الحاسبات

4th Class

Computers & Data Security

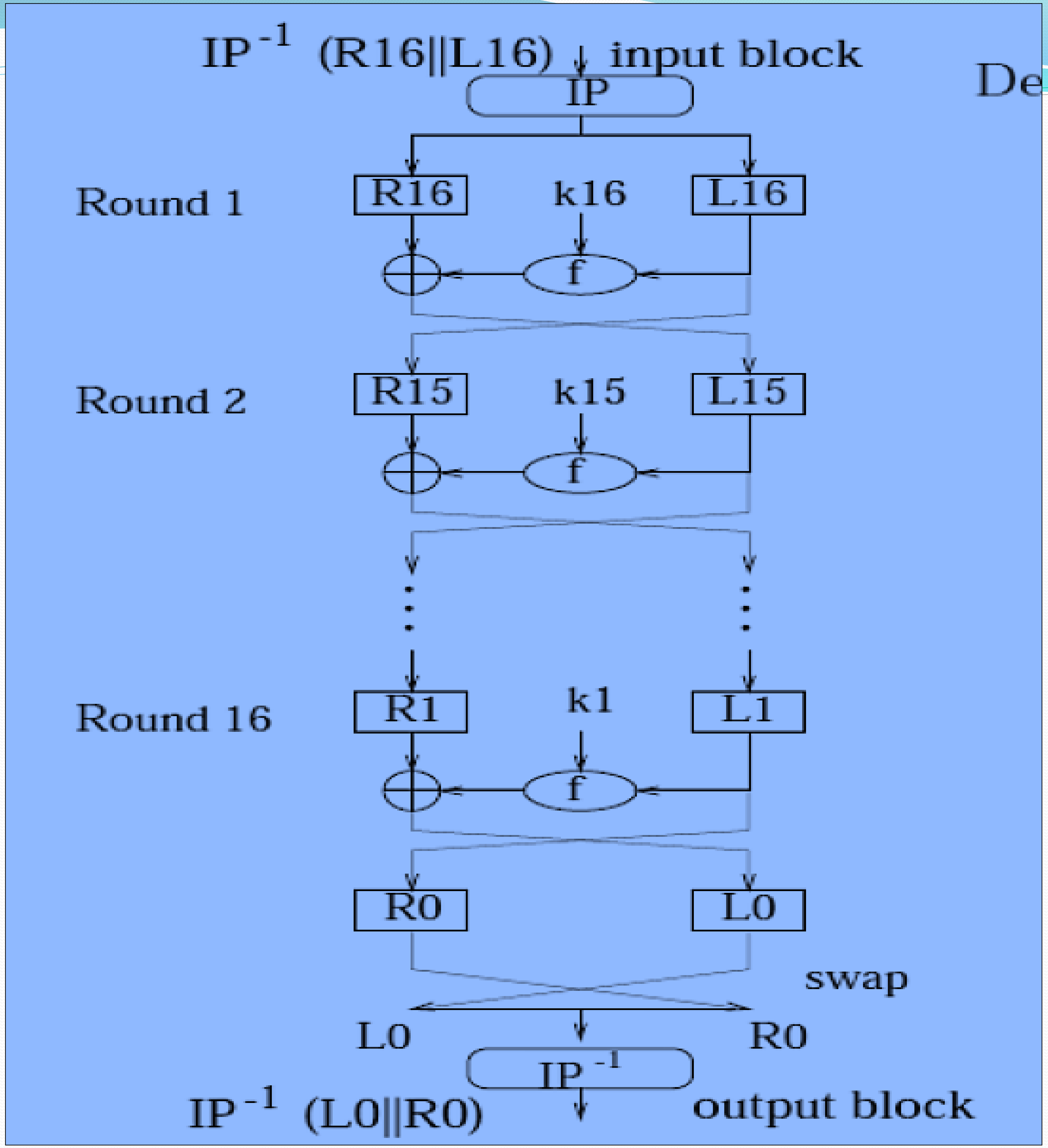
أمنية الحاسوب والبيانات

أستاذ المادة

م . د . اخلاص عباس البحراني

Decrypting DES

- same function to encrypt or decrypt a block.
- The only difference is that the keys must be used in the reverse order. That is, if the encryption keys for each round are K1, K2, K3, . . . , K16, then the decryption keys are K16, K15, K14, . . . , K1,.
- The algorithm that generates the key used for each round is circular as well.
- The key shift is a right shift and the number of positions shifted is 0, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1.


IP -1 (R16||L16)
input block
IP
De
Round 1
R16
k16
L16
f
Round 2
R15
k15
L15
f
Round 16
R1
k1
L1
f
R0
L0
swap
L0
R0
IP -1
IP -1 (L0||R0)
output block

Block vs. Stream Ciphers

Block cipher	Stream cipher
process messages in into blocks, each of which is then en/decrypted	process messages a bit or byte at a time when en/decrypting
Error propagation	Low error propagation
Slowness	Speed of transformation
High Diffusion	Low diffusion
Immunity to insertions	Susceptibility to attacks on integrity

Public-Key Cryptography

- public-key/two-key/asymmetric cryptography involves the use of two keys:
  - a public-key, which may be known by anybody, and can be used to encrypt messages, and verify signatures
  - a private-key, known only to the recipient, used to decrypt messages, and sign (create) signatures
- is asymmetric because those who encrypt messages or verify signatures cannot decrypt messages or create signatures

(a) Encryption

Public-Key Characteristics: -

- it is computationally infeasible to find decryption key knowing only algorithm & encryption key
- it is computationally easy to en/decrypt messages when the relevant (en/decrypt) key is known
- either of the two related keys can be used for encryption, with the other used for decryption (for some algorithms)

Public-Key Applications: -

- can classify uses into 3 categories:
  - encryption/decryption (provide secrecy)
  - digital signatures (provide authentication)
  - key exchange (of session keys)
- some algorithms are suitable for all uses, others are specific to one

Security of Public Key Schemes: -

- like private key schemes brute force exhaustive search attack is always theoretically possible
- but keys used are too large (>512bits)

Diffie-Hellman

- first public-key type scheme proposed by Diffie & Hellman in 1976 along with the exposition of public key concepts.
- Based on the difficulty of computing discrete logarithms of large numbers.

Public Parameter Creation	
A trusted party chooses and publishes a (large) prime p and an integer g having large prime order in $\mathbb{F}_p^*$.	
Private Computations	
Alice	**Bob**
Choose a secret integer a.	Choose a secret integer b.
Compute $A \equiv g^a \pmod{p}$.	Compute $B \equiv g^b \pmod{p}$.
Public Exchange of Values	
Alice sends A to Bob $\longrightarrow$	A
B	$\longleftarrow$ Bob sends B to Alice
Further Private Computations	
Alice	**Bob**
Compute the number $B^a \pmod{p}$.	Compute the number $A^b \pmod{p}$.
The shared secret value is $B^a \equiv (g^b)^a \equiv g^{ab} \equiv (g^a)^b \equiv A^b \pmod{p}$.	

- Where g is a primitive root of p.
- Let p be a prime. Then g is a *primitive root* for p if the powers of g, $1, g, g^2, g^3, \ldots$ include all of the residue classes mod p (except 0)
- **Examples:**

If p=7, then 3 is a primitive root for p because the powers of 3 are

1, 3, 2, 6, 4, 5---that is, every number mod 7 occurs except 0.

But 2 isn't a primitive root because the powers of 2 are

1, 2, 4, 1, 2, 4, 1, 2, 4...missing several values.

- **Example:**

If p=13, then 2 is a primitive root because the powers of 2 are

1, 2, 4, 8, 3, 6, 12, 11, 9, 5, 10, 7---which is all of the classes mod 13 except 0.

There are other primitive roots for 13 (?).

g

Eve

p

$$A = g^{x} \bmod p$$

$$B = g^{y} \bmod p$$

ALICE

BOB

A

$$k = B^{x} \bmod p$$

B

$$k' = A^{y} \bmod p$$

$$k' = k = g^{xy} \bmod p$$

- Example : -
- Alice and Bob agree on **p = 23** and **g = 5**.(show that 5 is primitive root of 23)
- Alice chooses **a= 6** and sends **5 6 mod 23 = 8**.
- Bob chooses **b = 15** and sends **515 mod 23 = 19**.
- Alice computes **19 6 mod 23 = 2**.
- Bob computes **815 mod 23 = 2**. Then **2** is the shared secret.
- Clearly, much larger values of **a**, **b**, and **p** are required.

Rivest, Shamir and Adleman (RSA)

- RSA stands for Rivest, Shamir, and Adleman, they are the inventors of the RSA cryptosystem. RSA is one of the algorithms used in PKI (Public Key Infrastructure), asymmetric key encryption scheme. RSA is a block chiper, it encrypt message in blocks (block by block). The common size for the key length now is 1024 bits for P and Q, therefore N is 2048 bits, if the implementation (the library) of RSA is fast enough, we can double the key size.
- **Key Generation Algorithm**
- Generate two large random primes, *p* and *q*, of approximately equal size such that their product n = pq is of the required bit length, e.g. 1024 bits.
- Compute n = pq and (ϕ) phi = (p-1)(q-1).
- Choose an integer *e*, 1 < e < phi, such that gcd(e, phi) = 1.
- Compute the secret exponent *d*, 1 < d < phi, such that ed ≡ 1 (mod phi).
- The public key is (n, e) and the private key is (n, d). Keep all the values d, p, q and phi secret.
- n is known as the *modulus*.
- e is known as the *public exponent* or *encryption exponent* or just the *exponent*.
- d is known as the *secret exponent* or *decryption exponent*.\`

- In encryption, represents the plaintext message as a positive integer *m and* computes the ciphertext $C = m^e \bmod n$.
- In decryption compute $m = c^d \bmod n$

Example : let p=17 & q=11 then

- Compute n = pq =17×11=187.
- Compute ø(n) or (ϕ) phi =(p–1)(q-1)=16×10=160.
- choose e=7 (1 < e < 160) where gcd(7,160)=1.
- d=23 where 1 < d < 160 and ed ≡ 1 (mod 160).(multiplication inverse).
- The public key is (187, 7) and the private key is (187, 23).
- given message M = 88 (88<187)
- encryption: $C = m^e \bmod n$: $C = 88^7 \bmod 187 = 11$.
- Decryption: $m = c^d \bmod n$: $m = 11^{23} \bmod 187 = 88$.
-

- **Ex**/ p=3,q=11,e=7,m=2 encrypt and decrypt using RSA Algorithm?
- Choose p = 3 and q = 11
- Compute n = p * q = 3 * 11 = 33
- Compute $\phi(n) = (p - 1) * (q - 1) = 2 * 10 = 20$
- Let e = 7
- Compute d = 3 [(3 * 7) mod 20 = 1]
- Public key is (e, n) => (7, 33)
- Private key is (d, n) => (3, 33)
- The encryption of $m = 2$ is $c = 2^7 \bmod 33 = 29$
- The decryption of $c = 29$ is $m = 29^3 \bmod 33 = 2$