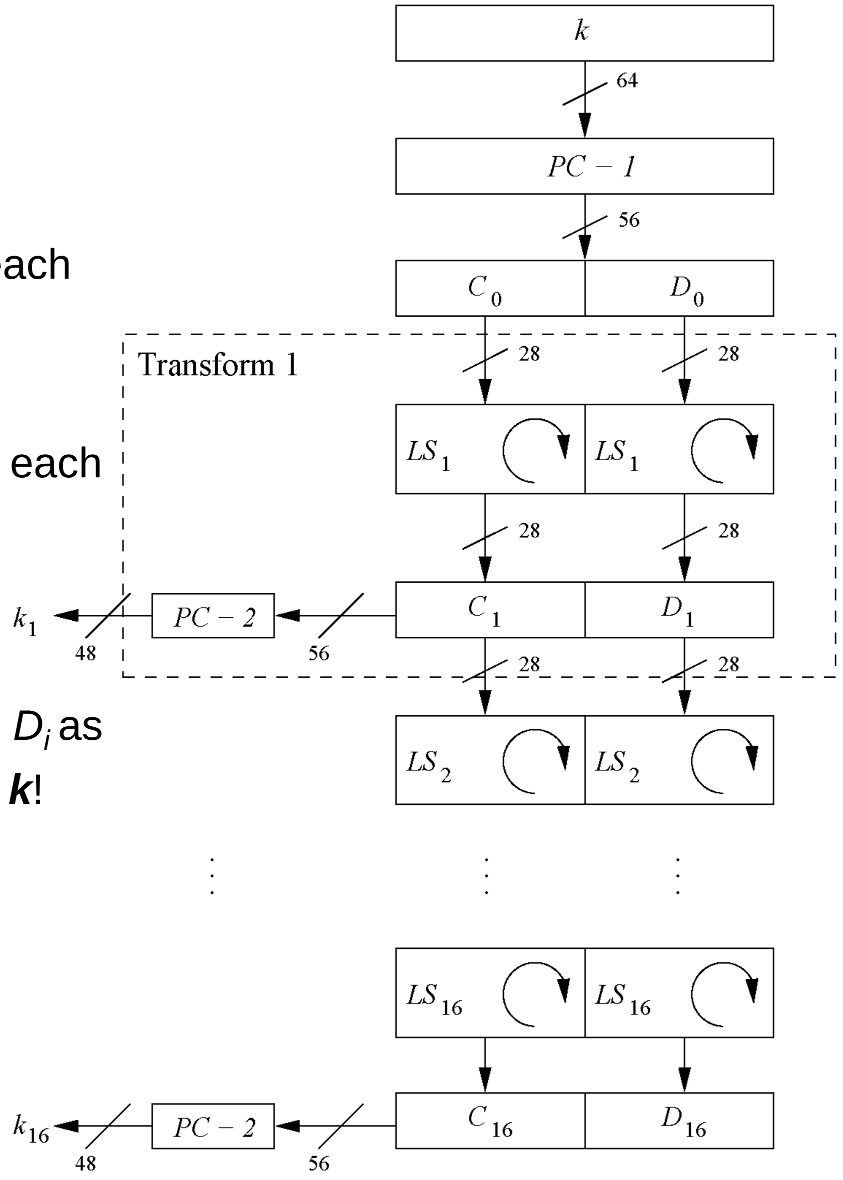
$PC-1$							
57	49	41	33	25	17	9	1
58	50	42	34	26	18	10	2
59	51	43	35	27	19	11	3
60	52	44	36	63	55	47	39
31	23	15	7	62	54	46	38
30	22	14	6	61	53	45	37
29	21	13	5	28	20	12	4

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

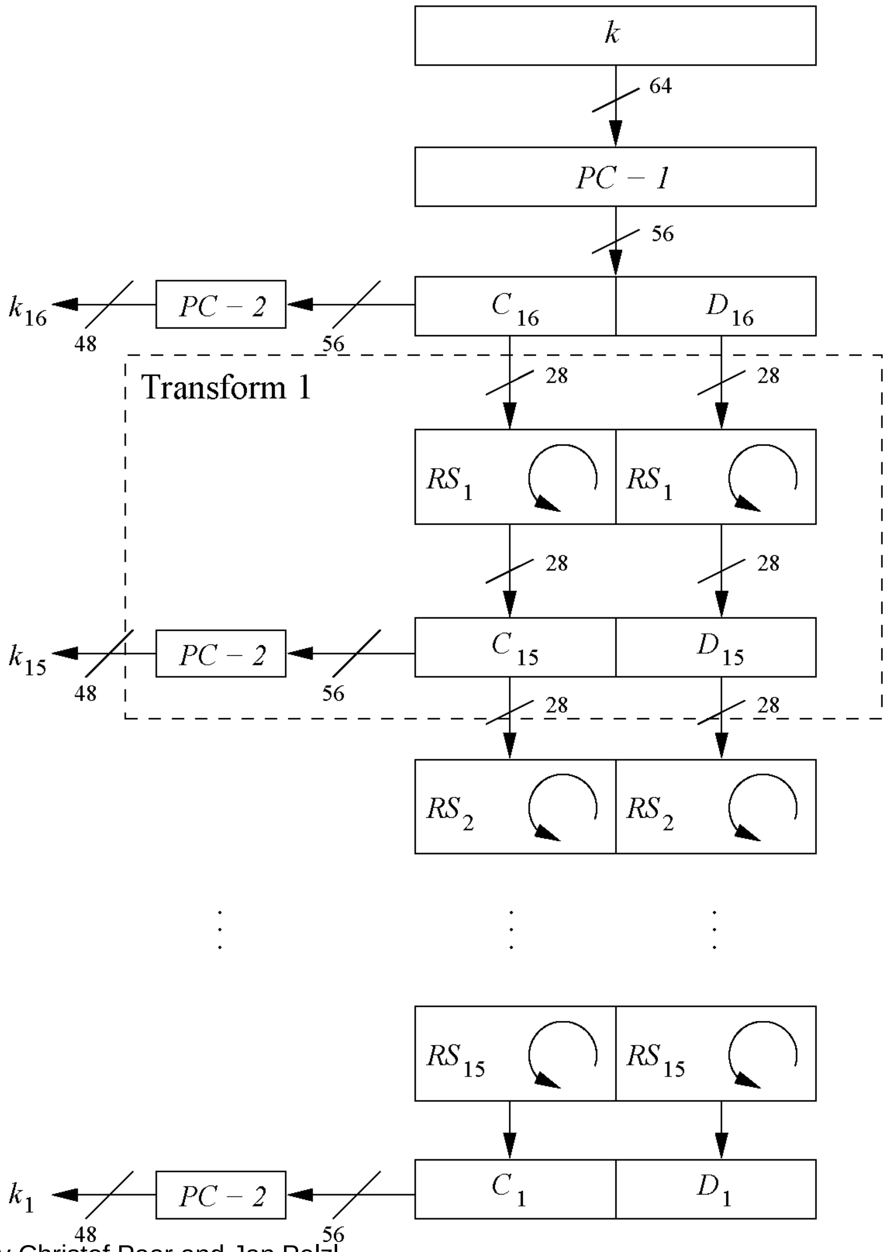
- Introduction to DES
- Overview of the DES Algorithm
- Internal Structure of DES
- **Decryption**
- Security of DES

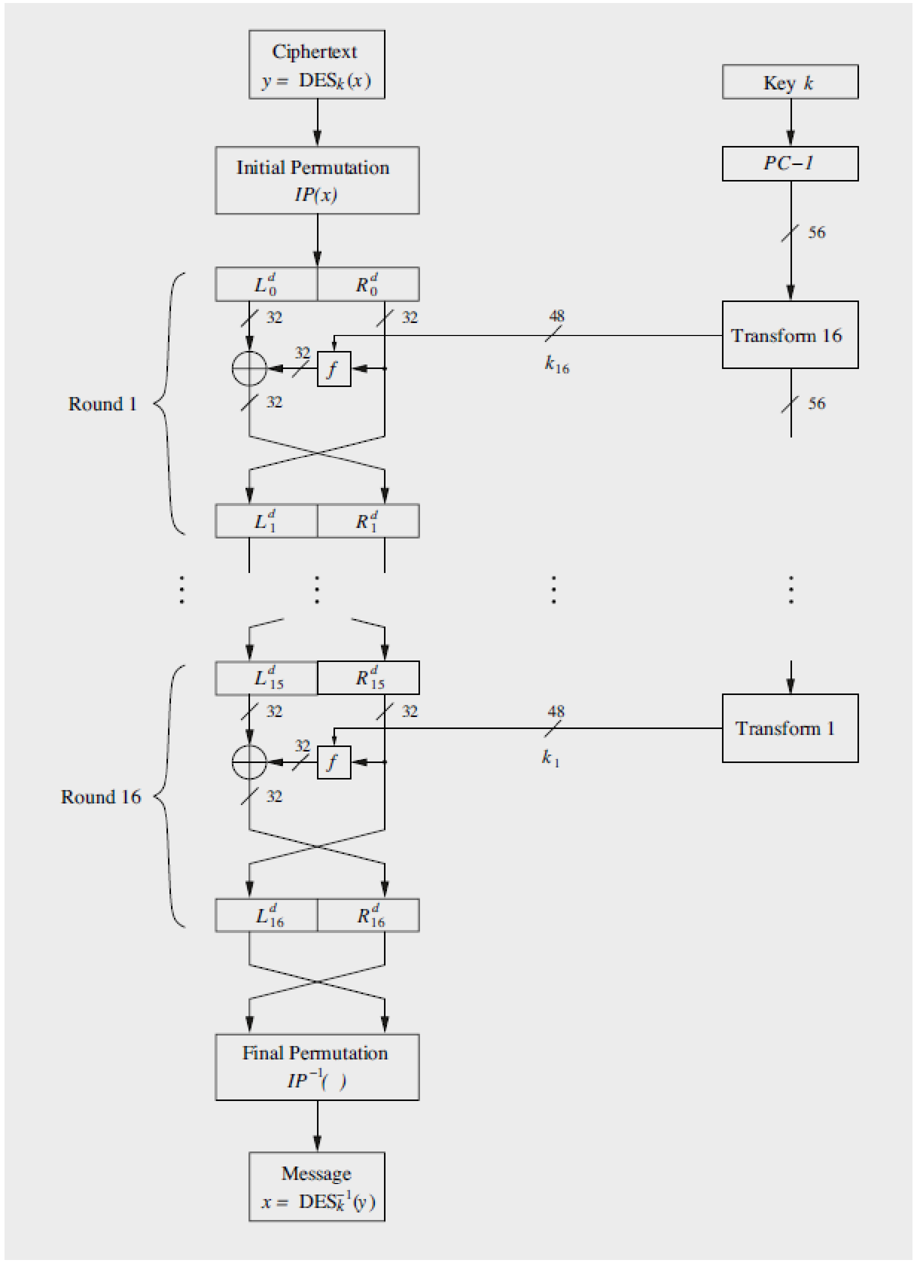
Decryption

One advantage of DES is that decryption is essentially the same function as encryption.

- 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.





DES decryption

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!

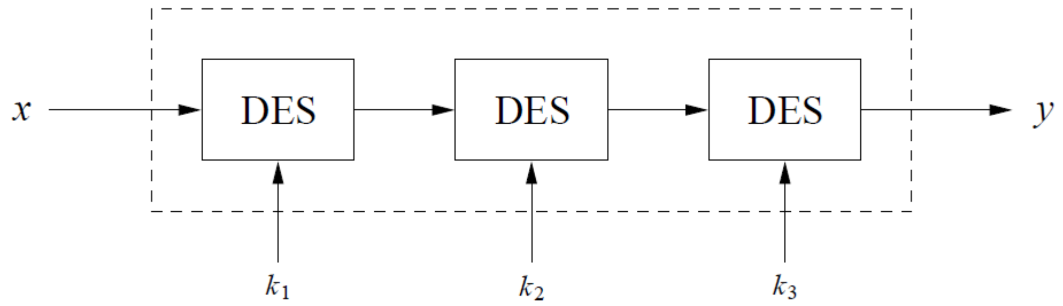
History of Attacks on DES

Year	Proposed/ implemented DES Attack
1977	Diffie & Hellman, (under-)estimate the costs of a key search machine
1990	Biham & Shamir propose differential cryptanalysis (2^{47} chosen ciphertexts)
1993	Mike Wiener proposes design of a very efficient key search machine: Average search requires 36h. Costs: $1.000.000
1993	Matsui proposes linear cryptanalysis (2^{43} chosen ciphertexts)
Jun. 1997	DES Challenge I broken, 4.5 months of distributed search
Feb. 1998	DES Challenge II--1 broken, 39 days (distributed search)
Jul. 1998	DES Challenge II--2 broken, key search machine *Deep Crack* built by the Electronic Frontier Foundation (EFF): 1800 ASICs with 24 search engines each, Costs: $250 000, 15 days average search time (required 56h for the Challenge)
Jan. 1999	DES Challenge III broken in 22h 15min (distributed search assisted by *Deep Crack*)
2006-2008	Reconfigurable key search machine *COPACOBANA* developed at the Universities in Bochum and Kiel (Germany), uses 120 FPGAs to break DES

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", 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)

AES (Advanced Encryption Standard)

•**Pros**:

- Strong security and resistance to attacks.
- Fast in both hardware and software.
- Widely adopted as a global standard.

•**Cons**:

- May be slower than simpler algorithms for smaller devices with limited resources.

Triple DES (3DES)

•**Pros**:

- More secure than DES due to applying DES three times.
- Easy transition for systems already using DES.

•**Cons**:

- Slow and inefficient, especially with small block size (64 bits).
- Less secure than modern algorithms like AES.

MARS

•**Pros**:
- High security and efficient in hardware.
- Supports a wide range of key lengths (up to 448 bits).

•**Cons**:
- Complex design makes it harder to implement.
- Less widely used compared to AES.

RC6

•**Pros**:
- Fast and efficient in software, especially on processors with multiple instructions.
- Simple and flexible.

•**Cons**:
- Not chosen as the AES standard, less popular.
- More complex than AES due to additional operations.

Serpent

•**Pros**:
- Very secure, designed with a focus on cryptographic strength.
- Resistant to known cryptanalytic attacks.

•**Cons**:
- Slower than AES due to more rounds (32 rounds vs. AES's 10).
- Less adopted compared to AES.

Twofish

•**Pros**:
- Fast and flexible, efficient in hardware and software.
- Strong security and a good alternative to AES.

•**Cons**:
- Less widely adopted than AES.
- More complex key schedule compared to AES.

IDEA (International Data Encryption Algorithm)

•**Pros**:
- Strong security, widely used in earlier encryption tools like PGP.
- Simple design.

•**Cons**:
- Small block size (64 bits), less efficient for modern large-scale data.
- Less widely used in modern applications.

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.

Thank
You