Cryptography And Network Security

UNIT-III

Session 19

Dr. V. K. Pachghare

Department of Computer Engineering and Information Technology
College of Engineering Pune (COEP)
Forerunners in Technical Education

Modes of Operation

Introduction

- Some times the number of bits in the plaintext/message are not multiple of the block size

Introduction

- Some times the number of bits in the plaintext/message are not multiple of the block size
- Suppose the size of message is 140 bits and we are using DES algorithm

Introduction

- Some times the number of bits in the plaintext/message are not multiple of the block size
- Suppose the size of message is 140 bits and we are using DES algorithm
- Then we split the message into multiple of 64 bits

Introduction

- Some times the number of bits in the plaintext/message are not multiple of the block size
- Suppose the size of message is 140 bits and we are using DES algorithm
- Then we split the message into multiple of 64 bits
- So we get 2 complete blocks and 12 bits in the third block

- In this case, we use the techniques known as **modes of operation**.

- In this case, we use the techniques known as **modes of operation**.
- There are various modes of operations

- In this case, we use the techniques known as **modes of operation**.
- There are various modes of operations
- Any block cipher can be operated in one of several modes

Advantages

- Helps to use symmetric encryption algorithms for messages of arbitrary length.

Advantages

- Helps to use symmetric encryption algorithms for messages of arbitrary length.
- Provide additional level of security to the encryption algorithm.

Advantages

- Helps to use symmetric encryption algorithms for messages of arbitrary length.
- Provide additional level of security to the encryption algorithm.
- Replay attack of packets can be avoided

Modes of operation

1. Electronic Codebook (ECB) Mode
2. Cipher Block Chaining (CBC) Mode
3. Feedback Modes
4. Counter Mode

Electronic Codebook (ECB) Mode

Introduction

- The simplest mode of operation

Introduction

- The simplest mode of operation
- Each plaintext block of 64 bits is encrypted independently

Introduction

- The simplest mode of operation
- Each plaintext block of 64 bits is encrypted independently
- The input data is padded out to a multiple of the block size, broken into an integer number of blocks, each of which is encrypted independently using the key

ECB: Encryption

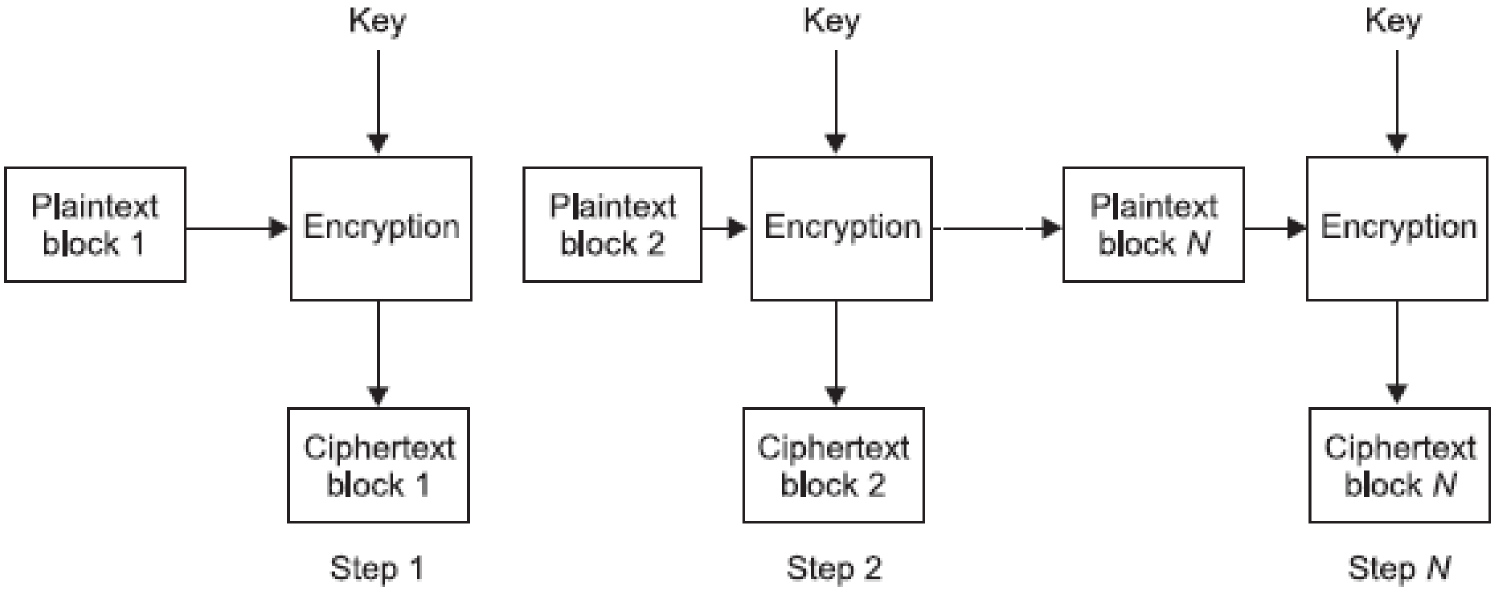


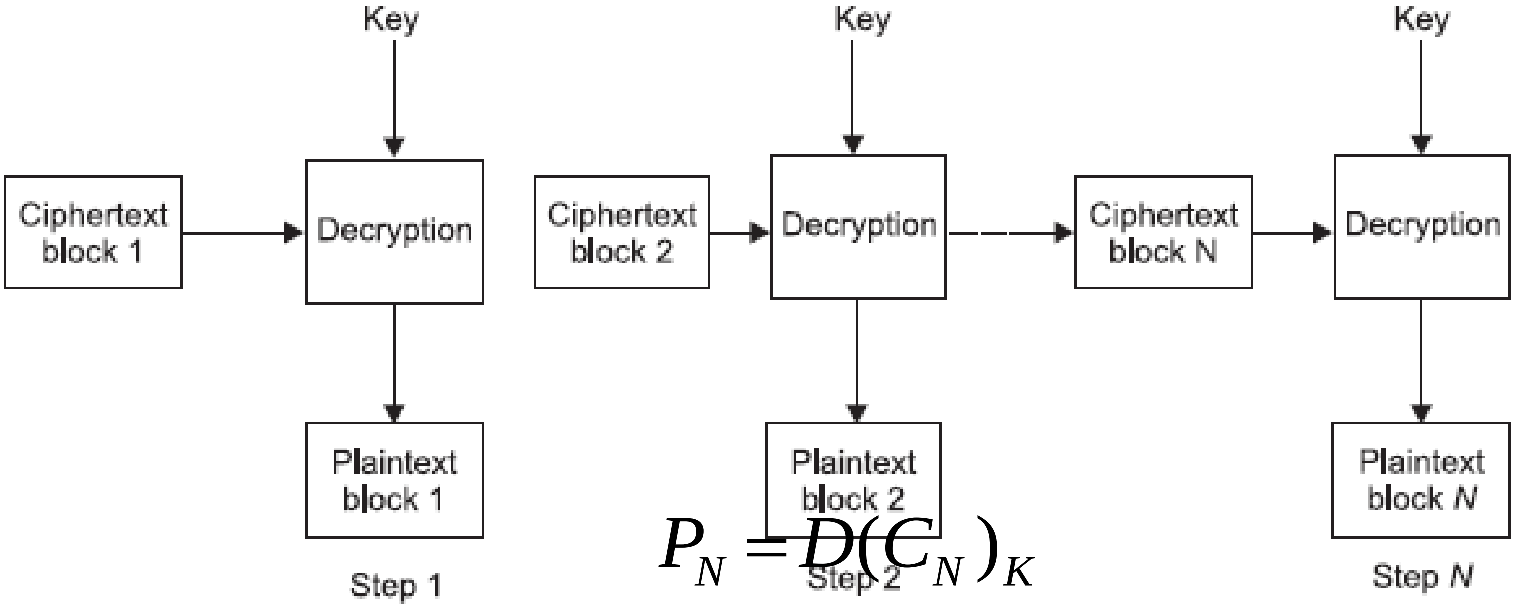
Figure 3.1 Electronic code book mode: Encryption.

$$C_N = E(P_N)_K$$

- Allows easy parallelization to yield higher performance

- Allows easy parallelization to yield higher performance
- Each identical block of plaintext gives an identical block of ciphertext that is the same block of plaintext appears more than once in the message, it always produces the same ciphertext

ECB: Decryption



$$P_N = D(C_N)_K$$

Figure 3.2 Electronic code book mode: Decryption.

Advantages

- We can process multiple blocks simultaneously.

Advantages

- We can process multiple blocks simultaneously.
- If any plaintext or ciphertext blocks lost, it does not affect on the output of other blocks.

Advantages

- We can process multiple blocks simultaneously.
- If any plaintext or ciphertext blocks lost, it does not affect on the output of other blocks.
- Parallel processing during encryption as well as decryption helps to increase the speed and the performance of the algorithm

Disadvantage

- If two plaintext blocks are identical, then the ciphertext block generated are also same
- Therefore, known plaintext attack is possible

Cipher Block Chaining (CBC) Mode

Encryption

- An initialization vector (IV) is used.

Encryption

- An initialization vector (IV) is used.
- It is selected randomly.

Encryption

- An initialization vector (IV) is used.
- It is selected randomly.
- It helps to increase the security

Encryption

- An initialization vector (IV) is used.
- It is selected randomly.
- It helps to increase the security
- Perform XOR operation between the first plaintext block and the initialization vector

Encryption

- An initialization vector (IV) is used.
- It is selected randomly.
- It helps to increase the security
- Perform XOR operation between the first plaintext block and the initialization vector
- The 64-bit output is encrypted using the secret key

Encryption

- An initialization vector (IV) is used.
- It is selected randomly.
- It helps to increase the security
- Perform XOR operation between the first plaintext block and the initialization vector
- The 64-bit output is encrypted using the secret key
- The result is 64-bit ciphertext

Encryption

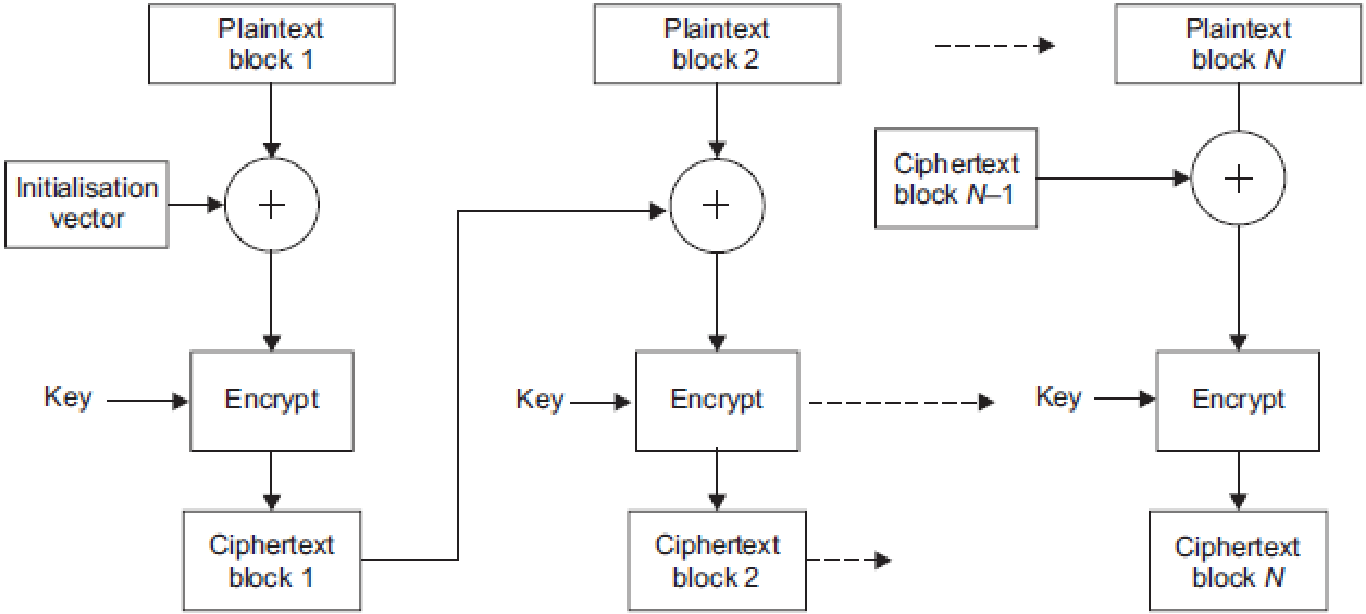


Figure 3.3 Cipher block chaining mode: Encryption.

$$\{P_N \oplus C_{N-1}\}_k \rightarrow C_N$$

Initialization Vector (IV)

- The initialisation vector (IV) is used to provide semantic security, i.e. identical blocks of plaintext generate different ciphertext

Initialization Vector (IV)

- The initialisation vector (IV) is used to provide semantic security, i.e. identical blocks of plaintext generate different ciphertext
- As the value of IV is different for each operation, the common start of plaintext blocks is also hidden.

Initialization Vector (IV)

- The initialisation vector (IV) is used to provide semantic security, i.e. identical blocks of plaintext generate different ciphertext
- As the value of IV is different for each operation, the common start of plaintext blocks is also hidden.
- For better security, the IV value should be unpredictable by the attacker

- It is better practice to use a random key for the generation of IV.

- It is better practice to use a random key for the generation of IV.
- IV is always transmitted as unencrypted, it is not necessary for the receiver to know this key

- It is better practice to use a random key for the generation of IV.
- IV is always transmitted as unencrypted, it is not necessary for the receiver to know this key
- If the same plaintext block is repeated, different ciphertext blocks are produced. IV is used as a seed for the encryption process. This helps to make each plaintext unique

Decryption

- The reverse process of encryption.

Decryption

- The reverse process of encryption.
- For decryption, the same key is used as encryption.

Decryption

- The reverse process of encryption.
- For decryption, the same key is used as encryption.
- First block of ciphertext is decrypted using the key.

Decryption

- The reverse process of encryption.
- For decryption, the same key is used as encryption.
- First block of ciphertext is decrypted using the key.
- Then XOR operation is performed with the initialization vector.

- The output is a plaintext.

- The output is a plaintext.
- Same operation is repeated for each ciphertext blocks.

- The output is a plaintext.
- Same operation is repeated for each ciphertext blocks.
- Only difference is that instead of initialization vector, ciphertext block of previous step is used.

- The output is a plaintext.
- Same operation is repeated for each ciphertext blocks.
- Only difference is that instead of initialization vector, ciphertext block of previous step is used.
- The complete decryption process is as shown

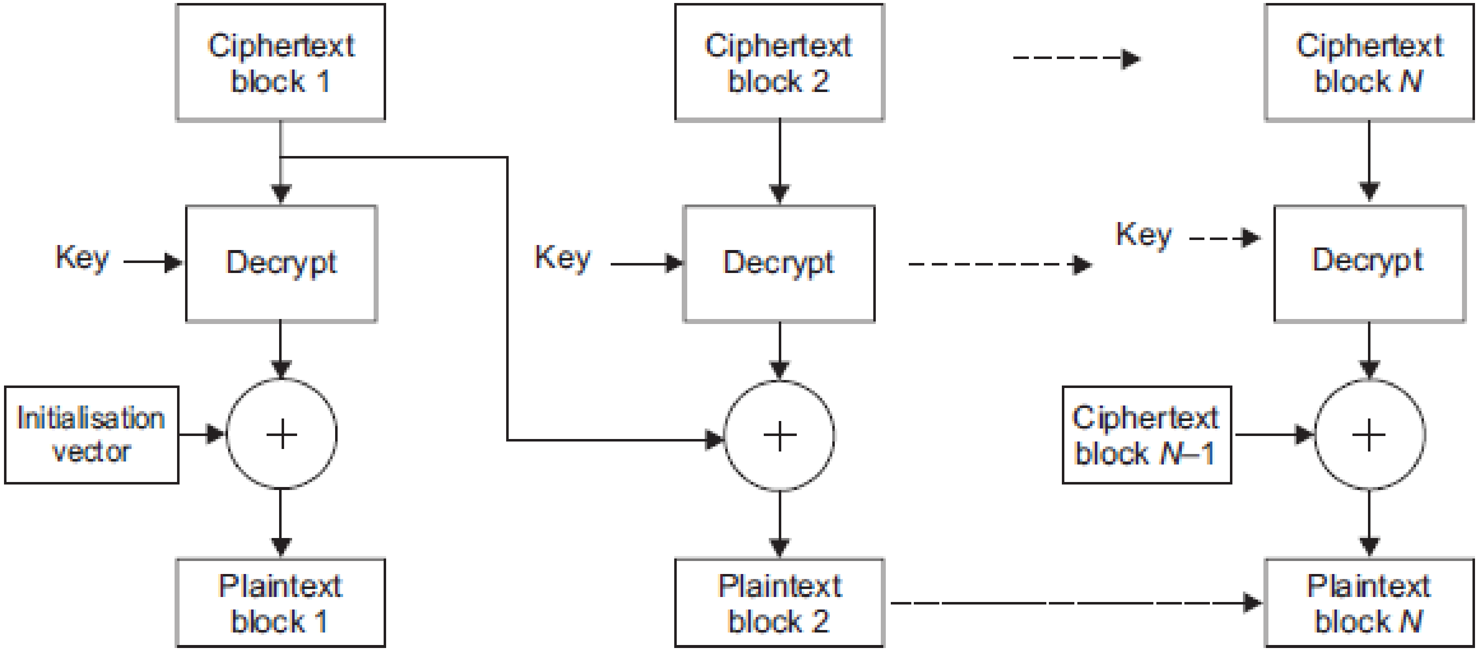


Figure 3.4 Cipher block chaining mode: Decryption.

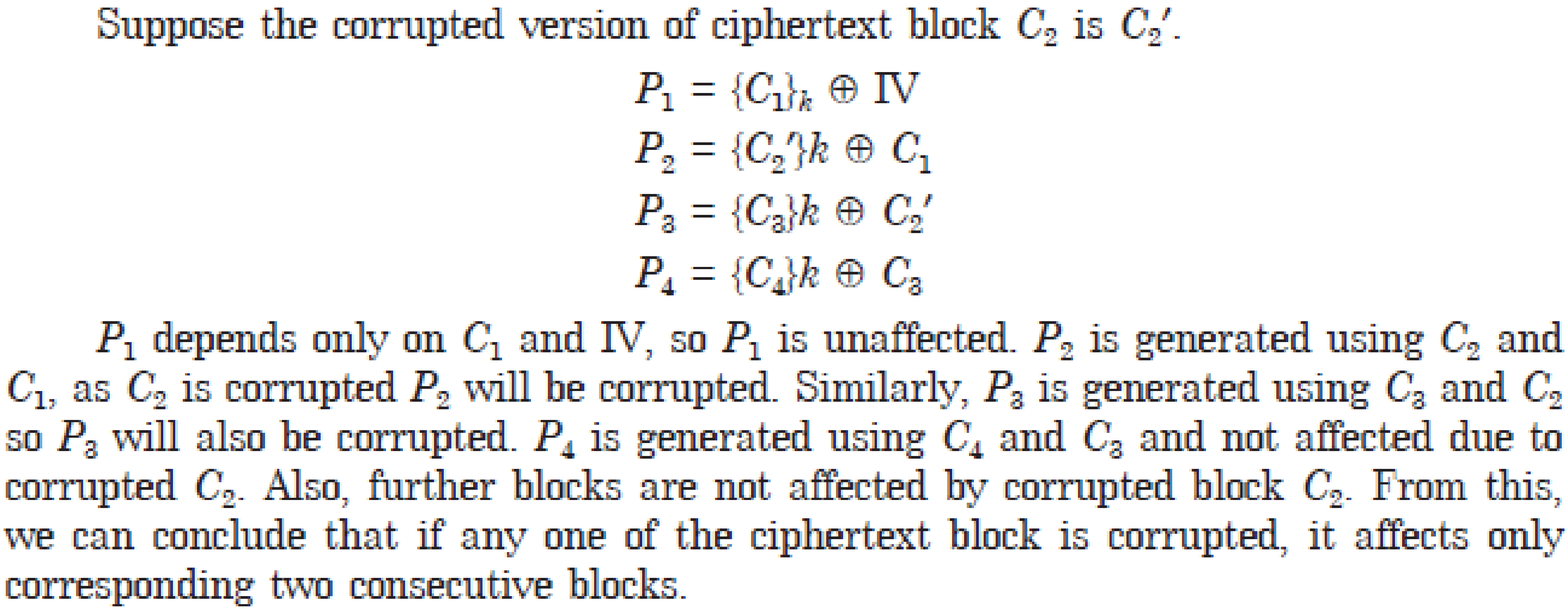
$$\{C_N\}_k \oplus C_{N-1} \rightarrow P_N$$

- Cipher block chaining can be used to generate the hash value.

- Cipher block chaining can be used to generate the hash value.
- The last ciphertext block is used as hash value for the given message.

- Cipher block chaining can be used to generate the hash value.
- The last ciphertext block is used as hash value for the given message.
- Because the last ciphertext block is dependent on all the plaintext blocks.

- Cipher block chaining can be used to generate the hash value.
- The last ciphertext block is used as hash value for the given message.
- Because the last ciphertext block is dependent on all the plaintext blocks.
- This hash value helps to check if any one of the ciphertext blocks lost or modified.



Suppose the corrupted version of ciphertext block C_2 is C_2'.

$$P_1 = \{C_1\}_k \oplus \text{IV}$$
$$P_2 = \{C_2'\}k \oplus C_1$$
$$P_3 = \{C_3\}k \oplus C_2'$$
$$P_4 = \{C_4\}k \oplus C_3$$

P_1 depends only on C_1 and IV, so P_1 is unaffected. P_2 is generated using C_2 and C_1, as C_2 is corrupted P_2 will be corrupted. Similarly, P_3 is generated using C_3 and C_2 so P_3 will also be corrupted. P_4 is generated using C_4 and C_3 and not affected due to corrupted C_2. Also, further blocks are not affected by corrupted block C_2. From this, we can conclude that if any one of the ciphertext block is corrupted, it affects only corresponding two consecutive blocks.

Advantages

- For identical blocks of plaintext, different ciphertext blocks are generated. So, CBC is more secure as compared to ECB mode.

Advantages

- For identical blocks of plaintext, different ciphertext blocks are generated. So, CBC is more secure as compared to ECB mode.
- Hash value, i.e., last ciphertext block, helps to identify if the message is original or modified.

Disadvantages

- Parallel operation cannot be performed. So, it is slower as compared to ECB.

Disadvantages

- Parallel operation cannot be performed. So, it is slower as compared to ECB.
- Lost/missing of any block of ciphertext stops the decryption process of the remaining blocks.

Feedback Modes

- Cipher feedback mode
- Output Feedback Mode (OFB)

Cipher Feedback Mode [CFB]

- Data encrypted in units are smaller than the block size

Cipher Feedback Mode [CFB]

- Data encrypted in units are smaller than the block size
- This mode can be used to encrypt any number of bits e.g. single bits or single characters (bytes) before sending across an insecure data link

- CBC cannot use parallelisation, as the ciphertext of the first plaintext block is used for the processing of the next plaintext block

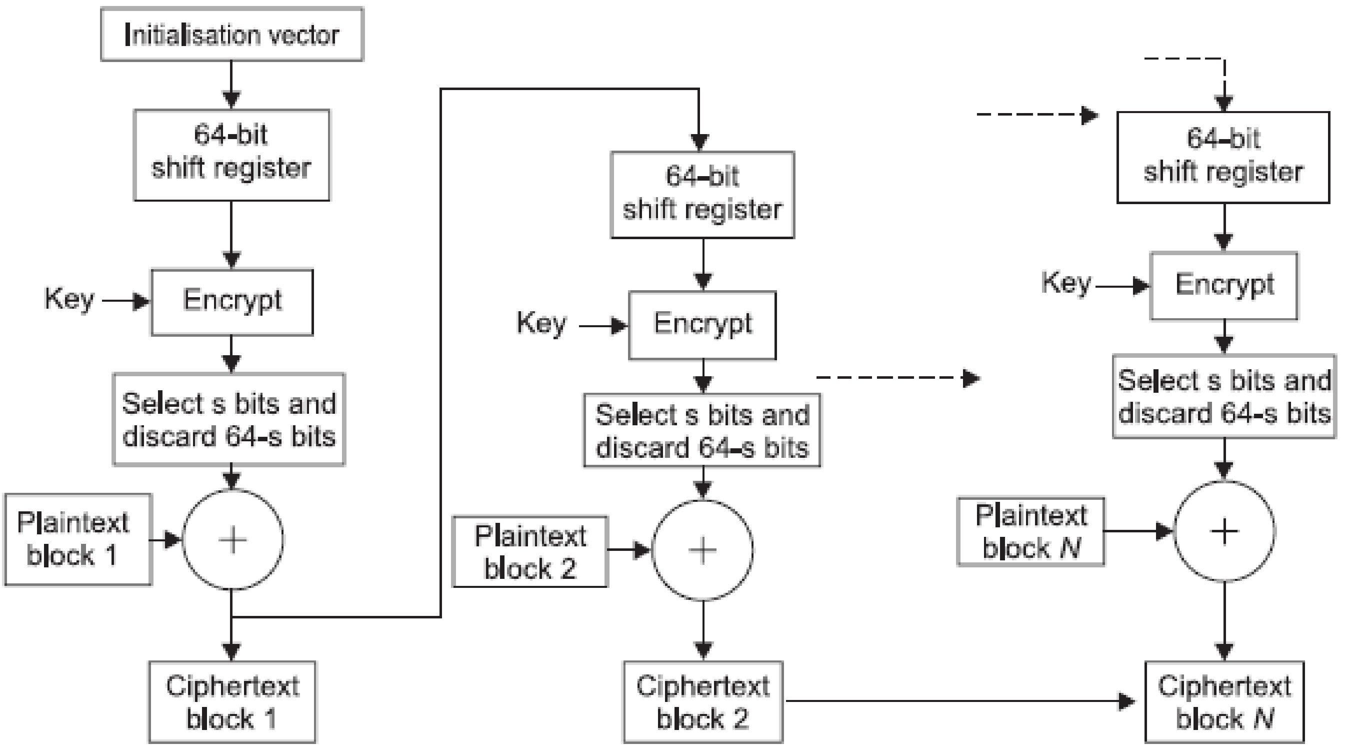


Figure 3.5 Cipher feedback mode: Encryption.

$$\{P_N \oplus C_{N-1}\}_k \rightarrow C_N$$

Encryption

- 64-bit shift register is used

Encryption

- 64-bit shift register is used
- shift register is filled by initialization vector, i.e., random number

Encryption

- 64-bit shift register is used
- shift register is filled by initialization vector, i.e., random number
- output of the shift register is encrypted using a key of size 64 bits

Encryption

- 64-bit shift register is used
- shift register is filled by initialization vector, i.e., random number
- output of the shift register is encrypted using a key of size 64 bits
- select leftmost “s” bits from the 64 bits and discard remaining bits. “s” is equal to the number of bits in the plaintext block

- XOR operation between the plaintext and the “s” bits.

- XOR operation between the plaintext and the “s” bits.
- The output is ciphertext of block size “s”.

- XOR operation between the plaintext and the “s” bits.
- The output is ciphertext of block size “s”.
- The ciphertext is also sent as input to the next step as input to the shift register.

- XOR operation between the plaintext and the “s” bits.
- The output is ciphertext of block size “s”.
- The ciphertext is also sent as input to the next step as input to the shift register.
- This process continues on all the blocks of plaintext to generate the ciphertext.

Decryption

- The shift register is filled with initialization vector same as used for encryption.

Decryption

- The shift register is filled with initialization vector same as used for encryption.
- Then encrypt the bits in the shift register using the key

Decryption

- The shift register is filled with initialization vector same as used for encryption.
- Then encrypt the bits in the shift register using the key
- Select “j” bits from the 64 bits and XOR with the ciphertext.

Decryption

- The shift register is filled with initialization vector same as used for encryption.
- Then encrypt the bits in the shift register using the key
- Select "j" bits from the 64 bits and XOR with the ciphertext.
- The output is plaintext.

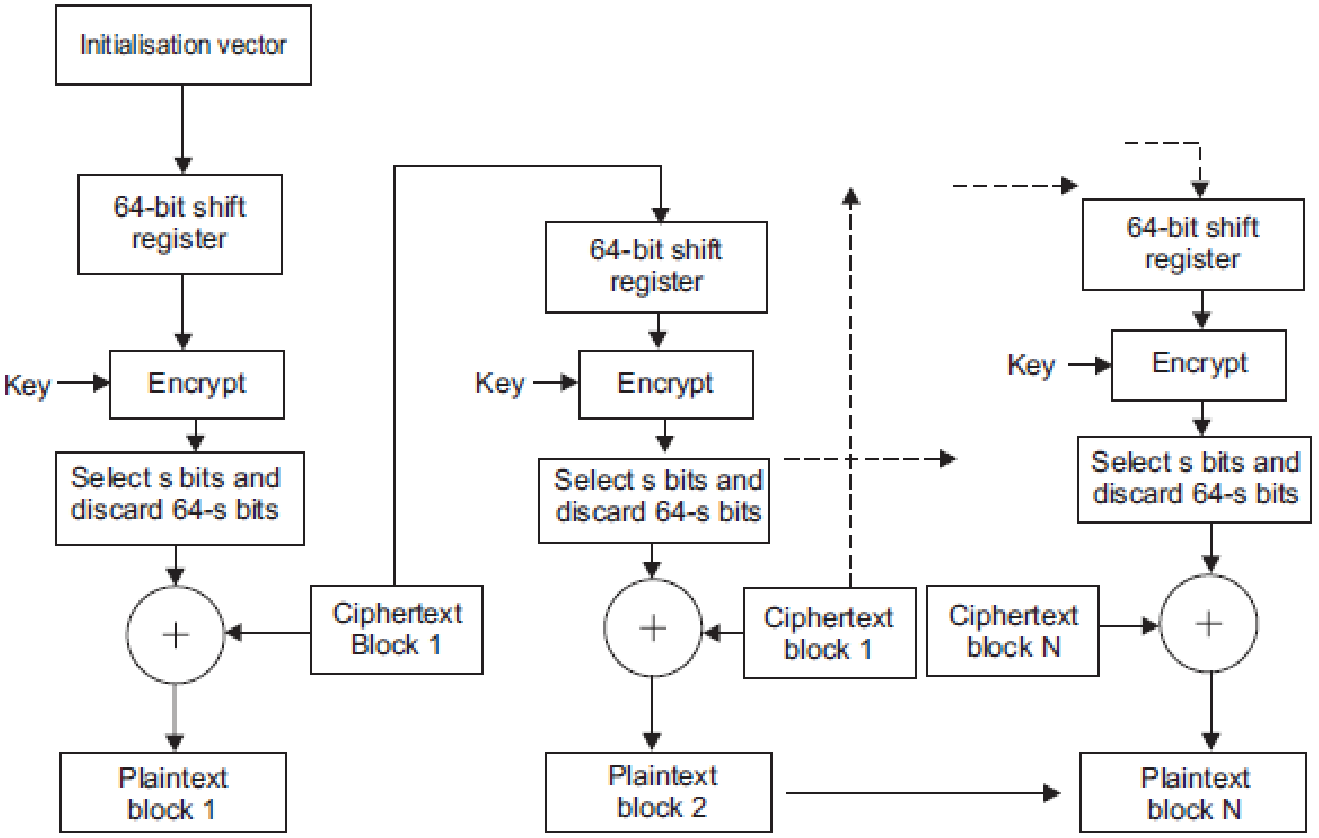


Figure 3.6 Cipher feedback mode: Decryption.

$$\{C_{N-1}\}_k \oplus C_N \rightarrow P_N$$

Disadvantage

- CFB is suffered from bit errors.

Disadvantage

- CFB is suffered from bit errors.
- If in the incoming cipher block, any one bit error is there, then it causes the bit error at the same bit position in the plaintext block.

Disadvantage

- CFB is suffered from bit errors.
- If in the incoming cipher block, any one bit error is there, then it causes the bit error at the same bit position in the plaintext block.
- The same ciphertext block is used as input to the shift register of the next step and causes bit errors in the next plaintext block.

- It cause bit errors in the plaintext for as long as the erroneous bits stay in the shift register

- It cause bit errors in the plaintext for as long as the erroneous bits stay in the shift register
- Suppose there is a bit error at 4th bit position of the 8-bit CFB, then subsequent 8 bytes will be garbled. After that the correct plaintext will be generated

Output Feedback Mode (OFB)

- Similar to CFB mode, except that the ciphertext output of DES is fed back into the Shift Register, rather than the actual final ciphertext

Output Feedback Mode (OFB)

- Similar to CFB mode, except that the ciphertext output of DES is fed back into the Shift Register, rather than the actual final ciphertext
- The Shift Register is set to an arbitrary initial value, and passed through the DES algorithm

Encryption

- Select 64-bit random for IV as input to the 64-bit shift register.

Encryption

- Select 64-bit random for IV as input to the 64-bit shift register.
- Encrypt the output of shift register with the key.

Encryption

- Select 64-bit random for IV as input to the 64-bit shift register.
- Encrypt the output of shift register with the key.
- Select “s” (value of “s” is equal to the size of plaintext block) bits from the encrypted output and discard 64-s bits.

- Performed XOR operation between the selected “s” bit and the plaintext block.

- Performed XOR operation between the selected “s” bit and the plaintext block.
- The output is ciphertext.

- Performed XOR operation between the selected “s” bit and the plaintext block.
- The output is ciphertext.
- The selected “s” bits are used as input for the shift register on the next step.

Encryption

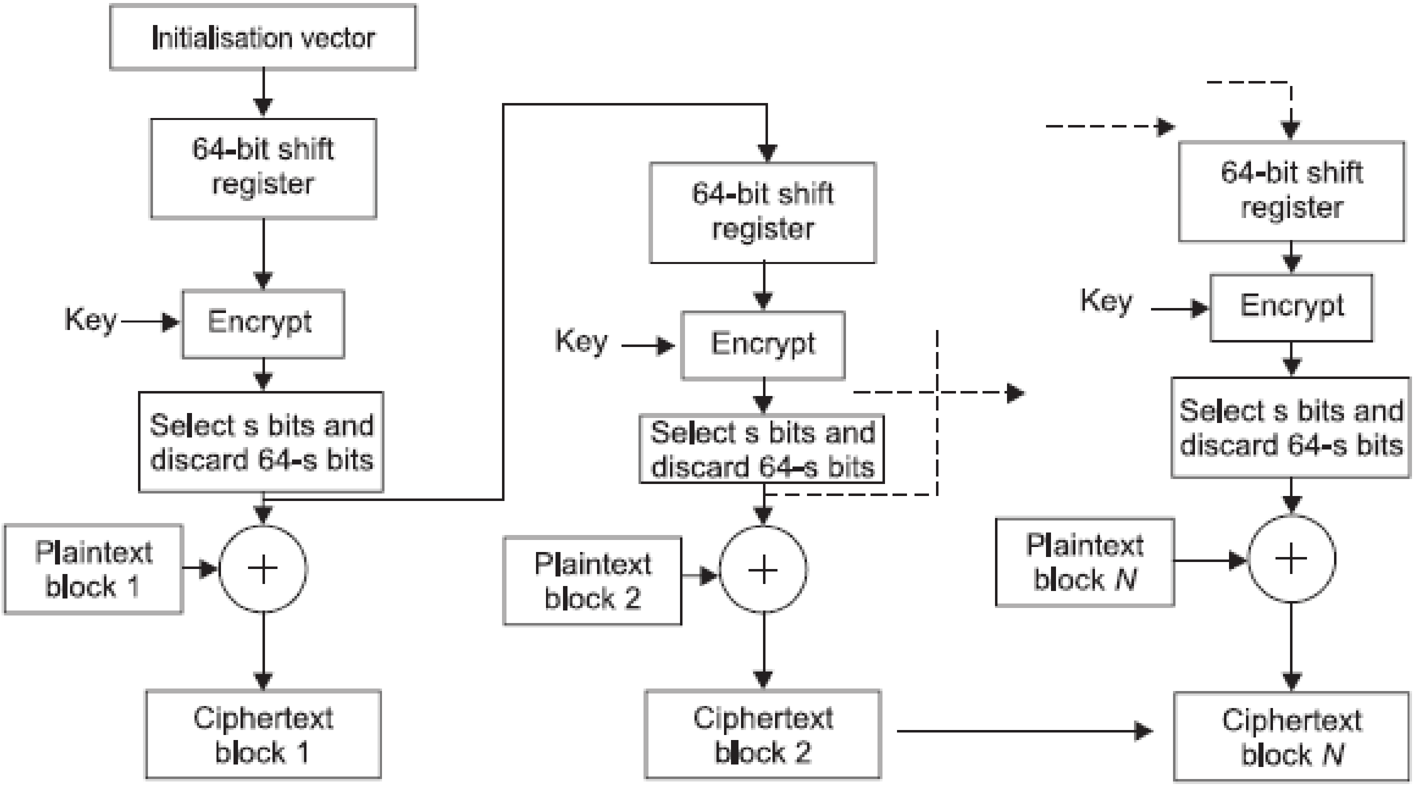


Figure 3.7 Output feedback mode: Encryption.

Decryption

- Similar to encryption.

Decryption

- Similar to encryption.
- Instead of plaintext block, corresponding ciphertext block is used for XOR operation

Decryption

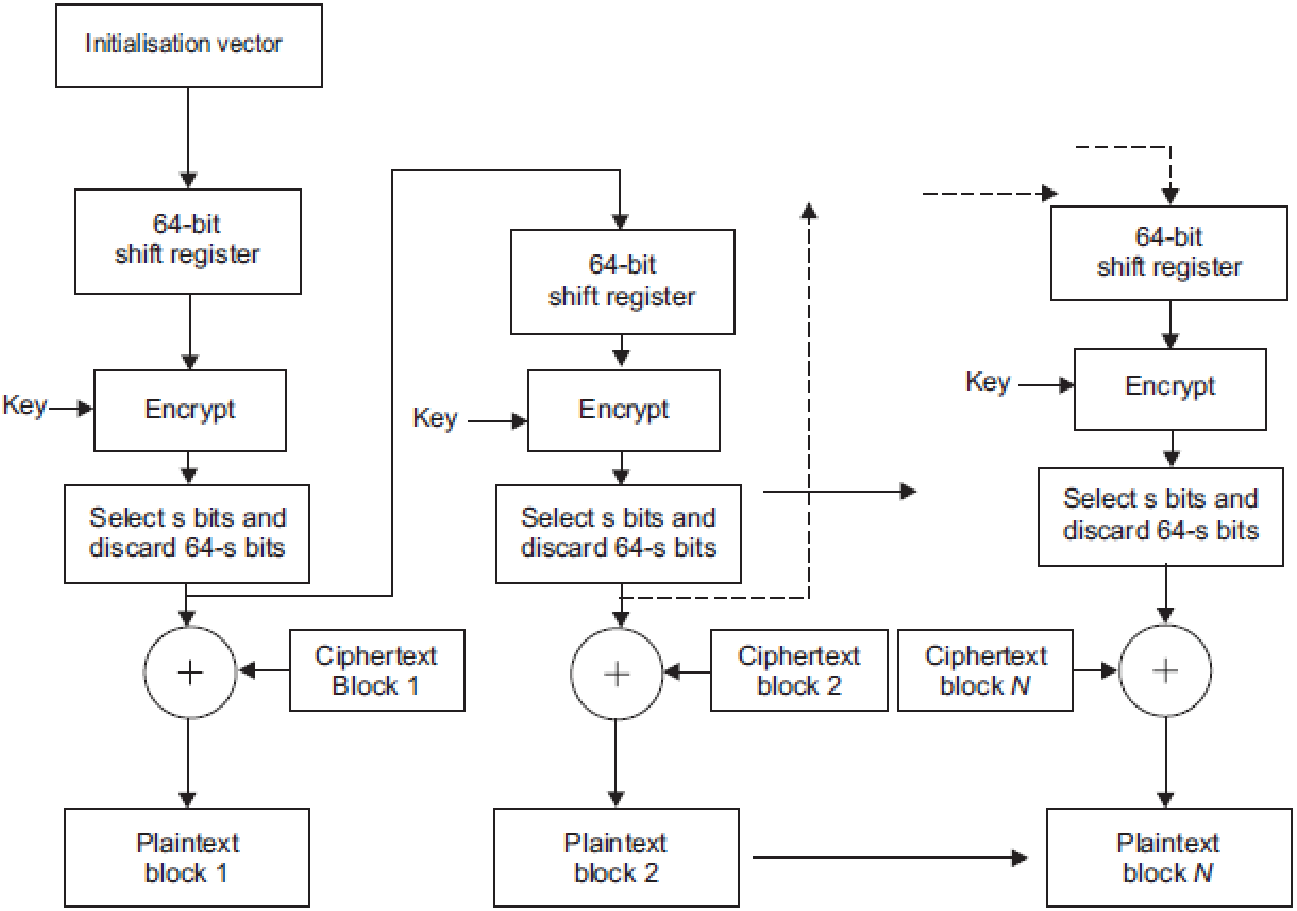


Figure 3.8 Output feedback mode: Decryption.

Disadvantages

- Cryptanalysis of output feedback mode is easy

Disadvantages

- Cryptanalysis of output feedback mode is easy
- Only a ciphertext block and encrypted “s” bits are sufficient to get the plaintext block.

Disadvantages

- Cryptanalysis of output feedback mode is easy
- Only a ciphertext block and encrypted “s” bits are sufficient to get the plaintext block.
- Here information about the key is not required, which help the cryptanalyst to break the cipher easily.

Disadvantages

- Cryptanalysis of output feedback mode is easy
- Only a ciphertext block and encrypted "s" bits are sufficient to get the plaintext block.
- Here information about the key is not required, which help the cryptanalyst to break the cipher easily.
- Therefore, this mode is less secure than cipher feedback mode.

Counter mode

Introduction

- A block cipher is worked like a stream cipher.

Introduction

- A block cipher is worked like a stream cipher.
- The counter is used whose value is changed in each round.

Introduction

- A block cipher is worked like a stream cipher.
- The counter is used whose value is changed in each round.
- Initially, the user has to set some value to the counter.

Introduction

- A block cipher is worked like a stream cipher.
- The counter is used whose value is changed in each round.
- Initially, the user has to set some value to the counter.
- Encrypt the counter value and the key.

Introduction

- A block cipher is worked like a stream cipher.
- The counter is used whose value is changed in each round.
- Initially, the user has to set some value to the counter.
- Encrypt the counter value and the key.
- This encrypted value is XOR with the block of plaintext. The result is a block of ciphertext.

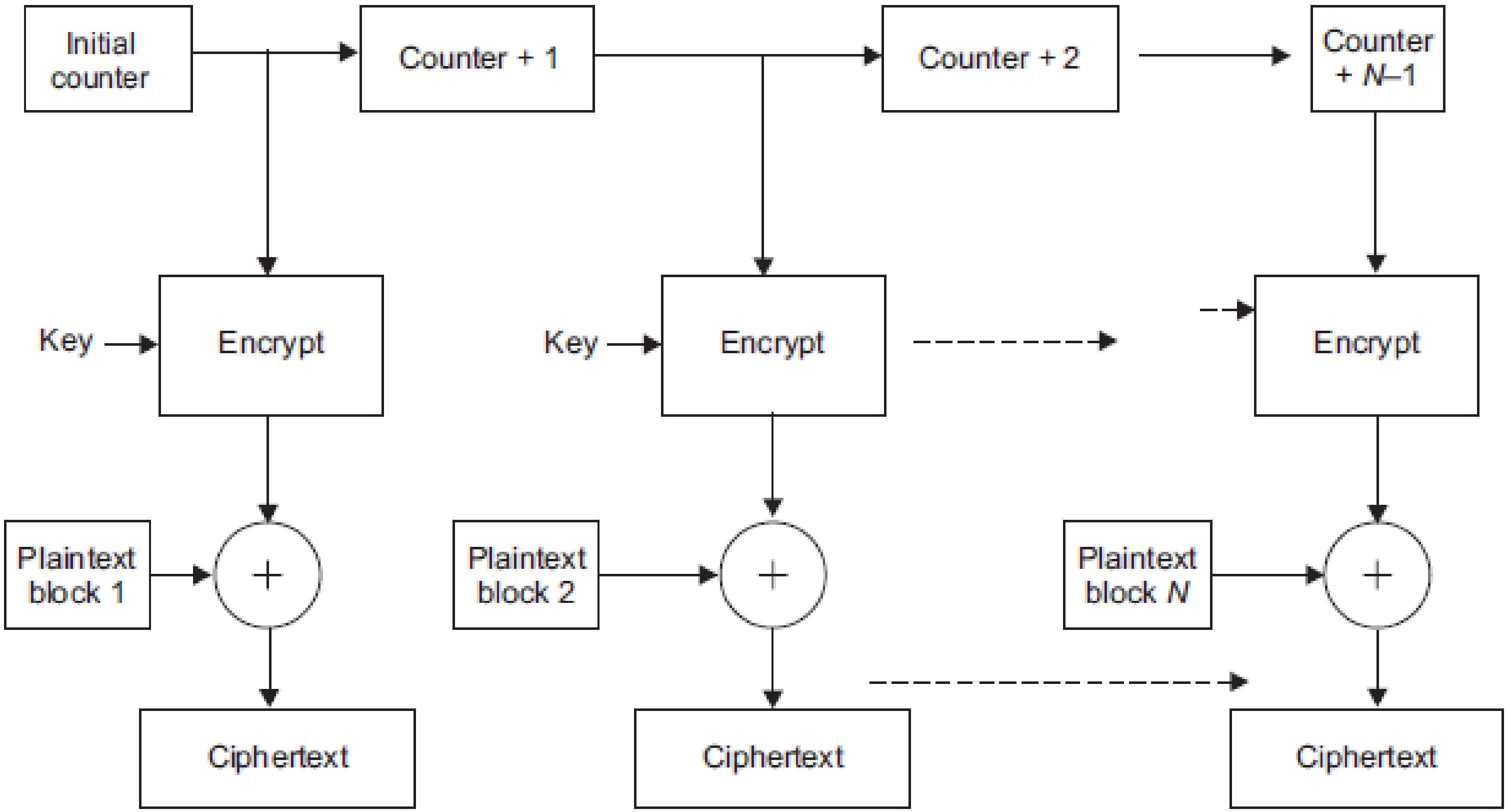


Figure 3.9 Counter mode: Encryption.

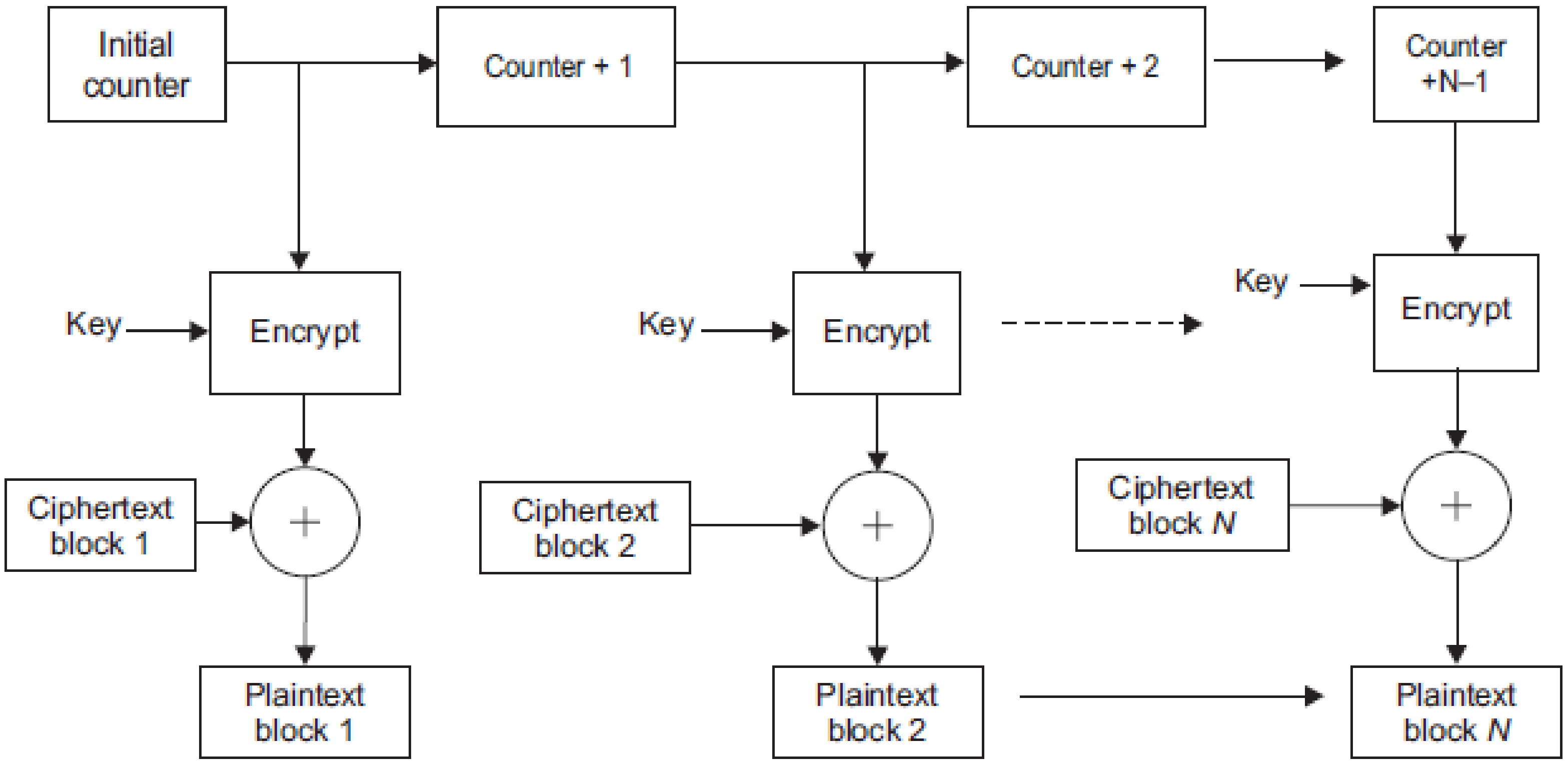


Figure 3.10 Counter mode: Decryption.

Advantages

- Faster than of cipher block chaining mode.

Advantages

- Faster than of cipher block chaining mode.
- Encryption can be done in parallel.

Advantages

- Faster than of cipher block chaining mode.
- Encryption can be done in parallel.
- Padding is not required.

Advantages

- Faster than of cipher block chaining mode.
- Encryption can be done in parallel.
- Padding is not required.
- Processing of plaintext blocks can be done randomly.

Advantages

- Faster than of cipher block chaining mode.
- Encryption can be done in parallel.
- Padding is not required.
- Processing of plaintext blocks can be done randomly.
- Only encryption algorithm is required.

Advantages

- Faster than of cipher block chaining mode.
- Encryption can be done in parallel.
- Padding is not required.
- Processing of plaintext blocks can be done randomly.
- Only encryption algorithm is required.
- It is as secure as the other modes.

Disadvantages

- Integrity of the message is not maintained.

Disadvantages

- Integrity of the message is not maintained.
- Reuse of counter value, compromise the security.