Unit 2

Modern block cipher

By: Dr. Upma Jain

Block Cipher

- In block cipher a group of plaintext symbols of size m (m>1) are encrypted together to creating a group of cipher text of the same size.
- Same key is used for both encryption and decryption
- Examples:
  - Playfair cipher, hill cipher

Modern block ciphers

- Encrypts n bits block of plaintext or decrypts n bit block of cipher text.
- Same k bit key is used for both encryption and decryption.
- Decryption algorithm is inverse of encryption algorithm.
- It can be designed to act as a substitution cipher or transposition cipher.
- If message has fewer bits than n bits padding must be added to make it an n bit block.
- **Same idea is used as in traditional ciphers, except that the symbols to be substituted are or transposed are bits instead of characters.**

- Ex. How many padding bits are added to a message of 100 characters if 8 bit ASCII is used for encoding and the block cipher accepts blocks of 64 bit.

- 800 mod 64 = 32 bits are needed

- Modern block ciphers are designed as substitution ciphers because the characteristic of transposition (preserving no. of 0's and 1's) makes it vulnerable to exhaustive search attacks.

Example

- Suppose we have given a block cipher where n =16 bits. If there are 10 1's in the ciphertext, how many error and trial test one have to do in order to recover ciphertext from plaintext.
  - The ciphertext is designed as substitution cipher.
  - The ciphertext is designed as substitution cipher.

- Answer:
- For substitution cipher have to try all possible combinations 2^64
- In Transposition: 64!/[(10)! * (54)!], much lesser as compared to 2^64.

Block cipher design principles

Block Cipher Principles

- A block cipher is designed by considering its three critical aspects:
  - Number of Rounds
  - Design of Function F
  - Key Scheduling Algorithm

- **Number of Rounds**
  - The number of rounds judges the strength of the block cipher algorithm. It is considered that more is the number of rounds, difficult is for cryptanalysis to break the algorithm.
  - It is considered that even if the function F is relatively weak, the number of rounds would make the algorithm tough to break.

Design of Function F

- The function F of the block cipher must be designed such that it must be impossible for any cryptanalysis to unscramble the substitution. The criterion that strengthens the function F is its non-linearity.
- More the function F is nonlinear, more it would be difficult to crack it. Well, while designing the function F it should be confirmed that it has a good avalanche property which states that a change in one-bit of input must reflect the change in many bits of output.
- The Function F should be designed such that it possesses a bit independence criterion which states that the output bits must change independently if there is any change in the input bit.

Key Schedule Algorithm

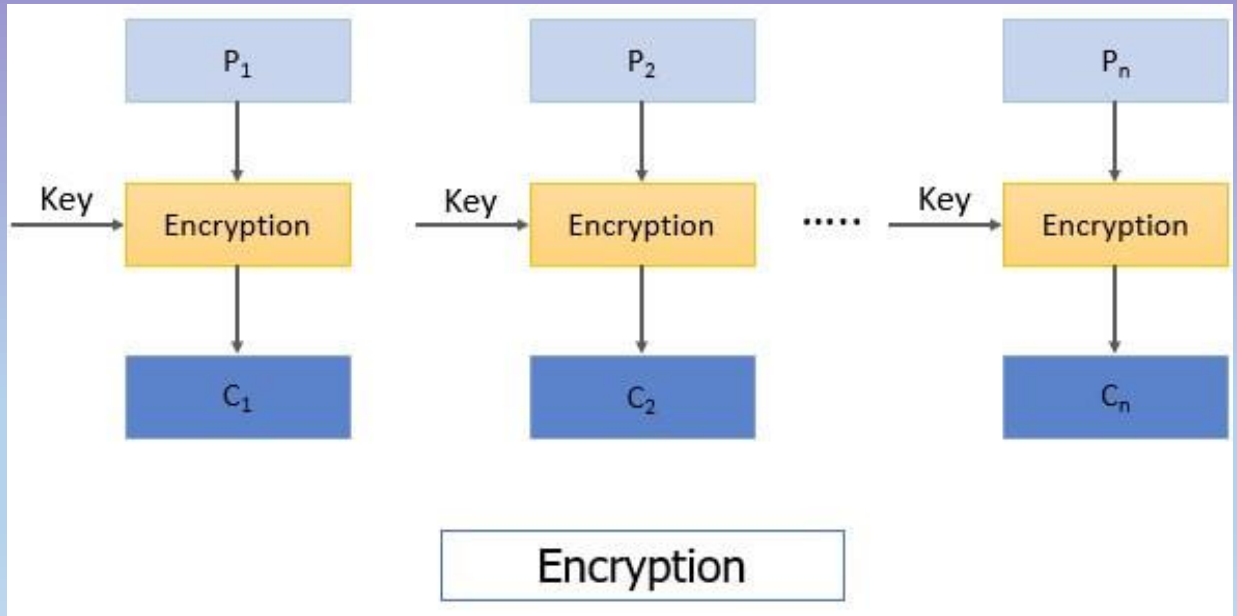
- It is suggested that the key schedule should confirm the strict avalanche effect and bit independence criterion.

Block Cipher Modes of Operation

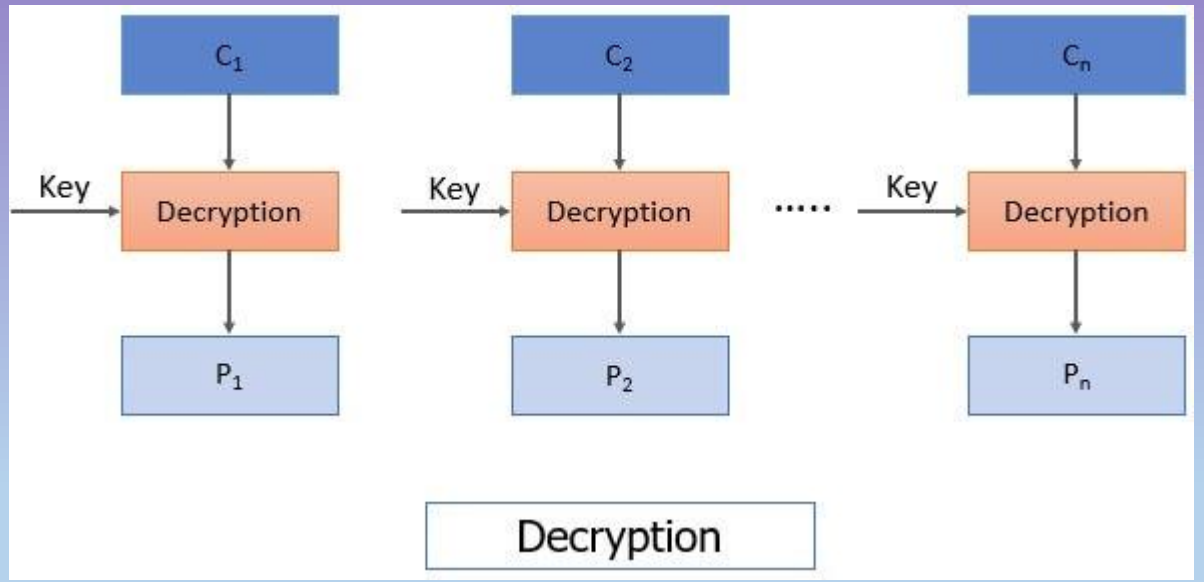
- There are five important block cipher modes of operation defined by NIST. These five modes of operation enhance the algorithm so that it can be adapted by a wide range of applications which uses block cipher for encryption.
- Electronic Code Book Mode
- Cipher Block Chaining Mode
- Cipher Feedback Mode
- Output Feedback Mode
- Counter Mode

Electronic Codebook Mode

- This is considered to be the easiest block cipher mode of operation. In electronic codebook mode (ECB) the plain text is divided into the blocks, each of 64-bit. Each block is encrypted one at a time to produce the cipher block. The same key is used to encrypt each block.



When the receiver receives the message i.e. ciphertext. This ciphertext is again divided into blocks, each of 64-bit and each block is decrypted independently one at a time to obtain the corresponding plain text block. Here also the same key is used to decrypt each block which was used to encrypt each block



Limitation

- As the same key used is to encrypt each block of plain text there arises an issue that for a repeating plain text block it would generate the same cipher and will ease the cryptanalysis to crack the algorithm. Hence, ECB is considered for encrypting the small messages which have a rare possibility of repeating text.

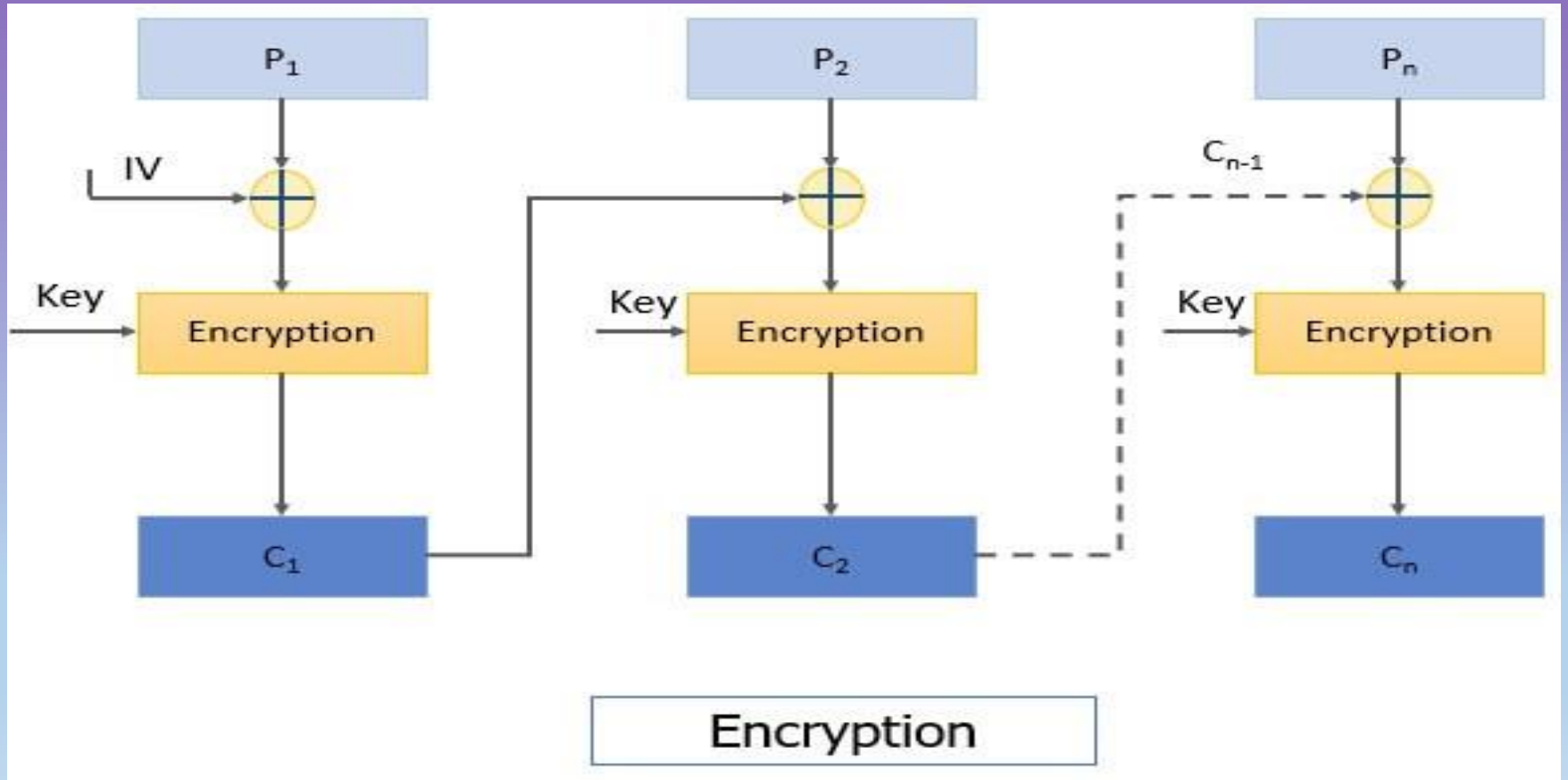
Cipher Block Chaining Mode

- To overcome the limitation of ECB i.e. the repeating block in plain text produces the same ciphertext, a new technique was required which is Cipher Block Chaining (CBC) Mode. CBC confirms that even if the plain text has repeating blocks its encryption won't produce same cipher block.
- To achieve totally different cipher blocks for two same plain text blocks **chaining** has been added to the block cipher. For this, the result obtained from the encryption of the first plain text block is fed to the encryption of the next plaintext box.
- In this way, each ciphertext block obtained is dependent on its corresponding current plain text block input and all the previous plain text blocks. But during the encryption of first plain text block, no previous plain text block is available so a random block of text is generated called **Initialization vector**.

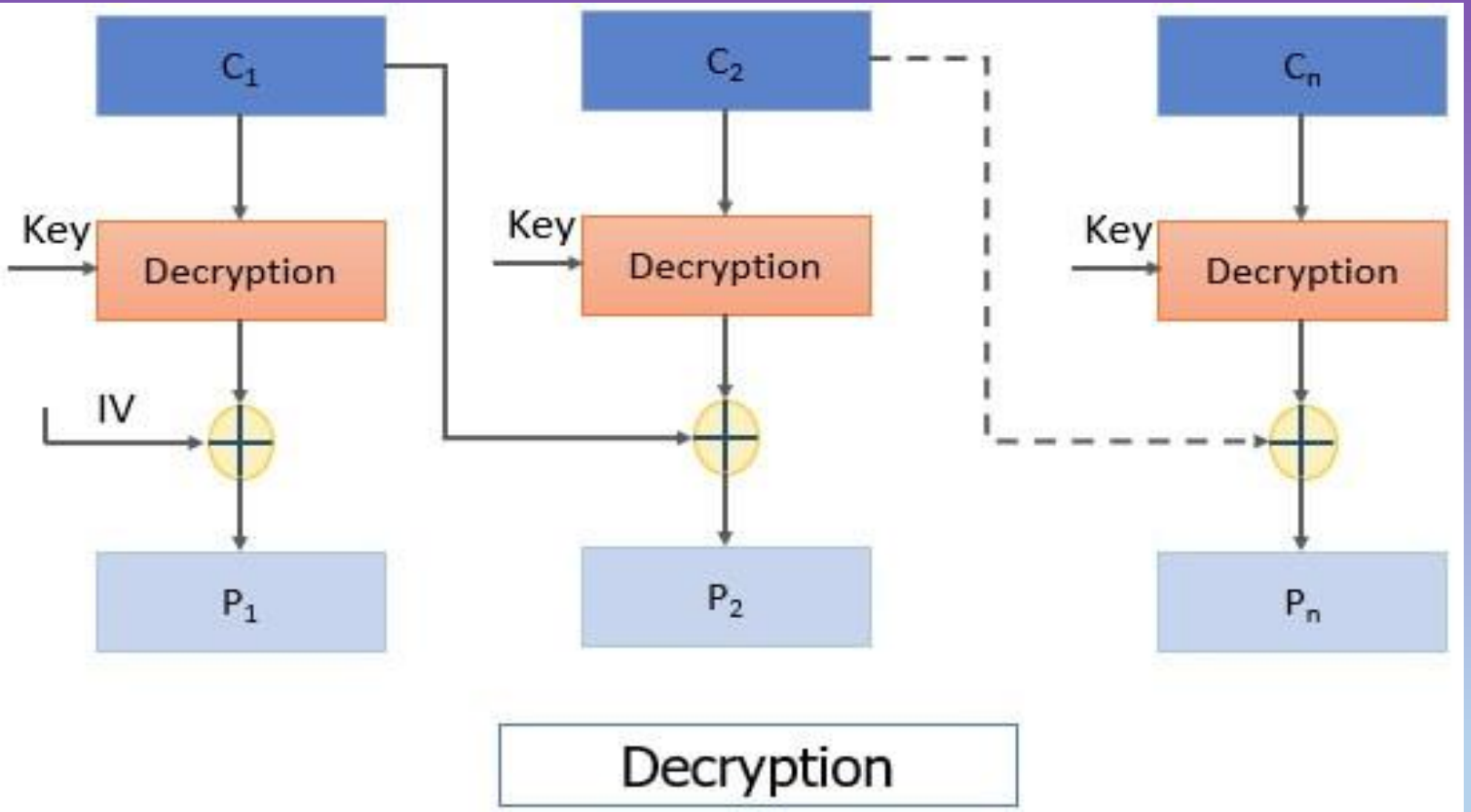
Encryption steps of CBC

- *Step 1:* The initialization vector and first plain text block are XORed and the result of XOR is then encrypted using the key to obtain the first ciphertext block.
- *Step 2:* The first ciphertext block is fed to the encryption of the second plain text block. For the encryption of second plain text block, first ciphertext block and second plain text block is XORed and the result of XOR is encrypted using the same key in step 1 to obtain the second ciphertext block.
- Similarly, the result of encryption of second plain text block i.e. the second ciphertext block is fed to the encryption of third plain text block to obtain third ciphertext block. And the process continues to obtain all the ciphertext blocks.

Encryption process of CBC



- **Decryption steps of CBC:**
- ***Step 1:*** The first ciphertext block is decrypted using the same key that was used for encrypting all plain text blocks. The result of decryption is then XORed with the initialization vector (IV) to obtain the first plain text block.
- ***Step 2:*** The second ciphertext block is decrypted and the result of decryption is XORed with the first ciphertext block to obtain the second plain text block. And the process continues till all plain text blocks are retrieved.


C1
C2
Cn
Key
Decryption
Key
Decryption
Key
Decryption
IV
P1
P2
Pn
Decryption

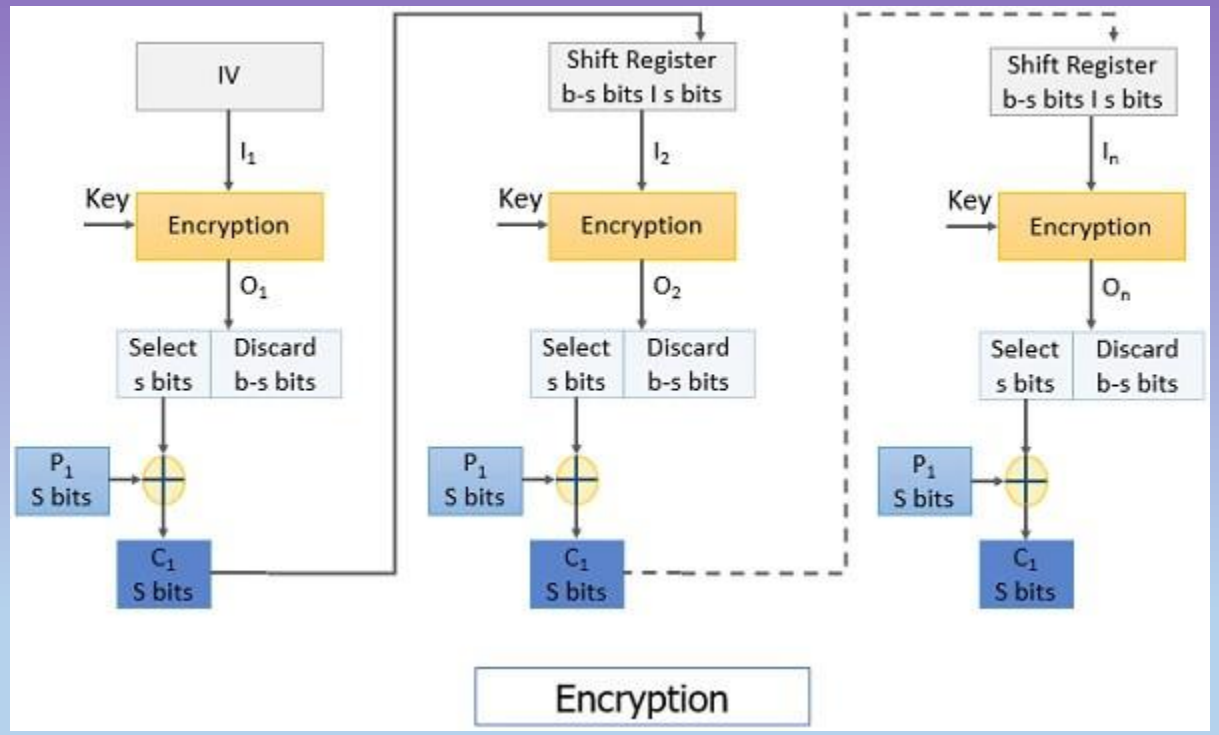
Limitation of CBC

- There was a limitation in CBC that if we have two identical messages and if we use the same IV for both the identical message it would generate the same ciphertext block.

Cipher Feedback Mode

- All applications may not be designed to operate on the blocks of data, some may be **character or bit-oriented**. Cipher feedback mode is used to operate on smaller units than blocks.
- Let us discuss the encryption steps in cipher feedback mode:
- ***Step 1:*** Here also we use initialization vector, IV is kept in the shift register and it is encrypted using the key.
- ***Step 2:*** The left most s bits of the encrypted IV is then XORed with the first fragment of the plain text of s bits. It produces the first ciphertext C1 of s bits.
- ***Step 3:*** Now the shift register containing initialization vector performs left shift by s bits and s bits C1 replaces the rightmost s bits of the initialization vector.
- Then again, the encryption is performed on IV and the leftmost s bit of encrypted IV is XORed with the second fragment of plain text to obtain s bit ciphertext C2.
- The process continues to obtain all ciphertext fragments.

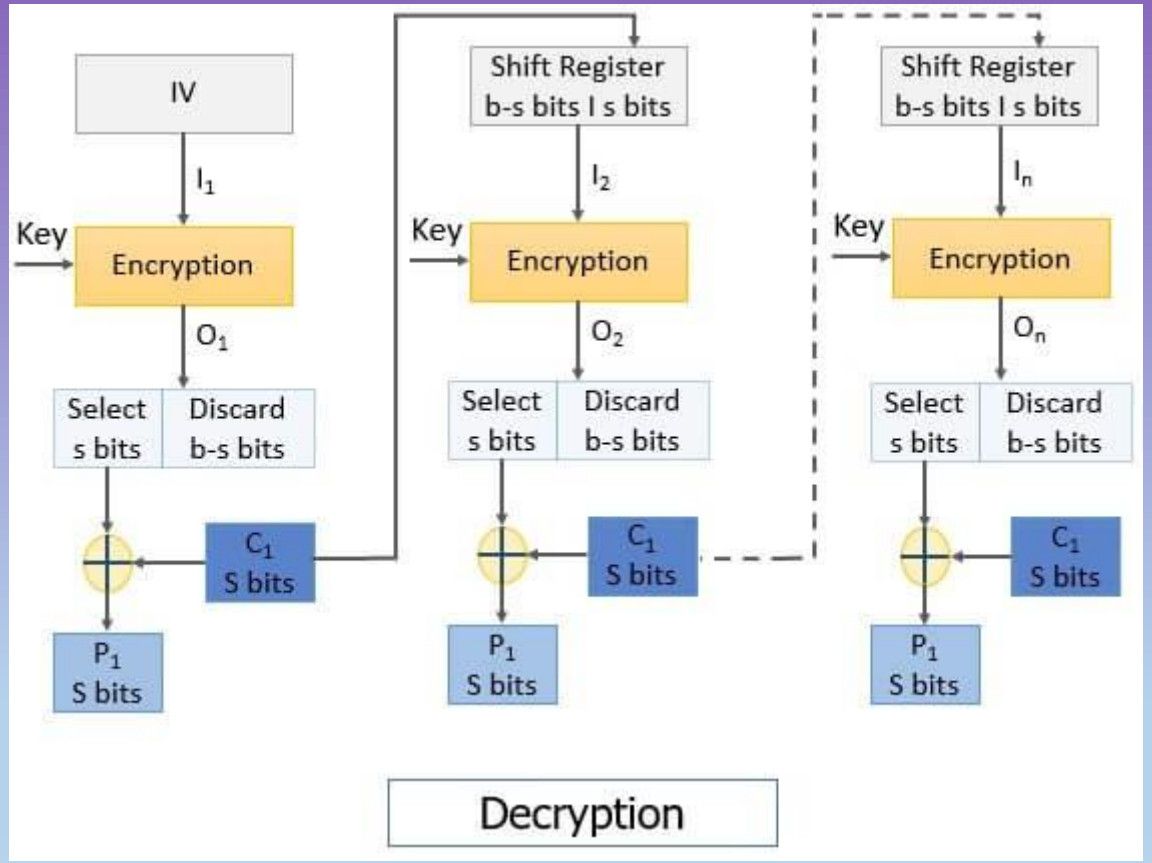
Encryption of CFM



Decryption Steps:

- ***Step 1:*** The initialization vector is placed in the shift register. It is encrypted using the same key.
- Keep a note that even in the **decryption process** the **encryption** algorithm is implemented instead of the decryption algorithm.
- Then from the encrypted IV s bits are XORed with the s bits ciphertext C1 to retrieve s bits plain text P1.
- ***Step 2:*** The IV in the shift register is left-shifted by s bits and the s bits C1 replaces the rightmost s bits of IV.
- The process continues until all plain text fragments are retrieved.

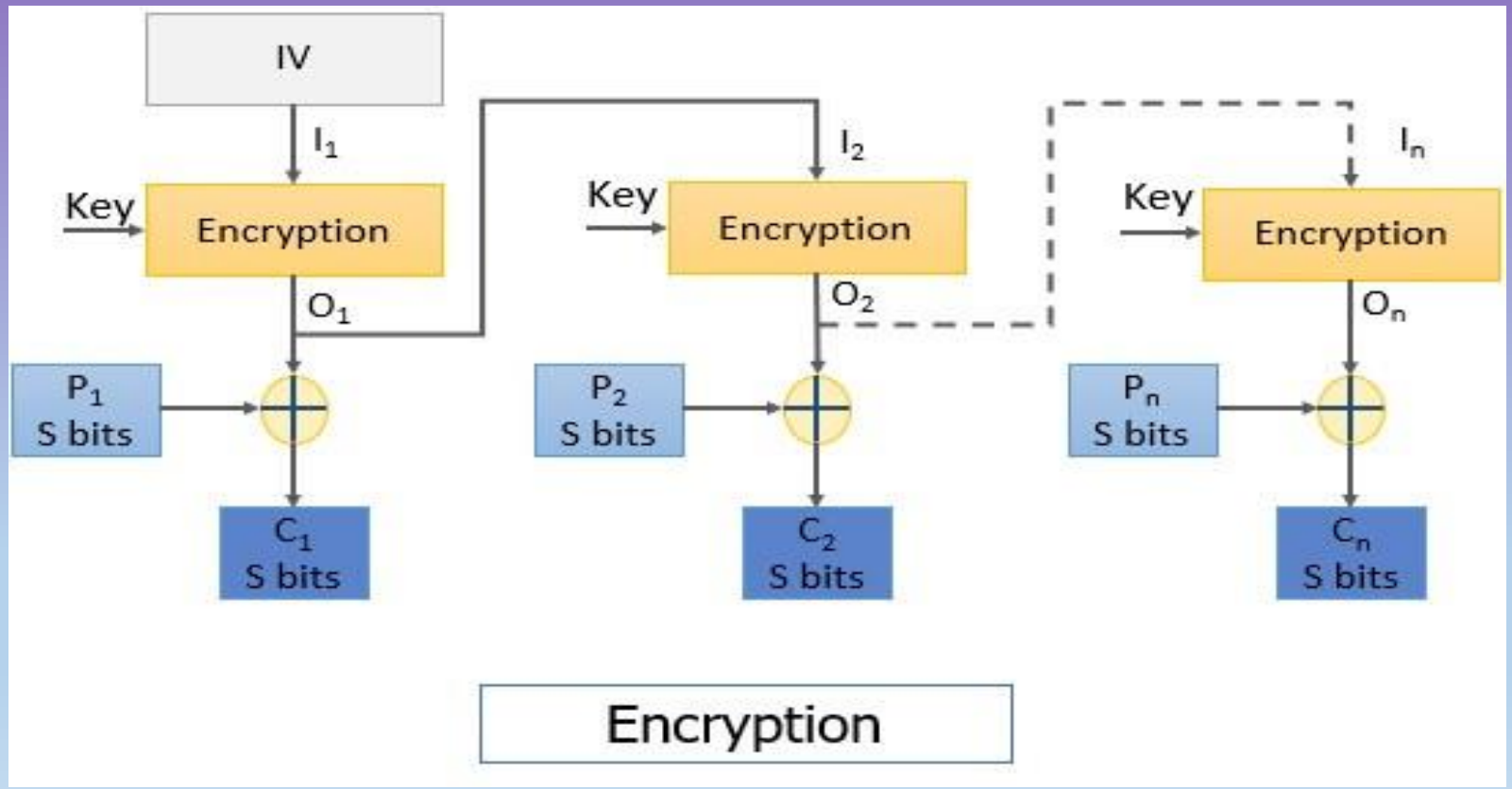
Decryption of CFM



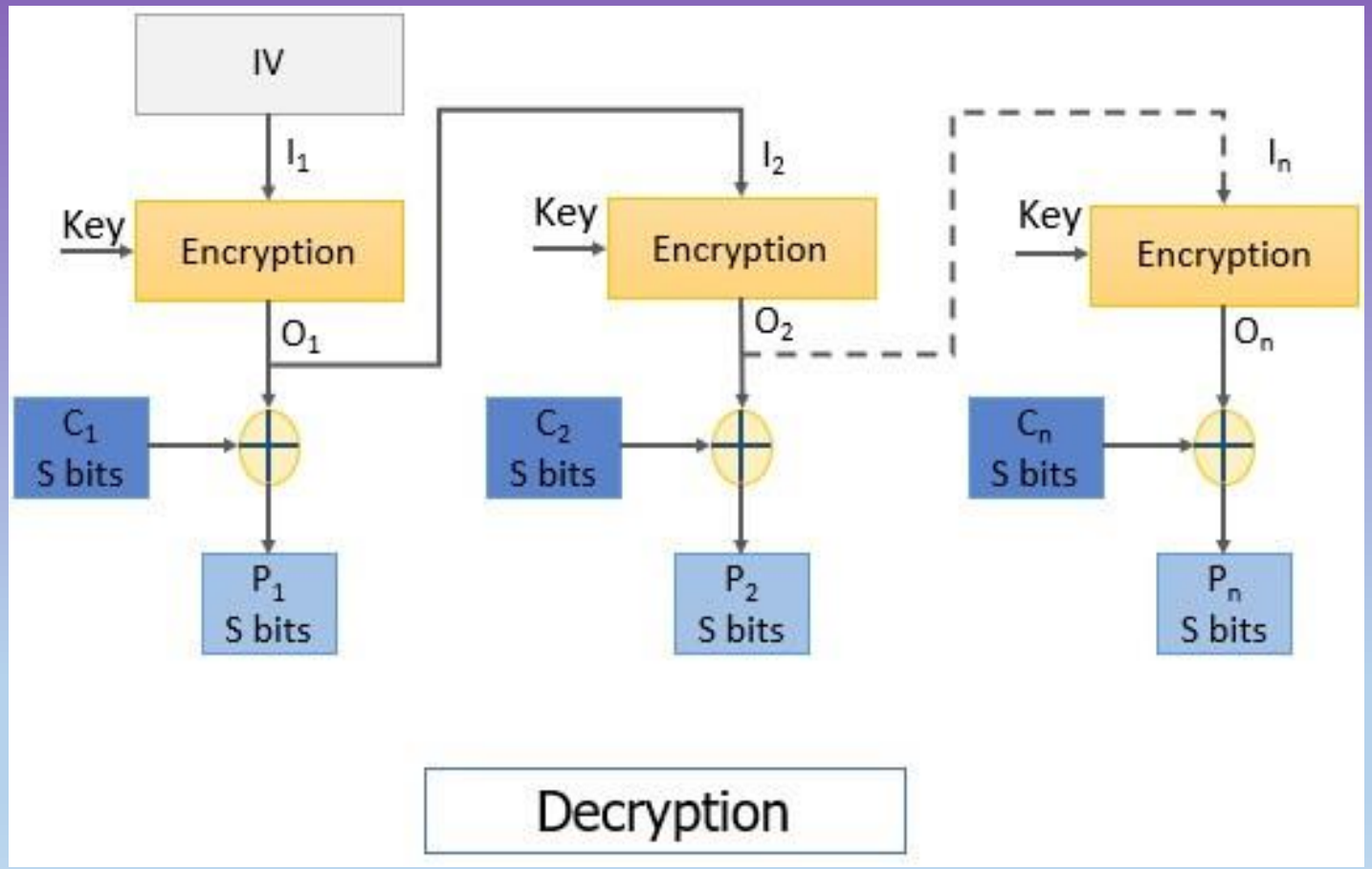
- It has a limitation that if there occur a bit error in any ciphertext C_i it would affect all the subsequent ciphertext units as C_i is fed to the encryption of next P_{i+1} to obtain C_{i+1}. In this way, bit error would propagate.

Output Feedback Mode

- The output feedback (OFB) mode is almost similar to the CFB. The difference between CFB and OFB is that unlike CFB, in OFB the encrypted IV is fed to the encryption of next plain text block. The other difference is that CFB operates on a stream of bits whereas OFB operates on the block of bits.
- Steps for encryption:
- ***Step 1:*** The initialization vector is encrypted using the key.
- ***Step 2:*** The encrypted IV is then XORed with the plain text block to obtain the ciphertext block.
- The encrypted IV is fed to the encryption of next plain text block as you can see in the image below.


IV
I_1
Key
Encryption
O_1
P_1
S bits
C_1
S bits
I_2
Key
Encryption
O_2
P_2
S bits
C_2
S bits
I_n
Key
Encryption
O_n
P_n
S bits
C_n
S bits
Encryption

- Steps for decryption:
- ***Step 1:*** The initialization vector is encrypted using the same key used for encrypting all plain text blocks.
- Note: In the **decryption process** also the **encryption** function is implemented.
- ***Step2:*** The encrypted IV is then XORed with the ciphertext block to retrieve the plain text block.
- The encrypted IV is also fed to the decryption process of the next ciphertext block.
- The process continues until all the plain text blocks are retrieved.

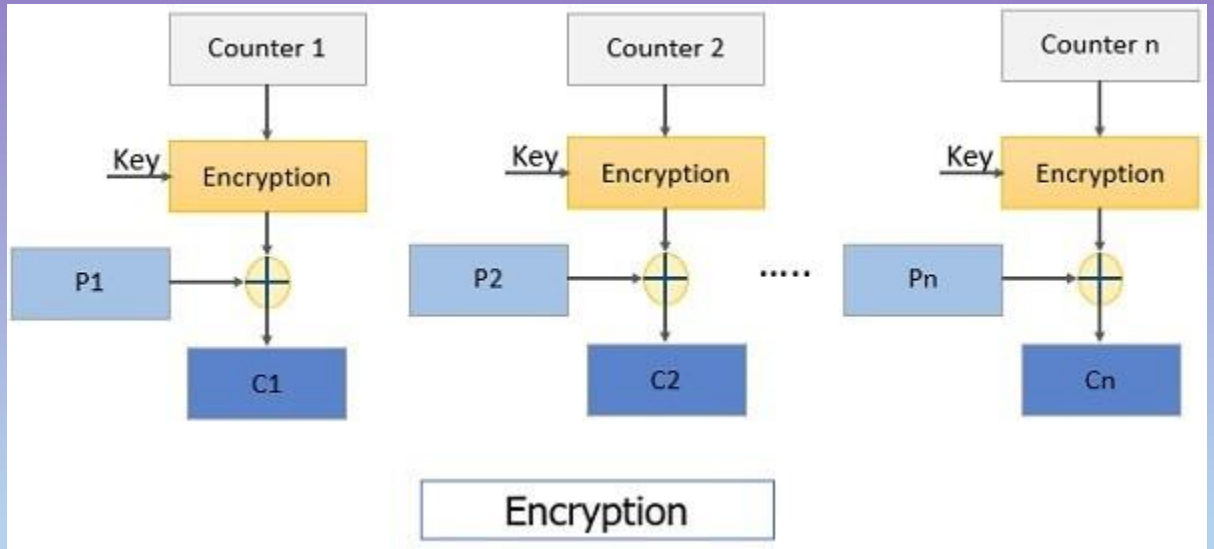

IV
I_1
Key
Encryption
O_1
C_1
S bits
P_1
S bits
I_2
Key
Encryption
O_2
C_2
S bits
P_2
S bits
I_n
Key
Encryption
O_n
C_n
S bits
P_n
S bits
Decryption

- The advantage of OFB is that it protects the propagation of bit error means the if there occurs a bit error in C1 it will only affect the retrieval of P1 and won’t affect the retrieval of subsequent plain text blocks.

5. Counter Mode

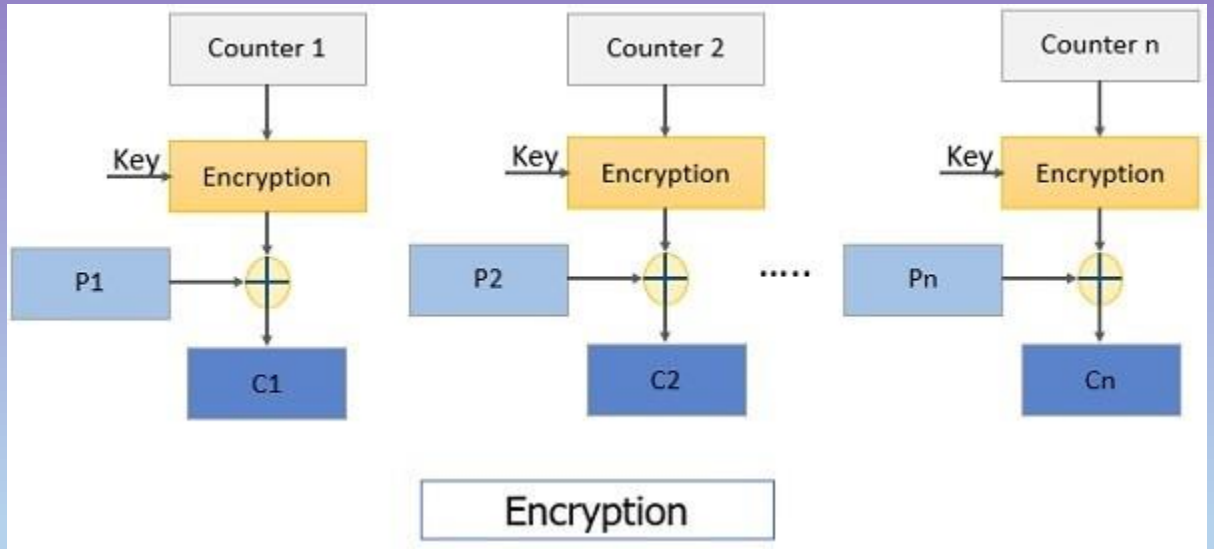
- It is similar to OFB but there is no feedback mechanism in counter mode. Nothing is being fed from the previous step to the next step instead it uses a sequence of number which is termed as a **counter** which is input to the encryption function along with the key. After a plain text block is encrypted the counter value increments by 1.
- Steps of encryption:
- ***Step1:*** The counter value is encrypted using a key.
- ***Step 2:*** The encrypted counter value is XORed with the plain text block to obtain a ciphertext block.
- To encrypt the next subsequent plain text block the counter value is incremented by 1 and step 1 and 2 are repeated to obtain the corresponding ciphertext.
- The process continues until all plain text block is encrypted.

Counter mode encryption



- Steps for decryption:
- ***Step1:*** The counter value is encrypted using a key.
- **Note:** Encryption function is used in the decryption process. The same counter values are used for decryption as used while encryption.
- ***Step 2:*** The encrypted counter value is XORed with the ciphertext block to obtain a plain text block.

Counter mode decryption



- To decrypt the next subsequent ciphertext block the counter value is incremented by 1 and step 1 and 2 are repeated to obtain corresponding plain text.
- The process continues until all ciphertext block is decrypted.
- So, this is all about Block cipher, its designing principles and its mode of operation

Evaluation Criteria For AES

The Origins of AES

- In 1999, NIST issued a new version of its DES standard (FIPS PUB 46-3) that indicated that DES should only be used for legacy systems and that triple DES (3DES) be used.
- 3DES has two attractions that assure its widespread use over the next few years.
- First, with its 168-bit key length, it overcomes the vulnerability to brute-force attack of DES.
- Second, the underlying encryption algorithm in 3DES is the same as in DES. This algorithm has been subjected to more scrutiny than any other encryption algorithm over a longer period of time, and no effective cryptanalytic attack based on the algorithm rather than brute force has been found.
- Accordingly, there is a high level of confidence that 3DES is very resistant to cryptanalysis. If security were the only consideration, then 3DES would be an appropriate choice for a standardized encryption algorithm for decades to come.

- The **principal drawback of 3DES is that the algorithm is relatively sluggish in software**. The original DES was designed for mid-1970s hardware implementation and does not produce efficient software code. 3DES, which has three times as many rounds as DES, is correspondingly slower.
- A secondary drawback is t**hat both DES and 3DES use a 64-bit block size.** For reasons of both efficiency and security, a larger block size is desirable.
- Because of these drawbacks, 3DES is not a reasonable candidate for long-term use.
- As a replacement, NIST in 1997 issued a call for proposals for a new Advanced Encryption Standard (AES), which should have a security strength equal to or better than 3DES and significantly improved efficiency.
- In addition to these general requirements, **NIST specified that AES must be a symmetric block cipher with a block length of 128 bits and support for key lengths of 128, 192, and 256 bits.**

- These criteria span the range of concerns for the practical application of modern symmetric block ciphers. In fact, two set of criteria evolved.
- The three categories of criteria were as follows:
- **Security:** This refers to the effort required to cryptanalyze an algorithm. The emphasis in the evaluation was on the practicality of the attack. Because the minimum key size for AES is 128 bits, brute-force attacks with current and projected technology were considered impractical. Therefore, the emphasis, with respect to this point, is cryptanalysis other than a brute-force attack.
- **Cost:** NIST intends AES to be practical in a wide range of applications. Accordingly, AES must have high computational efficiency, so as to be usable in high-speed applications, such as broadband links.
- **Algorithm and implementation characteristics:** This category includes a variety of considerations, including flexibility; suitability for a variety of hardware and software implementations; and simplicity, which will make an analysis of security more straightforward.

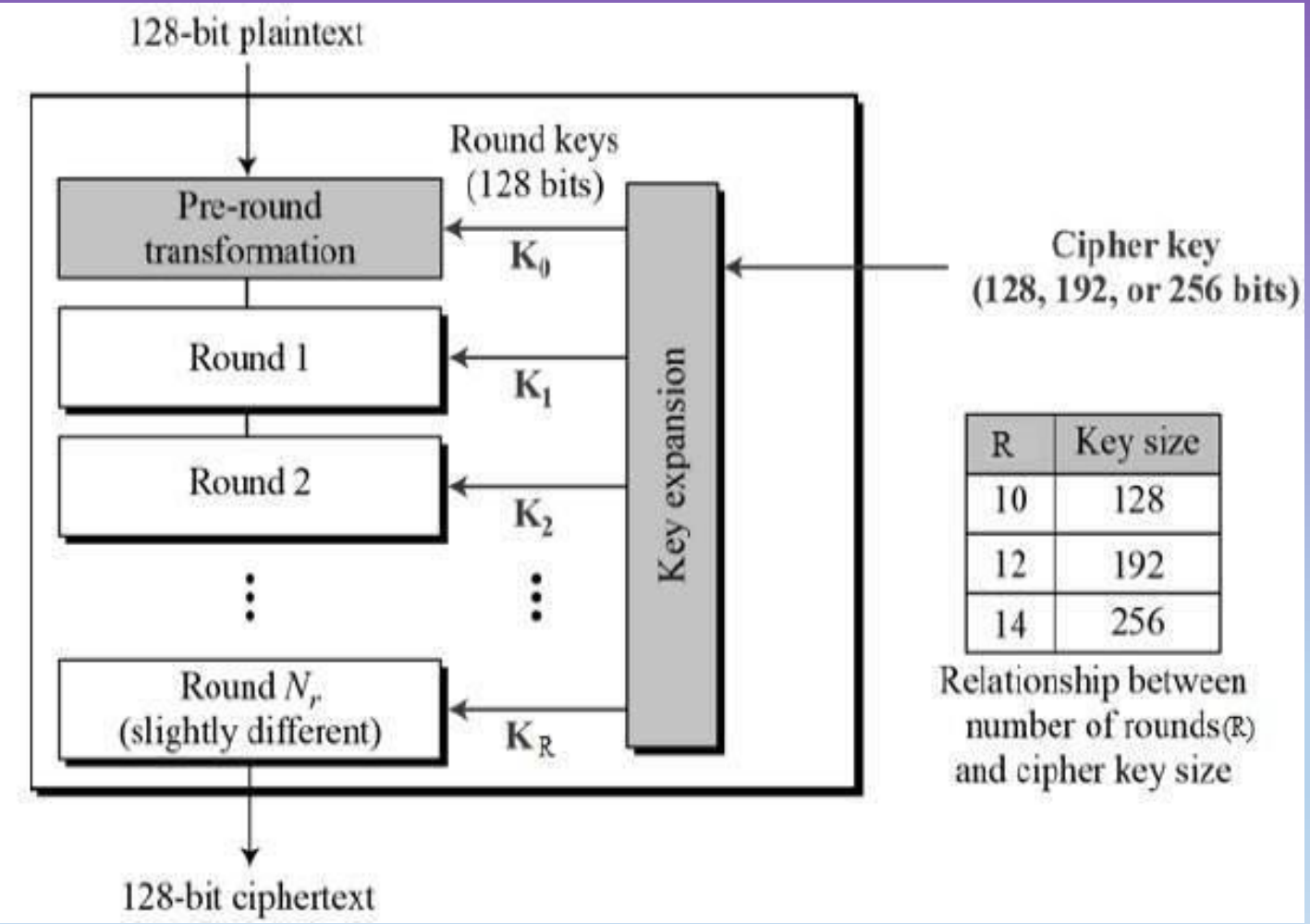
Advanced Encryption Standard (AES)

- Advanced Encryption Standard (AES)is a specification for the encryption of electronic data established by the U.S National Institute of Standards and Technology (NIST) in 2001. AES is widely used today as it is a much stronger than DES and triple DES despite being harder to implement.
- Points to remember
- AES is a block cipher.
- The key size can be 128/192/256 bits.
- Encrypts data in blocks of 128 bits each.
- That means it takes 128 bits as input and outputs 128 bits of encrypted cipher text as output. AES relies on substitution-permutation network principle which means it is performed using a series of linked operations which involves replacing and shuffling of

Working of the cipher :

- Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.
- The number of rounds depends on the key length as follows :
- 128 bit key – 10 rounds
- 192 bit key – 12 rounds
- 256 bit key – 14 rounds

The schematic of AES structure is given in the following illustration –

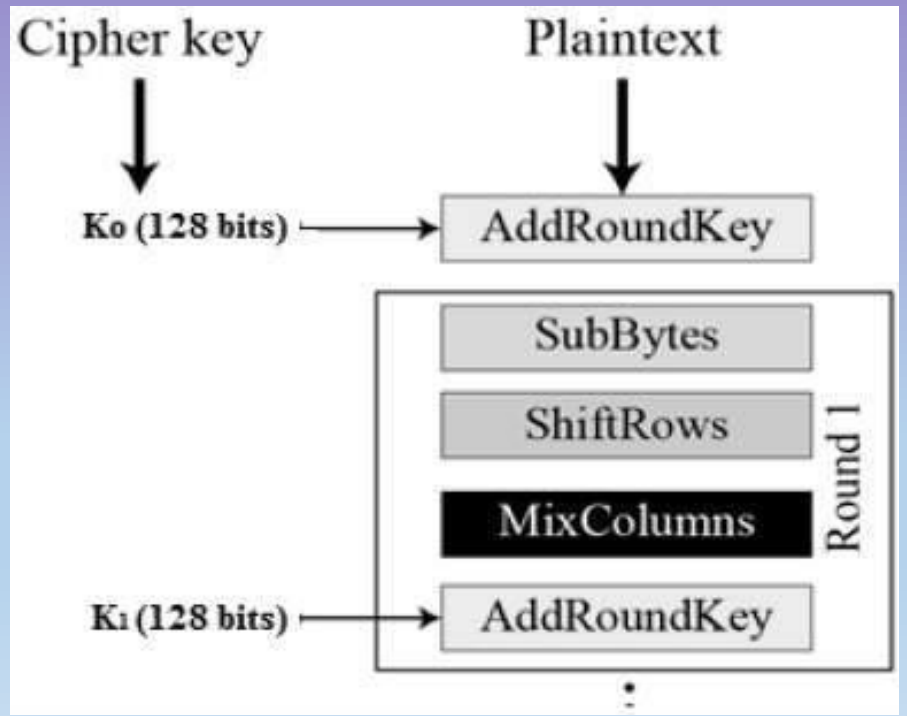


R	Key size
10	128
12	192
14	256

Relationship between number of rounds (R) and cipher key size

Encryption Process

- Here, we restrict to description of a typical round of AES encryption. Each round comprise of four sub-processes. The first round process is depicted below –



Encryption :

AES considers each block as a 16 byte (4 byte x 4 byte = 128) grid in a column major arrangement.

```
[ b0 | b4 | b8 | b12 |
| b1 | b5 | b9 | b13 |
| b2 | b6 | b10| b14 |
| b3 | b7 | b11| b15 ]
```

Each round comprises of 4 steps :

- SubBytes
- ShiftRows
- MixColumns
- Add Round Key

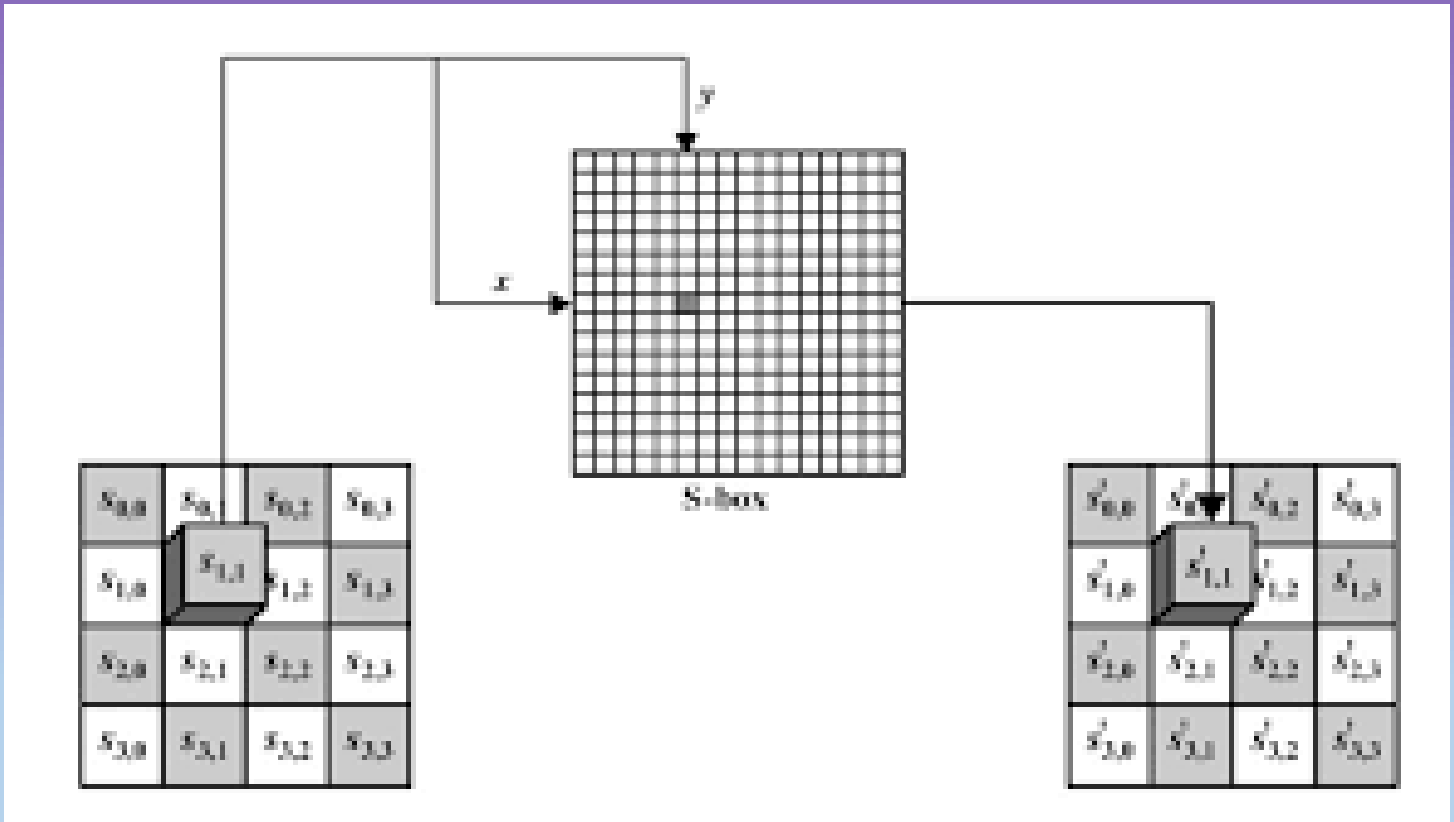
The last round doesn’t have the MixColumns round.

Subbytes

- The SubBytes does the substitution and ShiftRows and MixColumns performs the permutation in the algorithm.
- This step implements the substitution.
- In this step each byte is substituted by another byte. Its performed using a lookup table also called the S-box. This substitution is done in a way that a byte is never substituted by itself and also not substituted by another byte which is a compliment of the current byte. The result of this step is a 16 byte (4 x 4) matrix like before.

The next two steps implement the permutation.

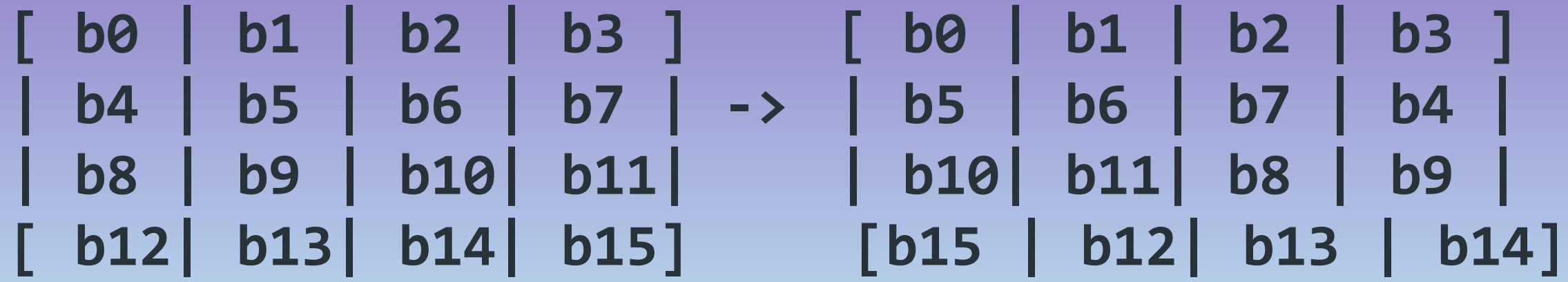
Substitute



ShiftRows :

- This step is just as it sounds. Each row is shifted a particular number of times.
- The first row is not shifted
- The second row is shifted once to the left.
- The third row is shifted twice to the left.
- The fourth row is shifted thrice to the left.
- (A left circular shift is performed.)

shiftrows


[b0 | b1 | b2 | b3]
| b4 | b5 | b6 | b7 |
| b8 | b9 | b10| b11|
[b12| b13| b14| b15]
->
[b0 | b1 | b2 | b3]
| b5 | b6 | b7 | b4 |
| b10| b11| b8 | b9 |
[b15 | b12| b13 | b14]

MixColumns :

- This step is basically a matrix multiplication. Each column is multiplied with a specific matrix and thus the position of each byte in the column is changed as a result.

This step is skipped in the last round.

$$\begin{bmatrix} c0 \\ c1 \\ c2 \\ c3 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} b0 \\ b1 \\ b2 \\ b3 \end{bmatrix}$$

- **Add Round Keys :**
  Now the resultant output of the previous stage is XOR-ed with the corresponding round key. Here, the 16 bytes is not considered as a grid but just as 128 bits of data.