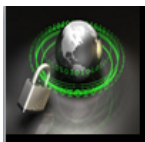
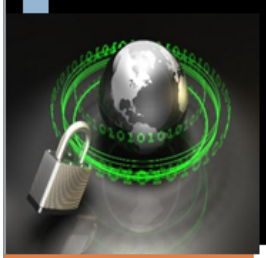


BLOCK CIPHER & DES

CHAPTER THREE -PART 2-



By : Eman Talib Jasim

Department of Information Technology 2018-2019

1. WHAT MAKES DES A STRONG CIPHER

• The substitution step is very effective as far as diffusion is concerned. It has been shown that if you change just one bit of the 64-bit input data block, on the average that alters 34 bits of the ciphertext block.
• The manner in which the round keys are generated from the encryption key is also very effective as far as confusion is concerned. It has been shown that if you change just one bit of the encryption key, on the average that changes 35 bits of the ciphertext.
• Both effects mentioned above are referred to as the avalanche effect.
• And, of course, the 56-bit encryption key means a key space of size 256 ≈ 7.2 × 1016.
• Assuming that, on the average, you'd need to try half the keys in a brute-force attack, a machine able to process 1000 keys per microsecond would need roughly 13 months to break the code. However, a parallel-processing machine trying 1 million keys simultaneously would need only about 10 hours. (EFF took three days on a specially architectured machine to break the code.)

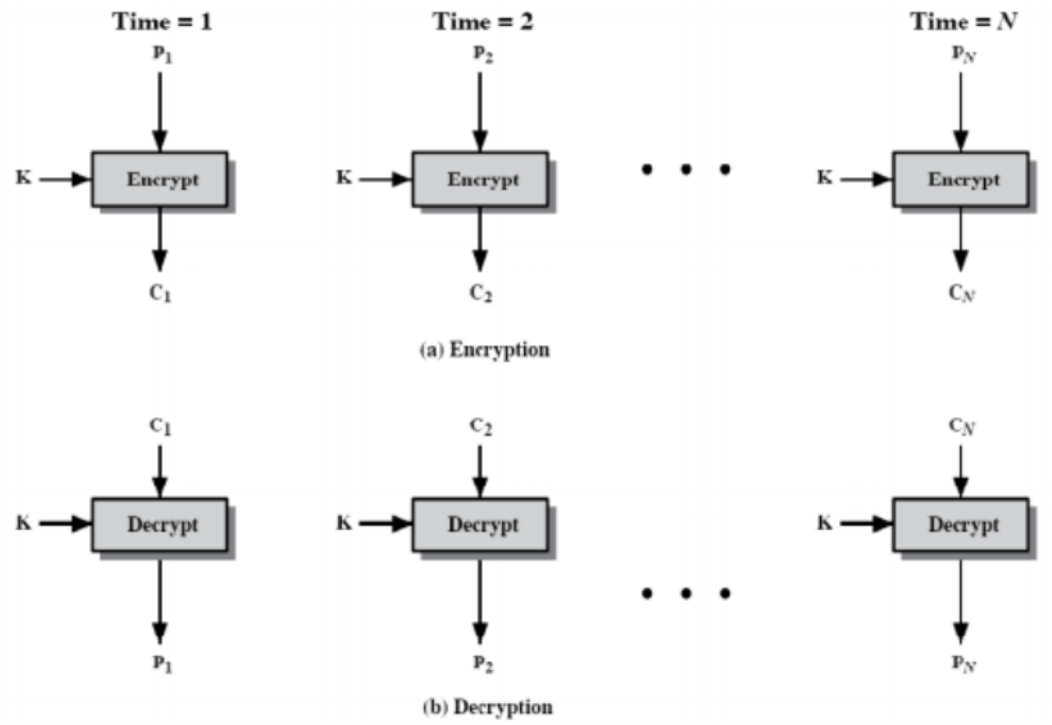
Block Cipher Modes of operations

Block cipher is basic building block, it divides the plaintext into fixed size blocks then encrypt them block by block. To enhance effect of algorithm, each is suitable for certain applications. There are five modes of operations to encrypt the blocks as shown in the following table:

Mode	Description	Typical Application
Electronic Codebook (ECB)	Each block of 64 plaintext bits is encoded independently using the same key.	•Secure transmission of single values (e.g., an encryption key)
Cipher Block Chaining (CBC)	The input to the encryption algorithm is the XOR of the next 64 bits of plaintext and the preceding 64 bits of ciphertext.	•General-purpose block-oriented transmission •Authentication
Cipher Feedback (CFB)	Input is processed *s* bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.	•General-purpose stream-oriented transmission •Authentication
Output Feedback (OFB)	Similar to CFB, except that the input to the encryption algorithm is the preceding DES output.	•Stream-oriented transmission over noisy channel (e.g., satellite communication)
Counter (CTR)	Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block.	•General-purpose block-oriented transmission •Useful for high-speed requirements

1. Electronic Codebook Mode (ECB)

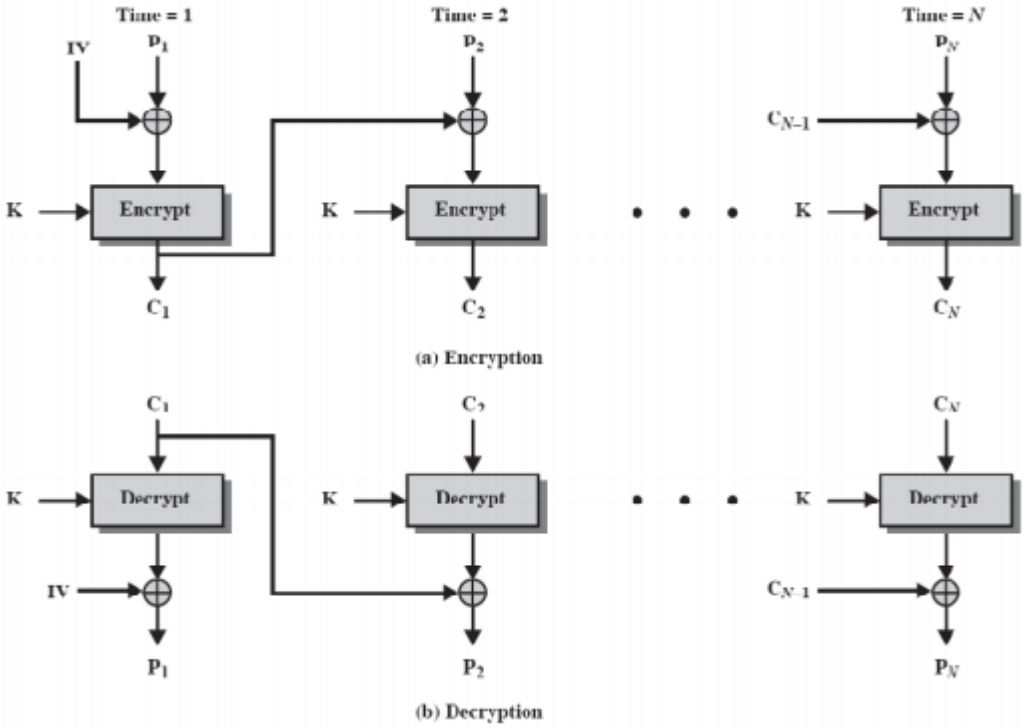
- Break message into b-bit blocks (e.g. 64) Ci=DESK1(Pi)
- Encrypt each block independently
- Suitable for short amount of data (e.g key)
- Identical P blocks produce same C blocks
- Cryptanalysis is possible



When identical block repeated in the message, the result is the same ciphertext block, it cause a weak point for the attacker, so it prefer to use this mode with the smaller messages.

2. **Cipher Block Chaining Mode (CBC)**

- XOR current P block and previous C block
  Ci=DESk1(Pi XOR Ci-1) Ci-1=IV
- Identical P blocks produce different C blocks
- Suitable for long messages
- Initial Vector (IV) is XORed with first block
- IV must be known to both sides
- IV can be encrypted using ECB mode



Encryption

- C1 = E(K, IV ⊕ P1)
- Ci = E(K, Ci-1 ⊕ Pi)

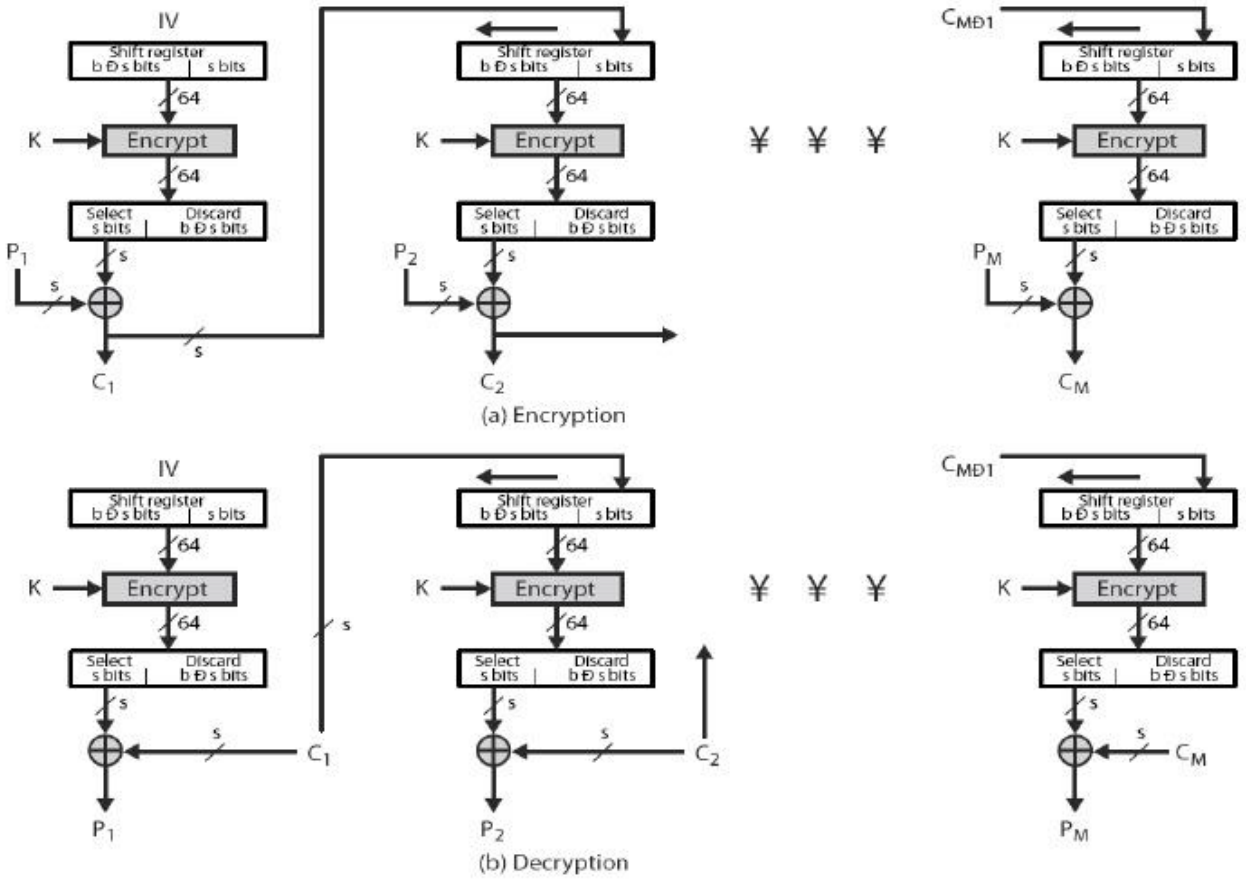
Decryption

- P1 = IV ⊕ D(K, C1)
- Pi = Ci-1 ⊕ D(K, Ci)

3. Cipher Feedback Mode (CFB)

- Use block cipher as stream cipher Ci=Pi Xor DESK1(Ci-1)
- Encrypt one byte at a time
- No need to pad message to 64 bits
- Chaining is used as in CBC
- Unit of transmission is s bits (usually 8)

Cipher FeedBack (CFB)



CFB Encryption

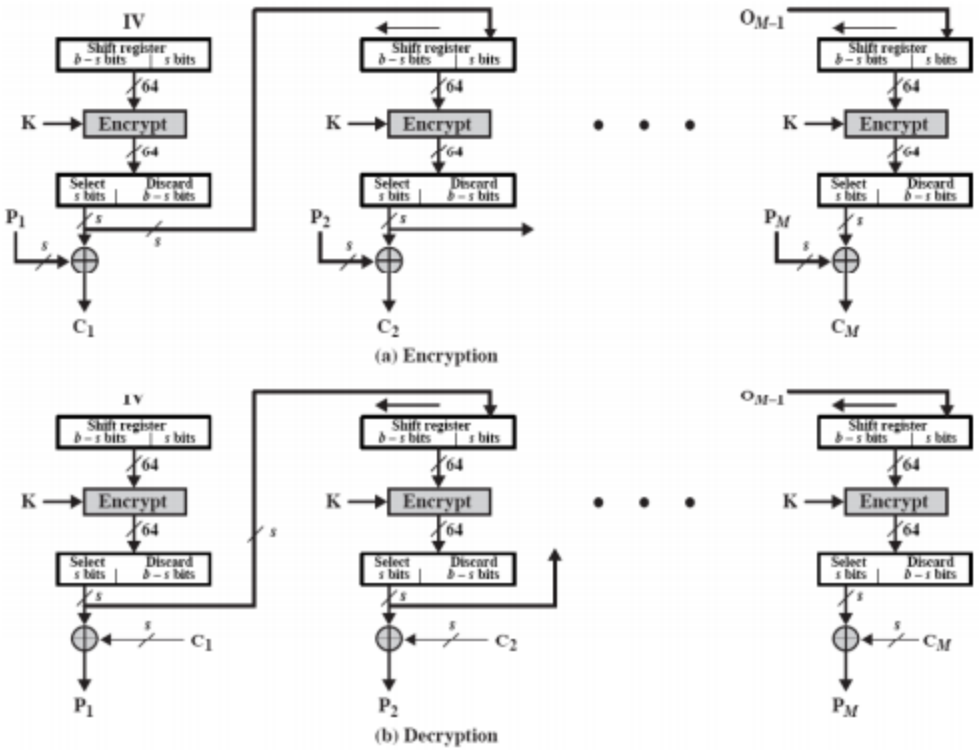
- Input to E is b-bit shift register, initially IV
- From the output of E, select most significant s bits: Ss[E]
- XOR with P1 to produce C1
- C1 is also placed in least significant s bits of shift register (left shift)
- Repeat the process until all units encrypted

CFB Decryption

4. Encryption algorithm is again used!
5. Initially IV as input
6. From output of E, select most significant s bits
7. XOR with C1 produces P1
8. C1 is also placed in least significant s bits of shift register (left shift)
9. C1 = P1 $\oplus$ Ss[E(K, IV)])]
10. P1 = C1 $\oplus$ Ss[E(K, IV)]

4. Output Feedback Mode (OFB)

- Similar to CFB with one difference
- Output of E instead of Ci is placed in shift register
- Advantage: bit errors in transmission don't propagate



OFB Encryption

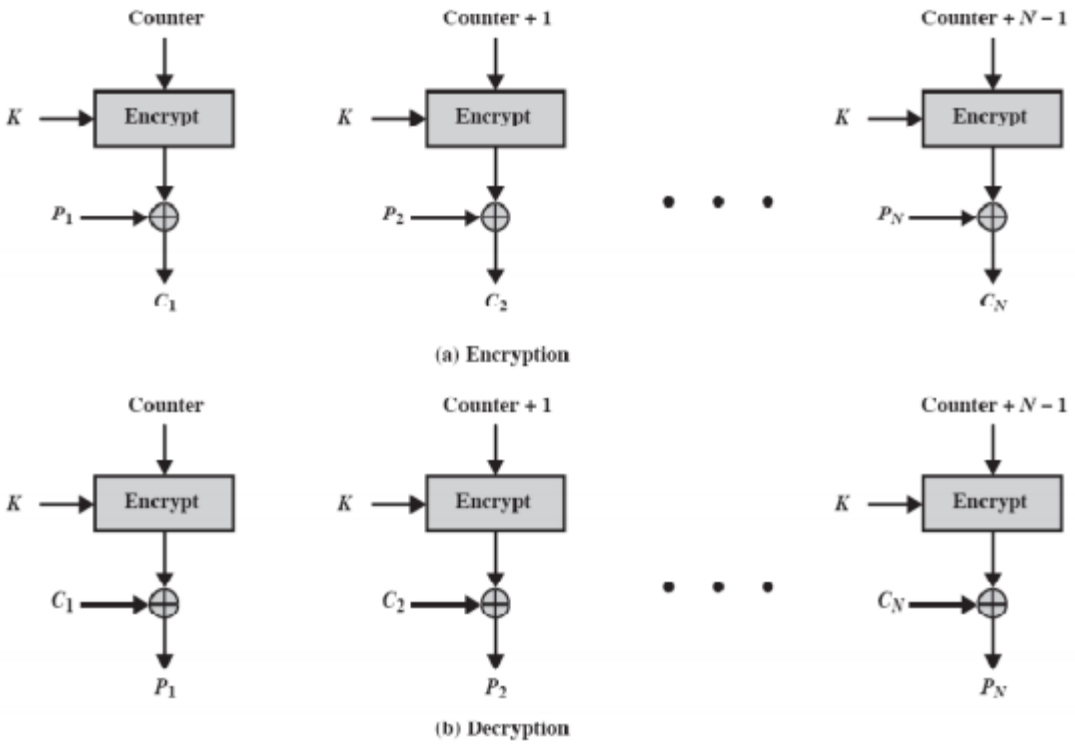
- Input to E is b-bit shift register, initially IV
- From the output of E, select most significant s bits: Ss[E]
- XOR with P1 to produce C1
- Output of E is also placed in least significant s bits of shift register (left shift)
- Repeat the process until all units encrypted

OFB Decryption

- Encryption algorithm is used
- Initially IV as input
- From output of E, select most significant s bits
- XOR with C1 produces P1
- Output of E is also placed in least significant s bits of shift register (left shift)
- C1 = P1 $\oplus$ Ss[E(K, IV)]
- P1 = C1 $\oplus$ Ss[E(K, IV)]

5. Counter Mode

- Counter with b-bits (block size of E) is used
- Counter value must be different for each P
- Counter value initialized to certain value
- Counter incremented for each subsequent P
- Encryption: Ci = Pi ⊕ E(K, Counter)]
- Decryption: Pi = Ci ⊕ E(K, Counter)]
- No chaining is used



Counter Mode Advantages

- Hardware efficiency
- No chaining blocks encrypted in parallel , blocks encrypted in parallel
- Software efficiency
- parallel processing can be used
- Preprocessing
- Output of E can be pre-computed & stored

- Random access
- i^{th} block of plain/cipher text can be processed randomly
- Provable security
- Simplicity
- Only encryption algorithm is needed