Block Ciphers Principles

Stream Ciphers and Block Ciphers:

A **stream cipher** is one that encrypts a digital data stream one bit or one byte at a time. Examples of classical stream ciphers are the auto keyed Vigenère cipher.

A **block cipher** is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length. Typically, a block size of 64 or 128 bits is used.

Differences between Stream Cipher and Block Cipher

Block Cipher	Stream Cipher
Block cipher converts the plaintext into ciphertext by taking plain text's block at a time	Stream Cipher converts the plaintext into cipher text by taking 1 bit plaintext at a time.
Block cipher uses either 64bits or more than 64bits	While stream cipher uses
Block cipher uses confusion as well as diffusion	While stream cipher uses only confusion
The algorithm modes which are used in block cipher are ECB(Electronic Code block) and CBc(Cipher Block Chainig)	The algorithm modes which are used in block cipher are CFB and OFB
Operates on fixed length of block data	Encrypts data one bit at a time

Motivation of Feistel Cipher Structure

A block cipher operates on a plaintext block of *n* bits to produce a ciphertext block of *n* bits.

There are 2^n possible different plaintext blocks and, for the encryption to be reversible (i.e., for decryption to be possible), each must produce a unique ciphertext block

Reversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	00
11	01

Irreversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	01
11	01

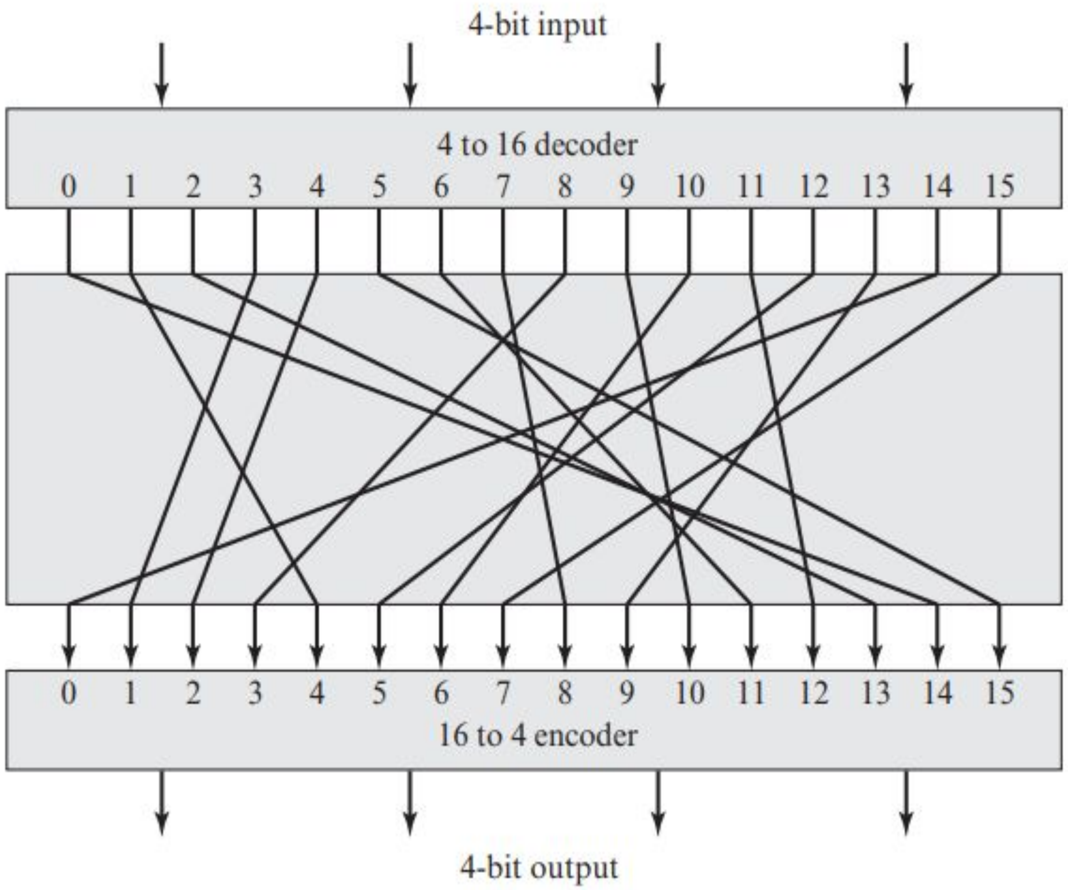


Figure 4.2 General n-bit-n-bit Block Substitution (shown with $n = 4$)

Table 4.1 Encryption and Decryption Tables for Substitution Cipher of Figure 4.2

Plaintext	Ciphertext
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

Ciphertext	Plaintext
0000	1110
0001	0011
0010	0100
0011	1000
0100	0001
0101	1100
0110	1010
0111	1111
1000	0111
1001	1101
1010	1001
1011	0110
1100	1011
1101	0010
1110	0000
1111	0101

In this case, using this straightforward method of defining the key, the required key length is (4 bits) x (16 rows) = 64 bits.

In general, for an *n*-bit ideal block cipher, the length of the key defined in this fashion is n x 2^n bits. For a 64-bit block, which
is a desirable length to thwart statistical attacks, the required key length is $64 \times 2^{64} = 2^{70} \approx 10^{21}$ bits.

For example, suppose we define the mapping in terms of a set of linear equations. In the case of *n* = 4, we have

$$y1 = k11x1 + k12x2 + k13x3 + k14x4$$

$$y2 = k21x1 + k22x2 + k23x3 + k24x4$$

$$y3 = k31x1 + k32x2 + k33x3 + k34x4$$

$$y4 = k41x1 + k42x2 + k43x3 + k44x4$$

Feistel Cipher

Feistel Cipher model is a structure or a design used to develop many block ciphers such as DES. Feistel cipher may have invertible, non-invertible and self invertible components in its design. A separate key is used for each round. However same round keys are used for encryption as well as decryption.

- **Substitution**: Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements.

- **Permutation**: A sequence of plaintext elements is replaced by a permutation of that sequence. That is, no elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence is changed.

The Feistel Cipher

Feistel's is a practical application of proposal by Claude Shannon to develop a product cipher that alternates **Confusion** and **Diffusion** functions.

Diffusion:-: It is a method to frustrate cryptanalysis. In diffusion, the statistical structure of the plaintext is dissipated into long range statistics of the ciphertext. This is achieved by having each plaintext digit affect the value of many ciphertext digits; generally this is equivalent to having each ciphertext digit be affected by many plaintext digits. An example of diffusion is to encrypt a message $M = m1, m2, m3,...$ of characters with an averaging operation:

$$y_n = \left(\sum_{i=1}^{k} m_{n+i}\right) \bmod 26$$

adding k successive letters to get a ciphertext letter yn. One can show that the statistical structure of the plaintext has been dissipated.

In a binary block cipher, diffusion can be achieved by repeatedly performing some permutation on the data followed by applying a function to that permutation; the effect is that bits from different positions in the original plaintext contribute to a single bit of ciphertext.

The Feistel Cipher

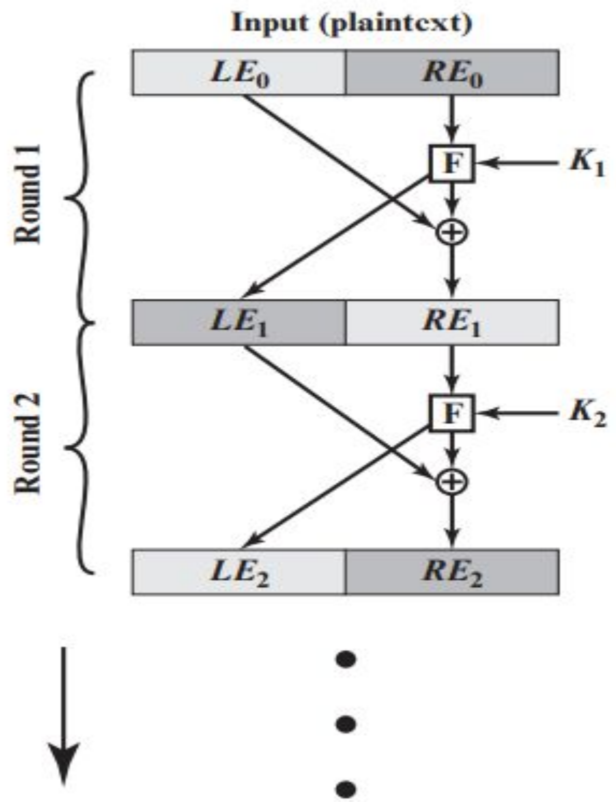
Confusion: Confusion defines making the relationship between the key and the cipher as difficult and as included as possible. In other words, the technique provides that the ciphertext provides no clue about the plaintext.

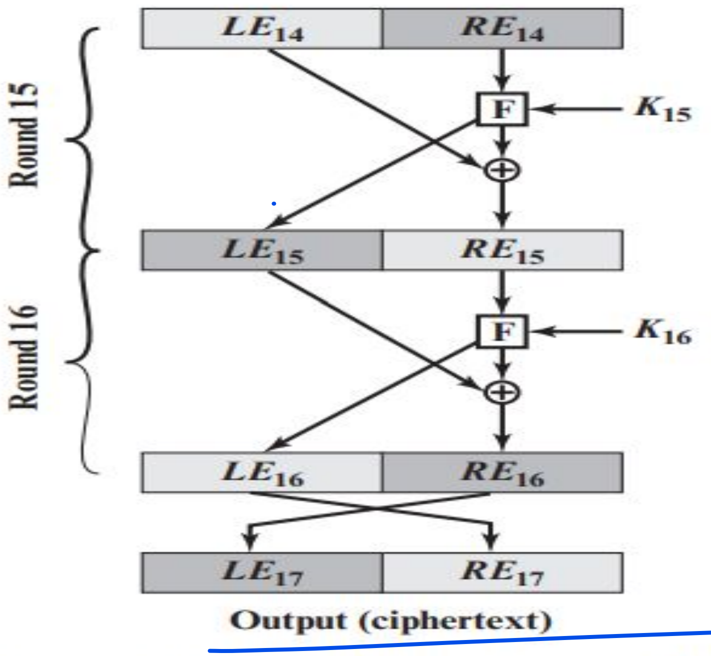
In confusion, it should influence the complete cipher text or change should appear on the complete cipher text and the relationship between the data of the ciphertext and the value of the encryption key is made difficult. It is achieved by substitution.

```
Confusion = Substitution
a --> b
Example : Caesar Cipher


Diffusion = Transposition or Permutation
abcd --> dacb
Example : DES
```

Feistel Cipher Encryption





Realization of Feistel network

Block size: Larger block sizes mean greater security (all other things being equal) but reduced encryption/decryption speed for a given algorithm. The greater security is achieved by greater diffusion

Traditionally, a block size of 64 bits has been considered a reasonable tradeoff and was nearly universal in block cipher design. However, the new AES uses a 128-bit block size.

Key size: Larger key size means greater security but may decrease encryption/decryption speed. The greater security is achieved by greater resistance to brute-force attacks and greater confusion.
Key sizes of 64 bits or less are now widely considered to be inadequate, and 128 bits has become a common size.

Number of rounds: The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security. A typical size is 16 rounds.

Subkey generation algorithm: Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.

Realization of Feistel network

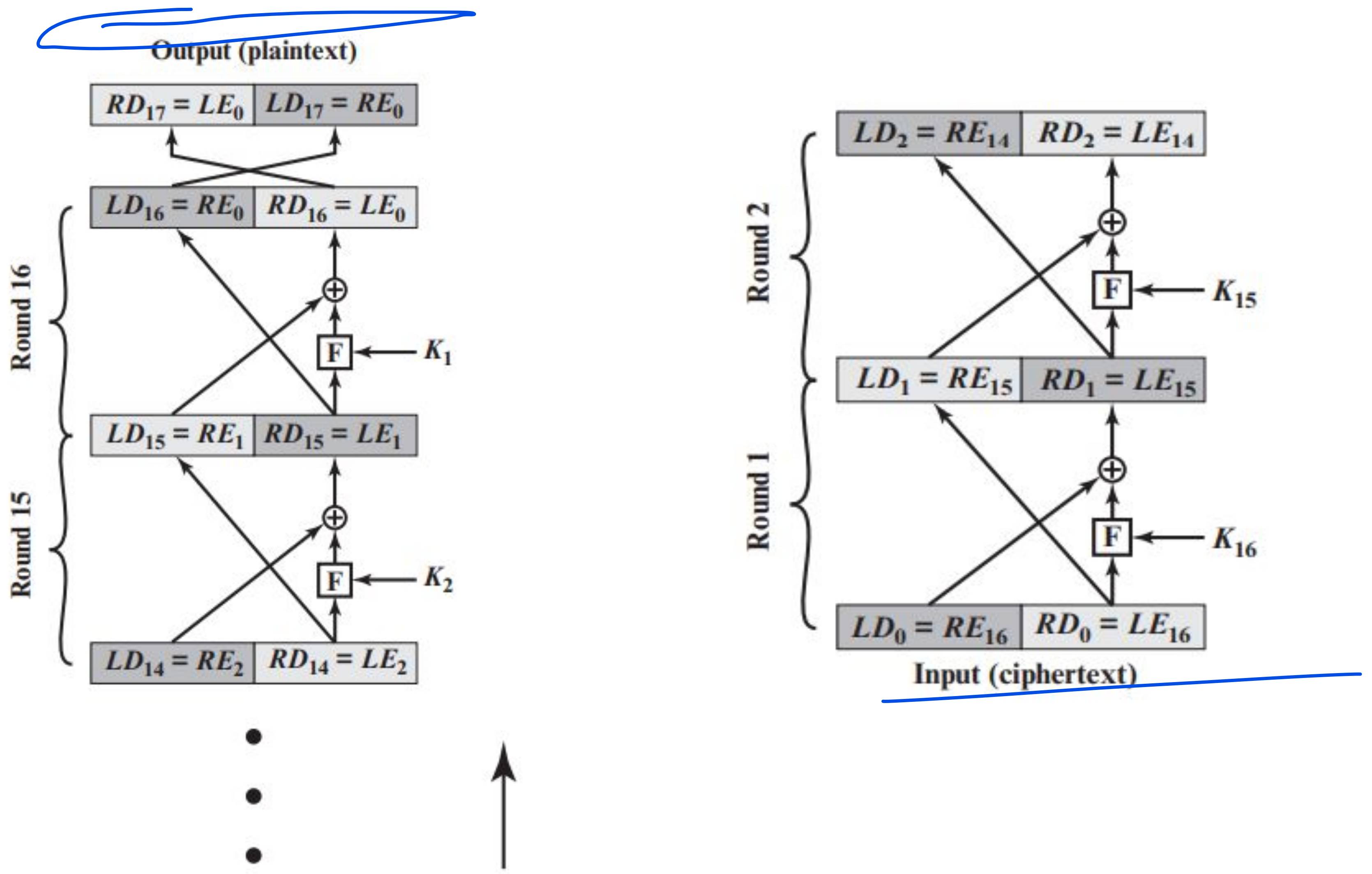
Round function: Again, greater complexity generally means greater resistance to cryptanalysis.

Fast software encryption/decryption: In many cases, encryption is embedded in applications or utility functions in such a way as to preclude a hardware implementation. Accordingly, the speed of execution of the algorithm becomes a concern.

Ease of analysis: Although we would like to make our algorithm as difficult as possible to cryptanalyze, there is great benefit in making the algorithm easy to analyze.

That is, if the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and therefore develop a higher level of assurance as to its strength. DES, for example, does not have an easily analyzed functionality.

Feistel Decryption algorithm



Feistel Decryption algorithm

Now we would like to show that the output of the first round of the decryption process is equal to a 32-bit swap of the input to the sixteenth round of the encryption process. First, consider the encryption process. We see that

$$LE_{16} = RE_{15}$$
$$RE_{16} = LE_{15} \oplus F(RE_{15}, K_{16})$$

On the decryption side,

$$LD_1 = RD_0 = LE_{16} = RE_{15}$$
$$RD_1 = LD_0 \oplus F(RD_0, K_{16})$$

Feistel Decryption

The XOR has the following properties:

$$[A \oplus B] \oplus C = A \oplus [B \oplus C]$$
$$D \oplus D = 0$$
$$E \oplus 0 = E$$

Thus, we have $LD_1 = RE_{15}$ and $RD_1 = LE_{15}$. Therefore, the output of the first round of the decryption process is $RE_{15} \| LE_{15}$, which is the 32-bit swap of the input to the sixteenth round of the encryption. This correspondence holds all the way through the 16 iterations, as is easily shown. We can cast this process in general terms. For the ith iteration of the encryption algorithm,

$$LE_i = RE_{i-1}$$
$$RE_i = LE_{i-1} \oplus \mathrm{F}(RE_{i-1}, K_i)$$

Rearranging terms:

$$RE_{i-1} = LE_i$$
$$LE_{i-1} = RE_i \oplus \mathrm{F}(RE_{i-1}, K_i) = RE_i \oplus \mathrm{F}(LE_i, K_i)$$

DES Encryption

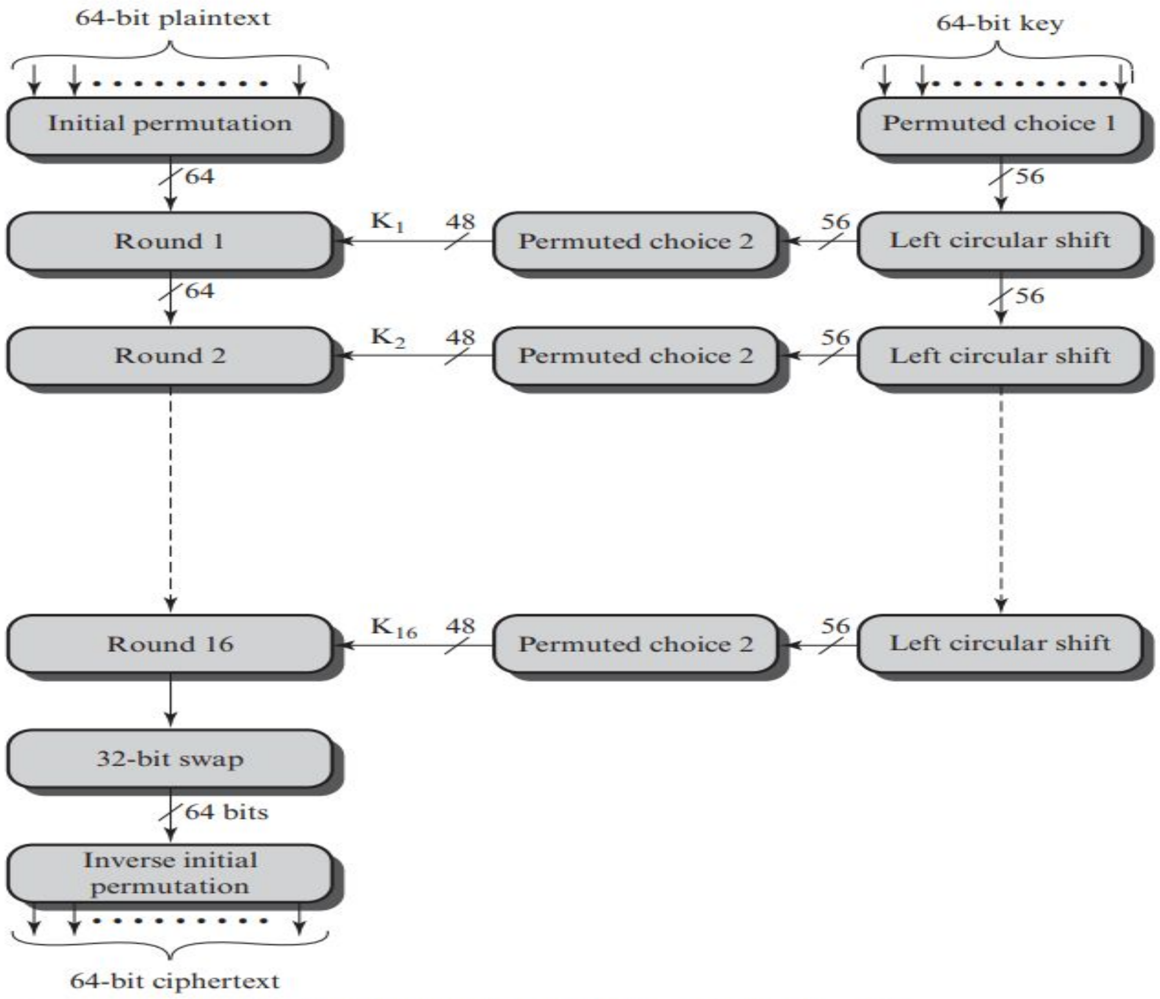
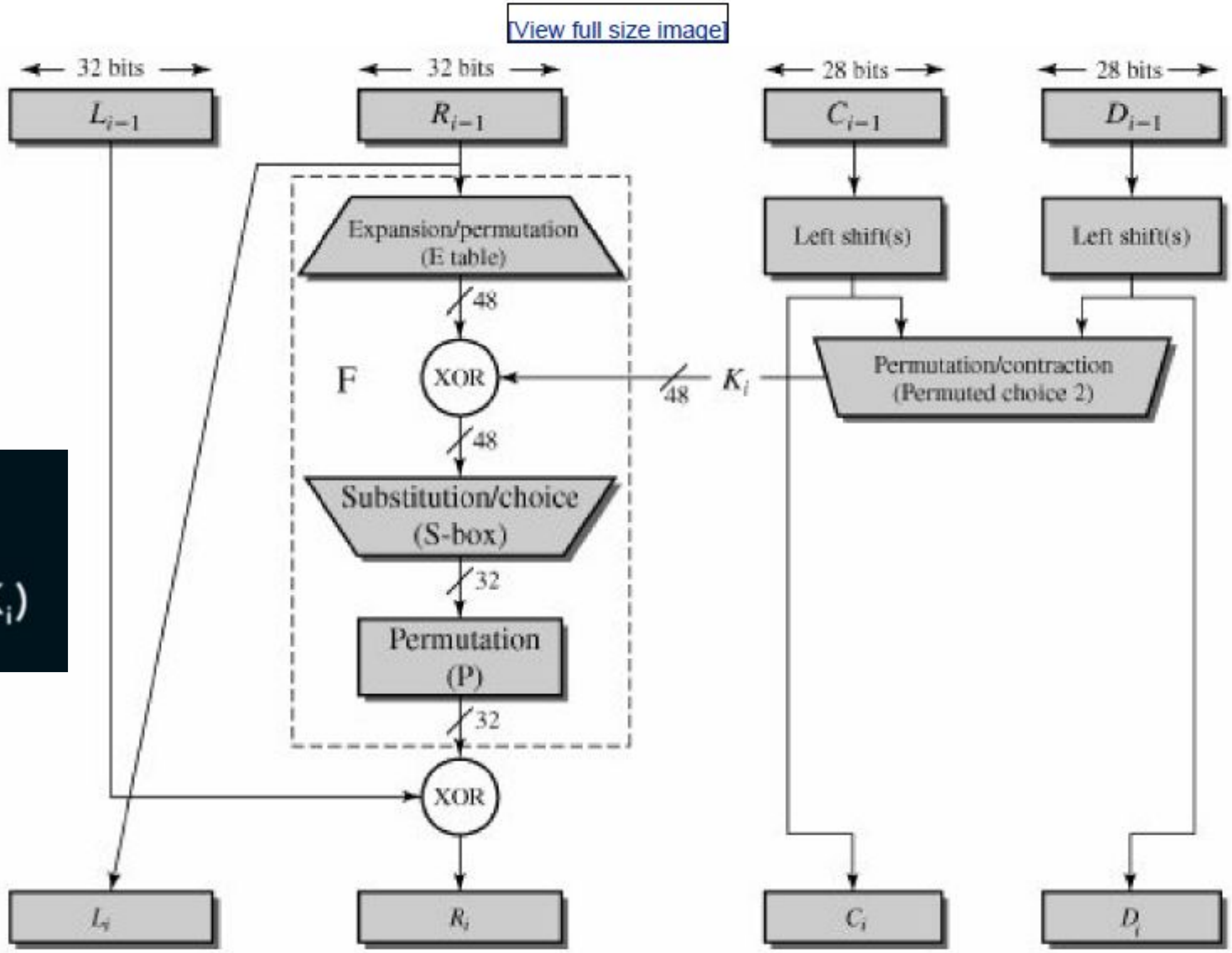


Figure 4.5 General Depiction of DES Encryption Algorithm

Data Encryption Standard(DES Algorithm)

Fig: Single Round of DES algorithm



$L_i = R_{i-1}$

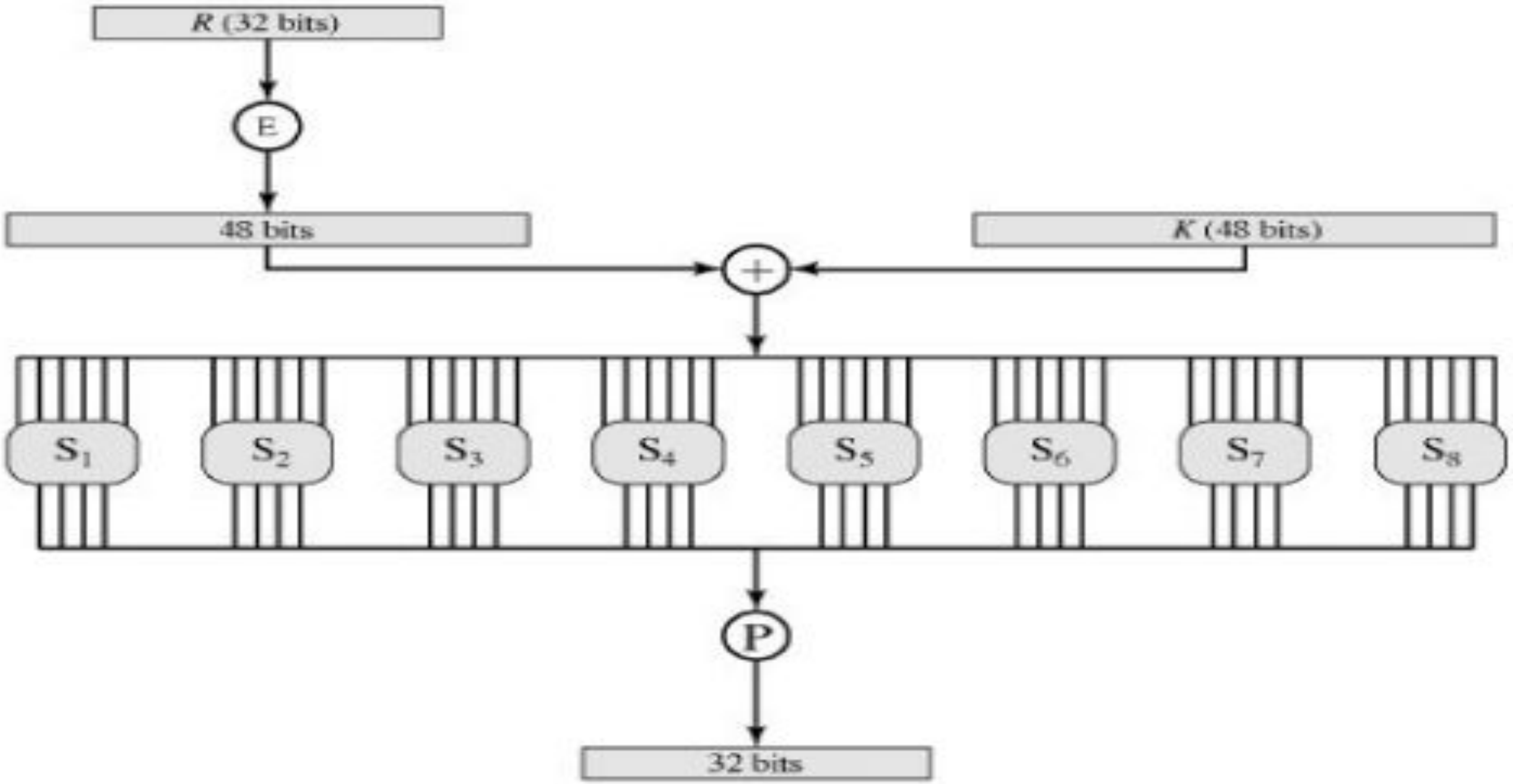
$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$

Data Encryption Standard(DES Algorithm)

Fig: Calculation of F(R,K)—F function

Figure 3.6. Calculation of F(R, K)

(This item is displayed on page 78 in the print version)



Data Encryption Standard(DES Algorithm)

Box S_1

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
01	0	15	7	4	14	2	13	1	10	6	12	11	6	5	3	8
10	4	1	14	8	13	$\boxed{6}$	2	11	15	12	9	7	3	10	5	0
11	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

For example, $S_1(101010) = 6 = 0110$.

Data Encryption Standard(DES Algorithm)

DES Decryption:-

Decryption uses the same algorithm as encryption, except that the application of subkeys is reversed. Additionally initial and final permutations are reversed.

Data Encryption Standard(DES Algorithm)

The Avalanche Effect:

- A desirable property of any encryption algorithm is that a small change in either plaintext or the key should produce a significant change in the ciphertext.
- A change in one bit of the plaintext or one of the bit of the key should produce a change in many bits of the ciphertext. This is referred as Avalanche Effect.
- If the change were smaller, this might provide a way to reduce the size of the plaintext or key space to be searched.

For Ex:

1 bit change in PT - 34 bits change in CT on average.

Data Encryption Standard(DES Algorithm)

Plaintext

00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000

10000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000

with the key
0000001 1001011 0100100 1100010 0011100
0011000 0011100 0110010

Plaintext
01101000 10000101 00101111 01111010
00010011 01110110 11101011 10100100

with two keys that differ in only one bit position:

1110010 1111011 1101111 0011000 0011101
0000100 0110001 11011100
0110010 1111011 1101111 0011000
0011101 0000100 0110001 11011100

Table 3.5. Avalanche Effect in DES

(a) Change in Plaintext		(b) Change in Key	
Round	Number of bits that differ	Round	Number of bits that differ
0	1	0	0
1	6	1	2
2	21	2	14
3	35	3	28
4	39	4	32
5	34	5	30
6	32	6	32
7	31	7	35
8	29	8	34
9	42	9	40
10	44	10	38
11	32	11	31
12	30	12	33
13	30	13	28
14	26	14	26
15	29	15	34
16	34	16	35

Strength of Data Encryption Standard(DES Algorithm)

Use of 56 bits keys:-

With a key length of 56 bits, there are 2^{56} possible keys, which is approximately 7.2×10^{16}. Thus, on the face of it, a brute-force attack appears impractical. Assuming that, on average, half the key space has to be searched, a single machine performing one DES encryption per microsecond would take more than a thousand years (see Table 2.2) to break the cipher.

DES finally and definitively proved insecure in July 1998, when the Electronic Frontier Foundation (EFF) announced that it had broken a DES encryption using a special-purpose "DES cracker" machine that was built for less than $250,000. The attack took less than three days. The EFF has published a detailed description of the machine, enabling others to build their own cracker

Simply running through all possible keys won't result in cracking the DES encryption. Unless known plain text is given, the attacker must be able to differentiate the plain text from other data. Some degree of knowledge about the target plain text and some techniques for automatically distinguishing plain text from garble are required to supplement the brute-force approach. If brute force attack is the only means to crack the DES encryption algorithm, then using longer keys will obviously help us to counter such attacks. An algorithm is guaranteed unbreakable by brute force if a 128- bit key is used.

Strength of Data Encryption Standard(DES Algorithm)

The Nature of DES Algorithm

Another concern is the possibility that cryptanalysis is possible by exploiting the characteristics of the DES algorithm. The focus of concern has been on the eight substitution tables, or S-boxes, that are used in each iteration. Because the design criteria for these boxes, and indeed for the entire algorithm, were not made public, there is a suspicion that the boxes were constructed in such a way that cryptanalysis is possible for an opponent who knows the weaknesses in the S-boxes.

Timing Attack

In essence, a timing attack is one in which information about the key or the plaintext is obtained by observing how long it takes a given implementation to perform decryptions on various ciphertexts. A timing attack exploits the fact that an encryption or decryption algorithm often takes slightly different amounts of time on different inputs

Strength of Data Encryption Standard(DES Algorithm)

Use of 56 bits keys:-

Key Size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 Decryptions/s	Time Required at 10^{13} Decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} years
26 characters (permutation)	Monoalphabetic	$2! = 4 \times 10^{26}$	2×10^{26} ns = 6.3×10^{9} years	6.3×10^{6} years

Block Cipher Design Principles

The cryptographic strength of a Feistel cipher derives from 3 aspects of the design: 3 critical aspects of block cipher design are:

1. **Number of Rounds:**

The greater number of rounds, more difficult it is to perform cryptanalysis, even for a relatively weak F. In general criterion should be that the number of rounds is chosen so that known cryptanalytic efforts require greater than effort than a simple brute force key search attack.

The number of Rounds is regularly considered in design criteria, it just reflects the number of rounds to be suitable for an algorithm to make it more complex, in DES we have 16 rounds ensuring it to be more secure while in AES we have 10 rounds which makes it more secure.

Block Cipher Design Principles

Design Function-F

The core part of the Feistel Block cipher structure is the Round Function. The complexity of cryptanalysis can be derived from the Round function i.e. the increasing level of complexity for the round function would be greatly contributing to an increase in complexity. To increase the complexity of the round function, the avalanche effect is also included in the round function, as the change of a single bit in plain text would produce a mischievous output due to the presence of avalanche effect.

S-Box would be selected based on below criteria

- **Random:** Use some pseudorandom number generation or some table of random digits to generate the entries in the S-boxes. This may lead to boxes with undesirable characteristics for small sizes (e.g., 6 x 4) but should be acceptable for large S-boxes (e.g., 8 x 32).
- **Random with testing:** Choose S-box entries randomly, then test the results against various criteria, and throw away those that do not pass.
- **Human-made:** This is a more or less manual approach with only simple mathematics to support it. It is apparently the technique used in the DES design. This approach is difficult to carry through for large S-boxes.
- **Math-made:** Generate S-boxes according to mathematical principles. By using mathematical construction, S-boxes can be constructed that offer proven security against linear and differential cryptanalysis, together with good diffusion.

Block Cipher Design Principles

Design Function-F

The core part of the Feistel Block cipher structure is the Round Function. The complexity of cryptanalysis can be derived from the Round function i.e. the increasing level of complexity for the round function would be greatly contributing to an increase in complexity. To increase the complexity of the round function, the avalanche effect is also included in the round function, as the change of a single bit in plain text would produce a mischievous output due to the presence of avalanche effect.

Key Schedule

The key schedule should be designed carefully to ensure that the keys used for encryption are independent and unpredictable. The key schedule should also resist attacks that exploit weak keys or key-dependent properties of the cipher.

Evaluation Criterion for AES:-

Origin of AES

- In 1999, NIST(National Institute of Standards and Technology) issued new version of DES standard-3DES.
- 3DES has 2 attractions:

✔ 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.

Evaluation Criterion for AES:-

Origin of AES

3DES has following drawbacks:

- ✔ 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 that both DES and 3DES use a 64-bit block size. For reasons of both efficiency and security, a larger block size is desirable.

- 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.

Evaluation Criterion for AES:-

AES Evaluation:

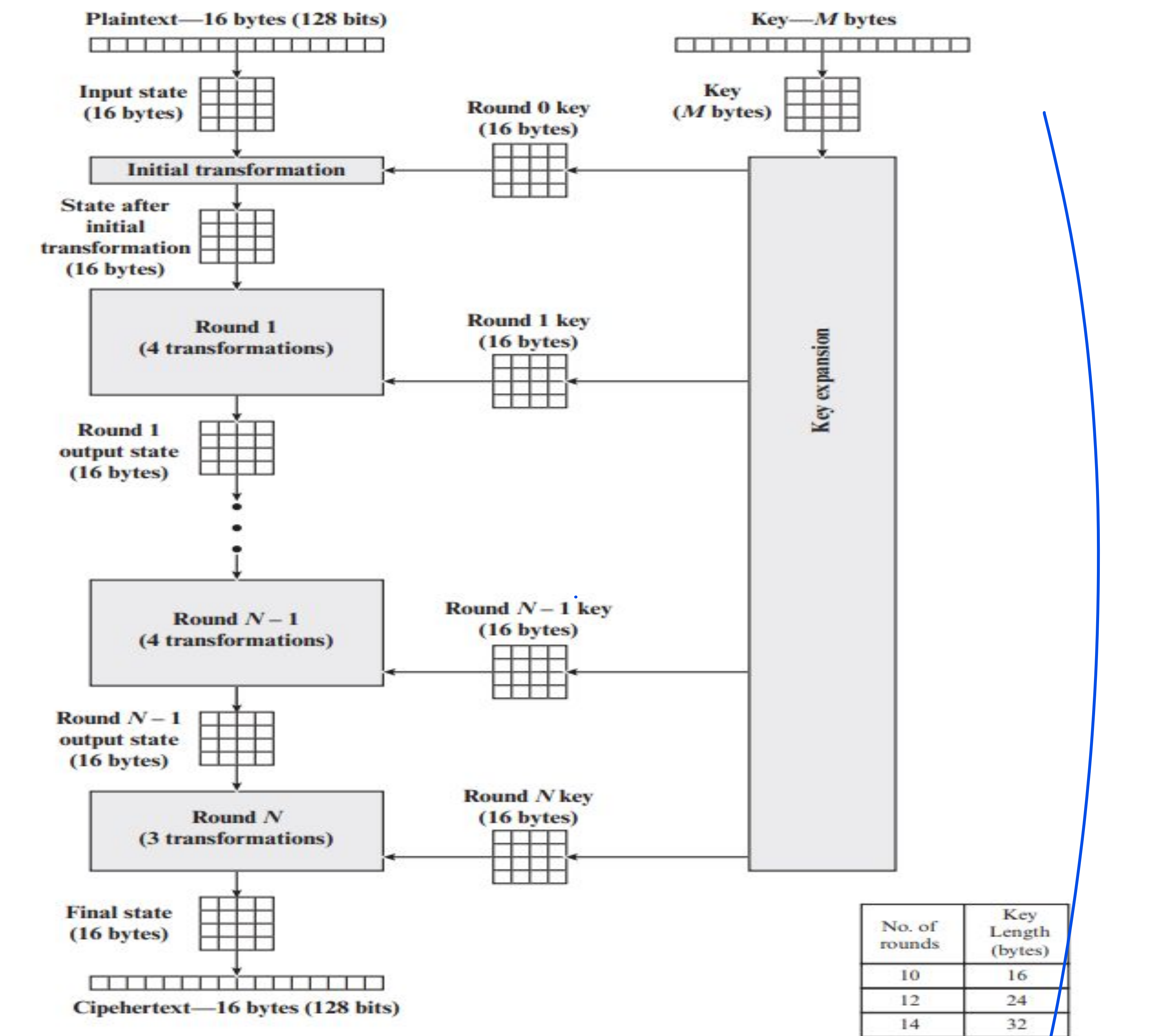
The three categories of criteria were as follows:

- **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.

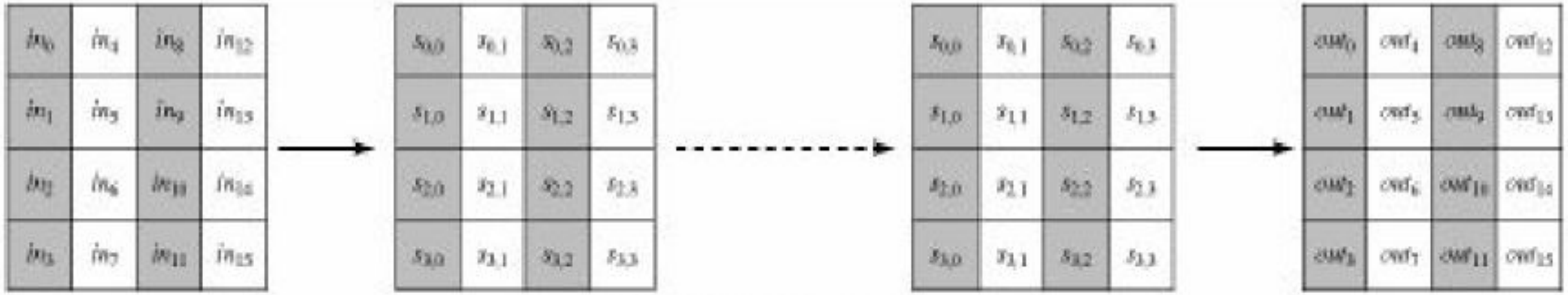
- **Cost:** NIST intends AES to be practical in a wide range of applications. Accordingly, AES must have high computational efficiency, so as to be usable in high-speed applications, such as broadband links.

- **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.

AES Structure



AES Cipher:-



(a) Input, state array, and output

(b) Key and expanded key

AES Cipher:-

The Rijndael proposal for AES defined a cipher in which the block length and the key length can be independently specified to be 128, 192, or 256 bits. The AES specification uses the same three key size alternatives but limits the block length to 128 bits. A number of AES parameters depend on the key length.

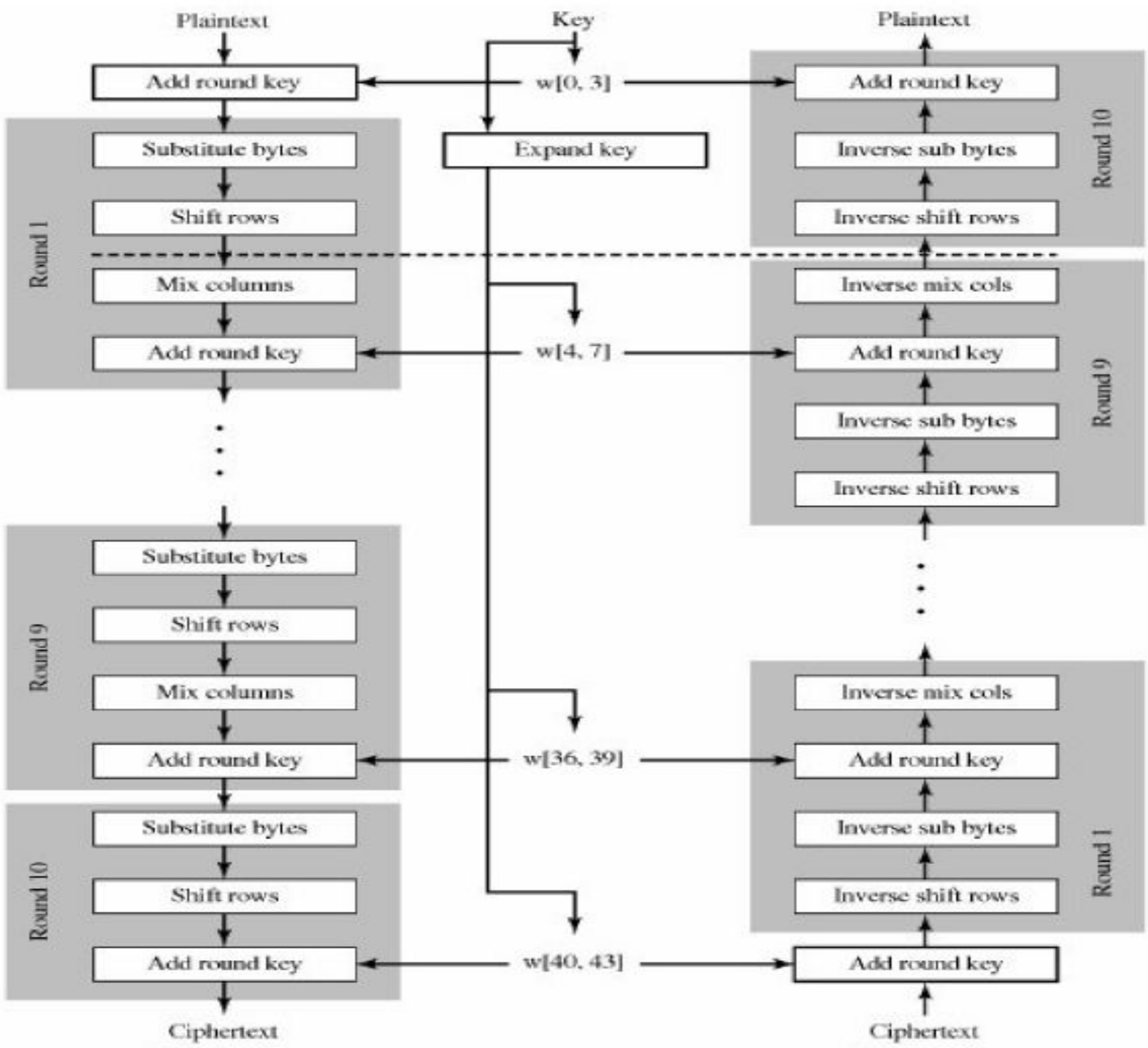
Table 1. AES Parameters

Key Size (words/bytes/bits)	4/16/128	6/24/192	8/32/256
Plaintext Block Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Number of Rounds	10	12	14
Round Key Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Expanded Key Size (words/bytes)	44/176	52/208	60/240

Rijndael was designed to have the following characteristics:

- Resistance against all known attacks
- Speed and code compactness on a wide range of platforms
- Design simplicity

AES Cipher:-



(a) Encryption

(b) Decryption

AES Substitute bytes:-

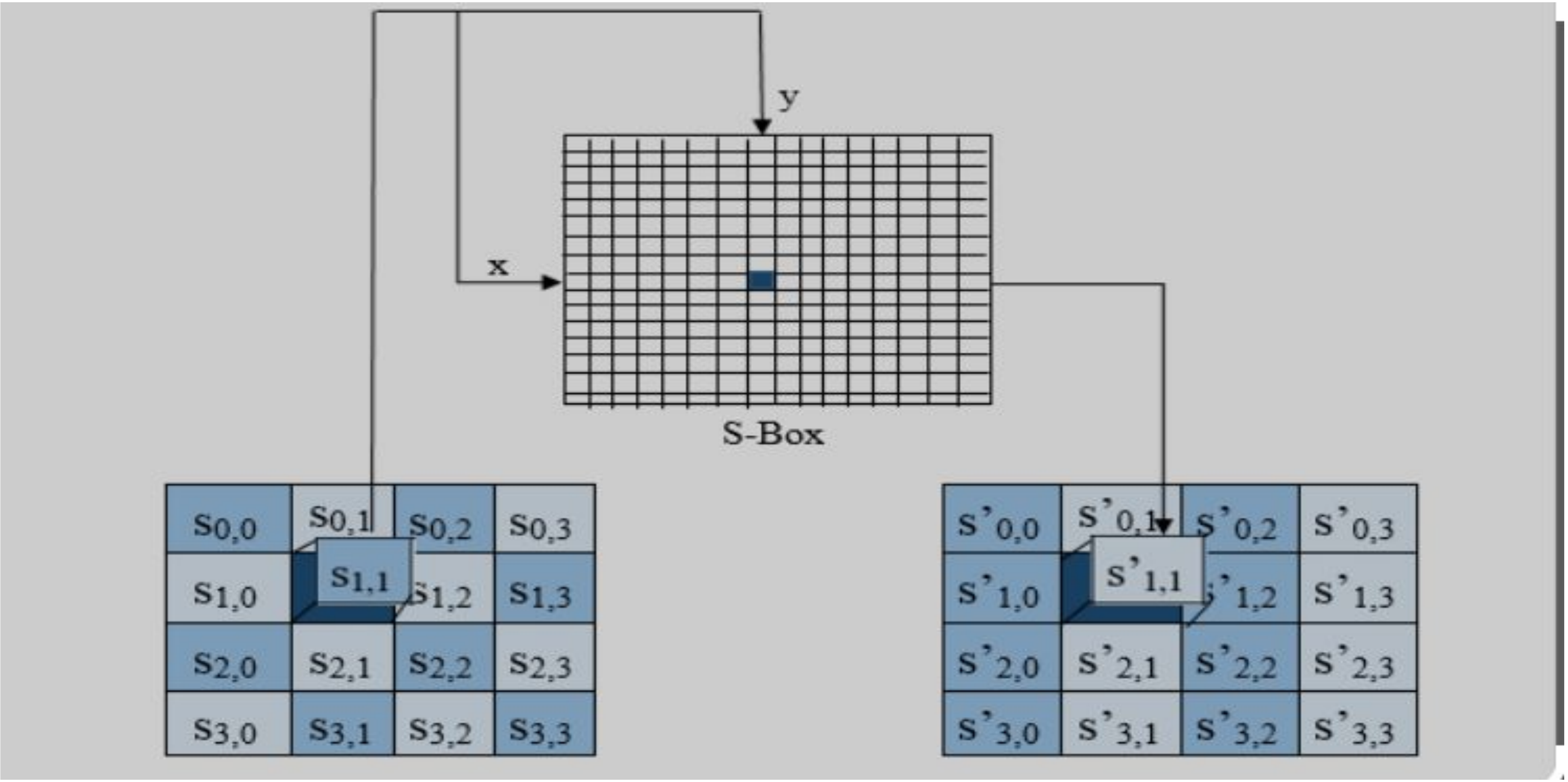
- First transformation function of AES is called substitute bytes(sub bytes).
- It is also known as forward substitute bytes transformation.
- AES defines 16 X16 matrix of byte values ,called an S-box that contains permutation of all possible 256 8-bit values.
- These row and column values serve as indexes into the S-box to select a unique 8 bit output value.

1. SubBytes / Substitute Bytes

- AES defines a 16 x 16 matrix of byte values, called an S-box, that contains a permutation of all possible 256 8-bit values.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

AES Substitute Bytes



AES Substitute Bytes Example:

State before substitution operation

34	35	89	10
A3	D5	6B	8F
89	78	67	56
45	63	6D	13

State after substitution operation

18	96	A7	CA
0A	03	7F	73
A7	BC	85	B1
6E	FB	3C	7D

S-Box

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF

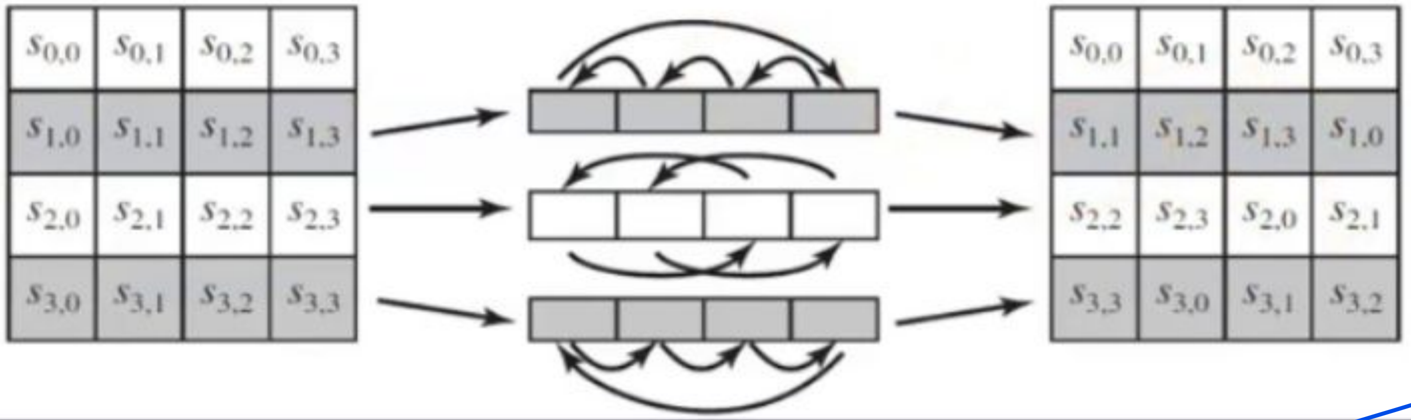
AES Shift Rows:

Shift Row transformation

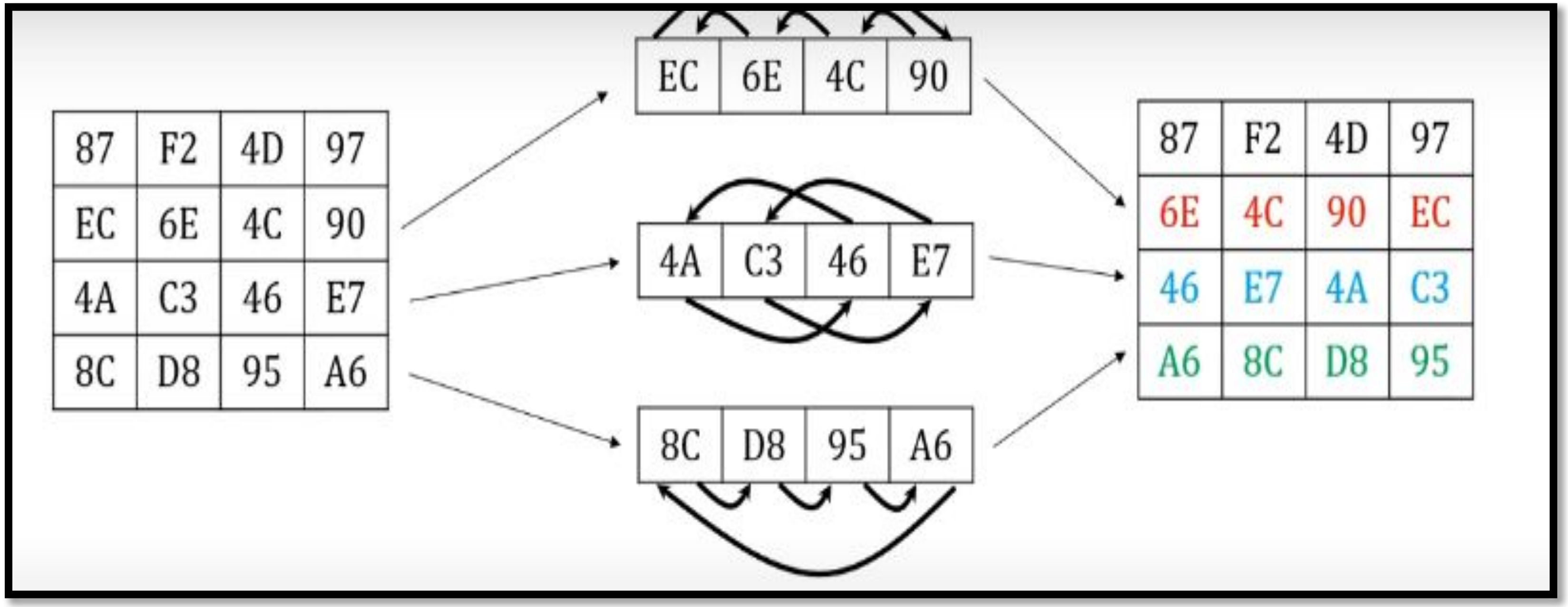
The shift row transformation is called ShiftRows.

Rules of shifting rows,

- Row 1 → No Shifting
- Row 2 → 1 byte left shift
- Row 3 → 2 byte left shift
- Row 4 → 3 byte left shift



AES Shift Rows:



- Row 1 → No Shifting
- Row 2 → 1 byte circular left shift
- Row 3 → 2 byte circular left shift
- Row 4 → 3 byte circular left shift

AES Mix Columns:

Mix Columns operates on each column individually.

$$
\underbrace{\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}}_{\text{Predefine Matrix}}
*
\underbrace{\begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix}}_{\text{State Array}}
=
\underbrace{\begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{bmatrix}}_{\text{New State Array}}
$$

State Array = Output of Shift Rows function.

AES Mix Columns:

Mix Columns operates on each column individually.

$$S'_{0,j} = (2 * S_{0,j}) \oplus (3 * S_{1,j}) \oplus S_{2,j} \oplus S_{3,j}$$

$$S'_{1,j} = S_{0,j} \oplus (2 * S_{1,j}) \oplus (3 * S_{2,j}) \oplus S_{3,j}$$

$$S'_{2,j} = S_{0,j} \oplus S_{1,j} \oplus (2 * S_{2,j}) \oplus (3 * S_{3,j})$$

$$S'_{3,j} = (3 * S_{0,j}) \oplus S_{1,j} \oplus S_{2,j} \oplus (2 * S_{3,j})$$

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} * \begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix} = \begin{bmatrix} ? & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}$$

$$\{02\} * \{87\} \oplus \{03\} * \{6E\} \oplus \{01\} * \{46\} \oplus \{01\} * \{A6\}$$

AES Mix Columns:

$$
\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}
*
\begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix}
=
\begin{bmatrix} ? & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}
$$

$$\{02\} * \{87\} \oplus \{03\} * \{6E\} \oplus \{01\} * \{46\} \oplus \{01\} * \{A6\}$$

$$
\begin{aligned}
02 &= 0000\ 0010 = X \\
87 &= 1000\ 0111 = X^7 + X^2 + X + 1 \\
02 * 87 &= X * (X^7 + X^2 + X + 1) \\
&= X^8 + X^3 + X^2 + X \\
&= X^4 + \cancel{X^3} + \cancel{X} + 1 + \cancel{X^3} + X^2 + \cancel{X} \\
&= X^4 + X^2 + 1 \\
&= 0001\ 0101
\end{aligned}
$$

$$
\begin{aligned}
03 &= 0000\ 0011 = X + 1 \\
6E &= 0110\ 1110 = X^6 + X^5 + X^3 + X^2 + X \\
03 * 6E &= (X+1) * (X^6 + X^5 + X^3 + X^2 + X) \\
&= X^7 + \cancel{X^6} + X^4 + \cancel{X^3} + \cancel{X^2} + \cancel{X^6} + X^5 + \cancel{X^3} + \cancel{X^2} + X \\
&= X^7 + X^5 + X^4 + X
\end{aligned}
$$

AES Mix Columns:

AES Mix Column

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} * \begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix} = \begin{bmatrix} 47 & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}$$

$$\{02\} * \{87\} \oplus \{03\} * \{6E\} \oplus \{01\} * \{46\} \oplus \{01\} * \{A6\} = \{47\}$$

02 * 87 = 0 0 0 1 0 1 0 1
03 * 6E = 1 0 1 1 0 0 1 0
01 * 46 = 0 1 0 0 0 1 1 0
01 * A6 = 1 0 1 0 0 1 1 0

= 0 1 0 0 0 1 1 1 = {47}

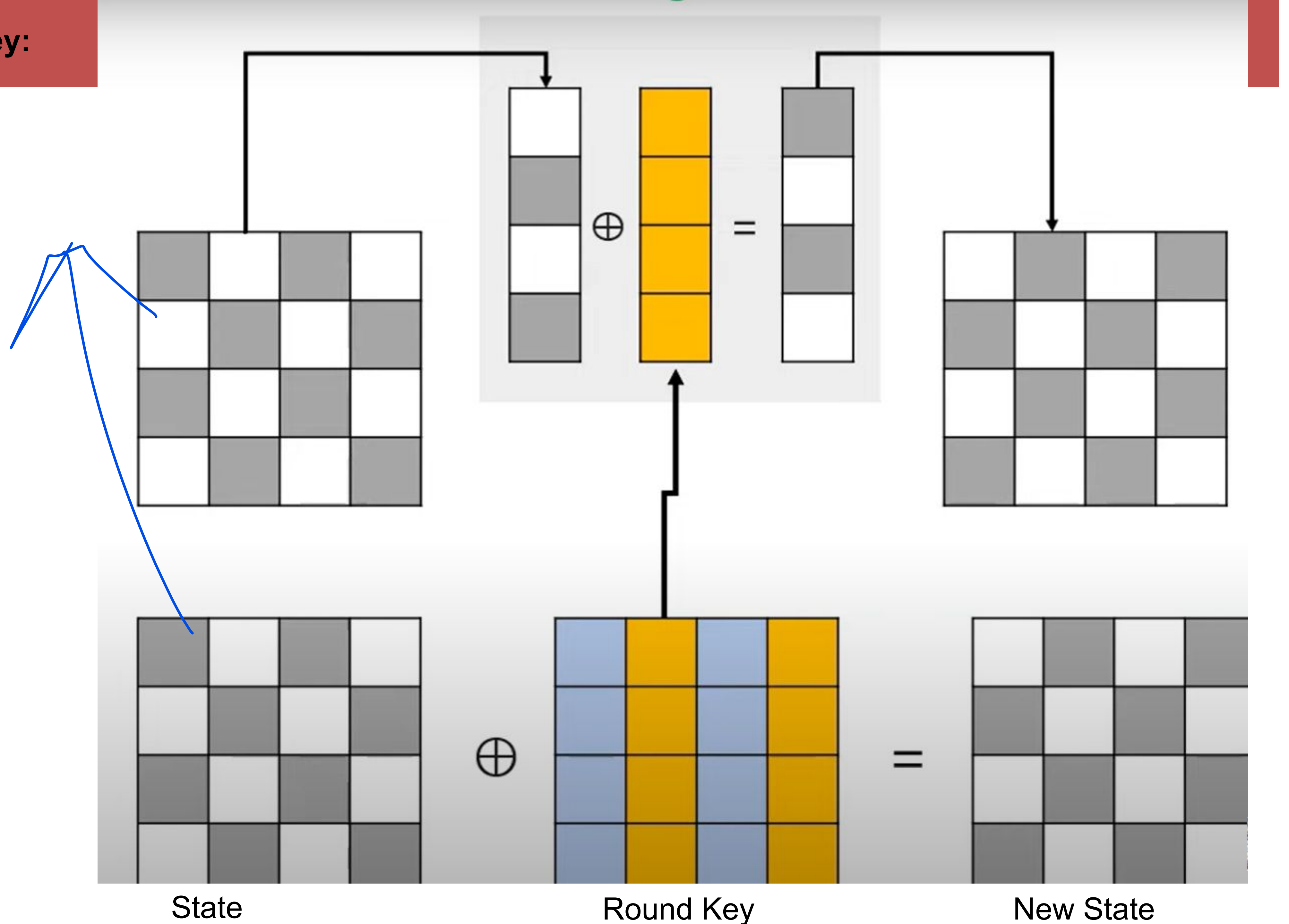
$\underbrace{0100}_{4}\ \underbrace{0111}_{7}$

In Ex-or
Odd tim
Even tin

AES Mix Columns:

$$
\begin{array}{|c|c|c|c|}
\hline
2 & 3 & 1 & 1 \\ \hline
1 & 2 & 3 & 1 \\ \hline
1 & 1 & 2 & 3 \\ \hline
3 & 1 & 1 & 2 \\ \hline
\end{array}
\;*\;
\begin{array}{|c|c|c|c|}
\hline
87 & F2 & 4D & 97 \\ \hline
6E & 4C & 90 & EC \\ \hline
46 & E7 & 4A & C3 \\ \hline
A6 & 8C & D8 & 95 \\ \hline
\end{array}
\;=\;
\begin{array}{|c|c|c|c|}
\hline
47 & 40 & A3 & 4C \\ \hline
37 & D4 & 70 & 9F \\ \hline
94 & E4 & 3A & 42 \\ \hline
ED & A5 & A6 & BC \\ \hline
\end{array}
$$

AES Add Round Key:



47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

$\oplus$

AC	19	28	57
77	FA	D1	5C
66	DC	29	00
F3	21	41	6A

=

EB			
40			
?			

$47 \oplus AC = ?$

$47 = 0100\ 0111$

$AC = 1010\ 1100$

$= \underbrace{1110}_{E}\ \underbrace{1011}_{B} = \{EB\}$

$37 \oplus 77 = ?$

$37 = 0011\ 0111$

$77 = 0111\ 0111$

$= \underbrace{0100}_{4}\ \underbrace{0000}_{0} = \{40\}$

AES Add Round Key

$$\begin{bmatrix} 47 & 40 & A3 & 4C \\ 37 & D4 & 70 & 9F \\ 94 & E4 & 3A & 42 \\ ED & A5 & A6 & BC \end{bmatrix} \oplus \begin{bmatrix} AC & 19 & 28 & 57 \\ 77 & FA & D1 & 5C \\ 66 & DC & 29 & 00 \\ F3 & 21 & 41 & 6A \end{bmatrix} = \begin{bmatrix} EB & 59 & 8B & 1B \\ 40 & 2E & A1 & C3 \\ F2 & 38 & 13 & 42 \\ 1E & 84 & E7 & D6 \end{bmatrix}$$

AES Diagram of each round:-

Overall AES Structure

1. AES is not a Feistel structure. AES do not use a Feistel structure but process the entire data block in parallel during each round using substitutions and permutation.

2. The key that is provided as input is expanded into an array of forty-four 32-bit words, **w**[*i*]. Four distinct words (128 bits) serve as
a round key for each round;

3. Four different stages are used, one of permutation and three of substitution:

Substitute bytes: Uses an S-box to perform a byte-by-byte substitution of the block
ShiftRows: A simple permutation
MixColumns: A substitution that makes use of arithmetic over $GF(2^8)$
AddRoundKey: A simple bitwise XOR of the current block with a portion of the expanded key

4. The structure is quite simple. For both encryption and decryption, the cipher begins with an AddRoundKey stage, followed by nine rounds that each includes all four stages, followed by a tenth round of three stages.

Overall AES Structure

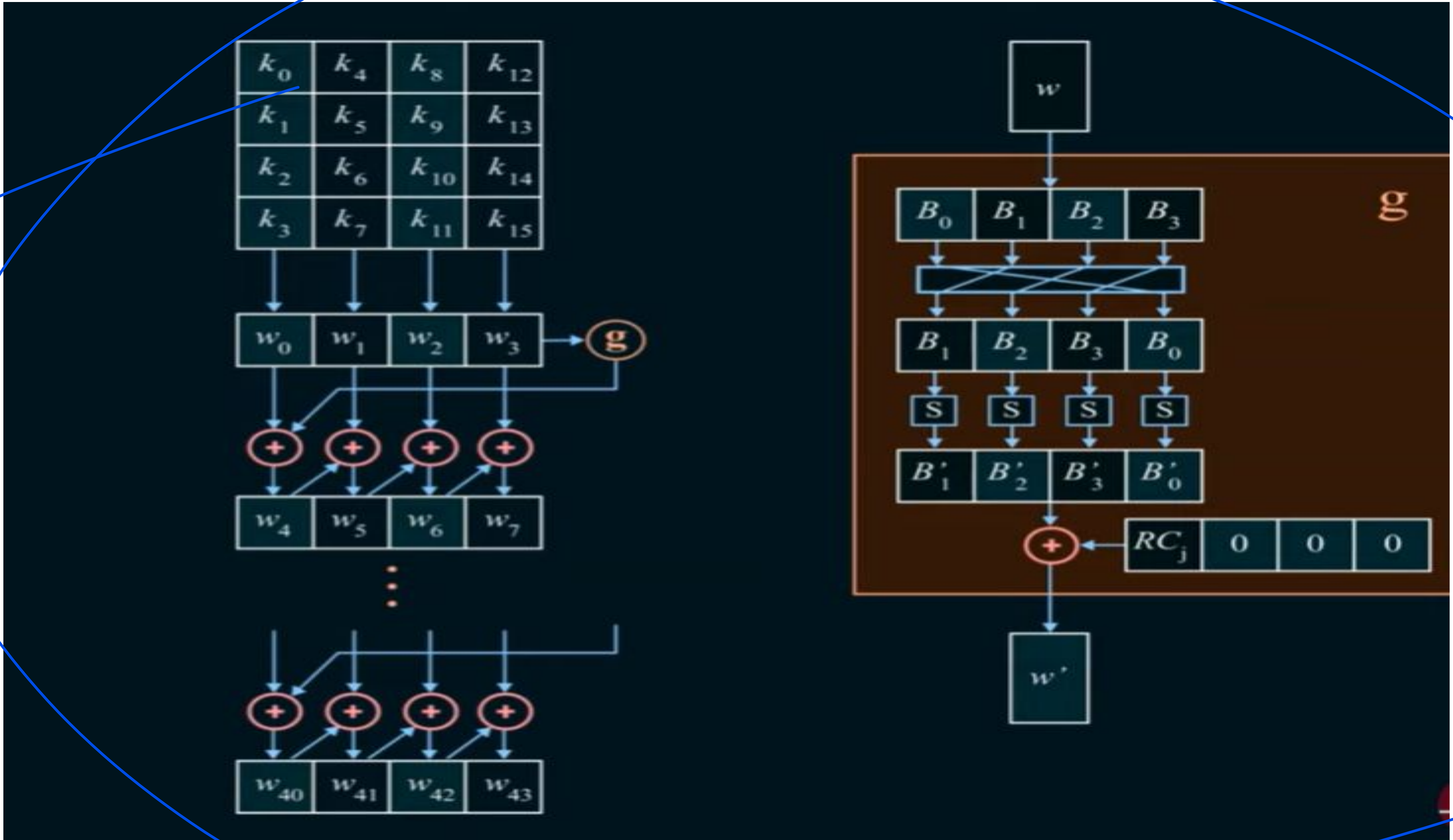
5. Only the AddRoundKey stage makes use of the key. For this reason, the cipher begins and ends with an AddRoundKey stage. Any other stage, applied at the beginning or end, is reversible without knowledge of the key and so would add no security.

6. Each stage is easily reversible. For the Substitute Byte, ShiftRows, and MixColumns stages, an inverse function is used in the decryption algorithm. For the AddRoundKey stage, the inverse is achieved by XORing the same round key to the block, using the result that $A \oplus A \oplus B = B$.

7. As with most block ciphers, the decryption algorithm makes use of the expanded key in reverse order. However, the decryption algorithm is not identical to the encryption algorithm. This is a consequence of the particular structure of AES.

8. The final round of both encryption and decryption consists of only three stages. Again, this is a consequence of the particular structure of AES and is required to make the cipher reversible.

Key Expansion



Key Expansion

Round Constant in AES

Round	RC	Round	RC
1	$(01\ 00\ 00\ 00)_{16}$	6	$(20\ 00\ 00\ 00)_{16}$
2	$(02\ 00\ 00\ 00)_{16}$	7	$(40\ 00\ 00\ 00)_{16}$
3	$(04\ 00\ 00\ 00)_{16}$	8	$(80\ 00\ 00\ 00)_{16}$
4	$(08\ 00\ 00\ 00)_{16}$	9	$(1B\ 00\ 00\ 00)_{16}$
5	$(10\ 00\ 00\ 00)_{16}$	10	$(36\ 00\ 00\ 00)_{16}$
Note: Initial Transformation takes $(00\ 00\ 00\ 00)_{16}$ as the RC.			

Key Expansion

- Key Expansion in AES-128 process

1. Cipher key is an array of 16bytes(k0 to k15) The first 4 words(w0,w1,w2,w3) are made from cipher key

   K0 to k3 -> w0

   k4 to k7 -> w1

   K8 to k11 ->w2

   K12 to k15 ->w3

2. The rest of the words (wi for i =4 to 43) are made as follows

   i) if (i mod 4) ≠ 0, wi = wi-1 ⊕wi-4

   ii) if (i mod 4) = 0, wi = t⊕wi-4

   Temporary word t

   ti = subword(Rotword(wi-1)) ⊕ Rcon i/4

KeyExpansion ([key_0 to key_{15}], [$\mathbf{w}_0$ to $\mathbf{w}_{43}$])

```
{
    for (i = 0 to 3)
        w_i ← key_4i + key_4i+1 + key_4i+2 + key_4i+3

    for (i = 4 to 43)
    {
        if (i mod 4 ≠ 0)    w_i ← w_i-1 + w_i-4
        else
        {
            t ← SubWord (RotWord (w_i-1)) ⊕ RCon_i/4        // t is a temporary word
            w_i ← t + w_i-4
        }
    }
}
```

Inverse Sub Bytes

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

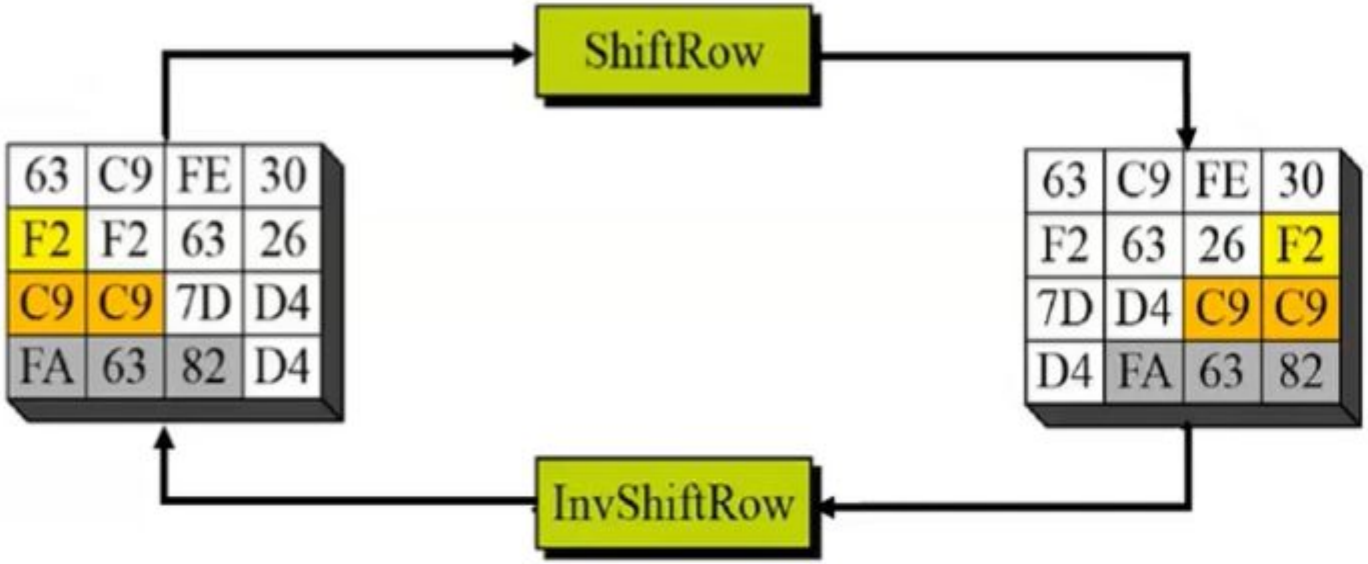
F2 $\longrightarrow$ 04

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

EA	?		

Inverse Shift Rows

The inverse shift row transformation, called InvShiftRows, performs the circular shifts in the opposite direction for each of the last three rows, with a one-byte circular right shift for the second row, and so on.



Inverse Mix Columns

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} * \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} = \begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{bmatrix}$$

Predefine Matrix For decryption — **State Array** — **New State Array**

State Array = Output of previous transformation function in decryption process

$$S'_{0,j} = (0E * S_{0,j}) \oplus (0B * S_{1,j}) \oplus (0D * S_{2,j}) \oplus (09 * S_{3,j})$$

$$S'_{1,j} = (09 * S_{0,j}) \oplus (0E * S_{1,j}) \oplus (0B * S_{2,j}) \oplus (0D * S_{3,j})$$

$$S'_{2,j} = (0D * S_{0,j}) \oplus (09 * S_{1,j}) \oplus (0E * S_{2,j}) \oplus (0B * S_{3,j})$$

$$S'_{3,j} = (0B * S_{0,j}) \oplus (0D * S_{1,j}) \oplus (09 * S_{2,j}) \oplus (0E * S_{3,j})$$

Inverse Mix Columns

0E	0B	0D	09
09	0E	0B	0D
0D	09	0E	0B
0B	0D	09	0E

*

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

=

?			

$\boxed{\{0E\} * \{47\}} \oplus \{0B\} * \{37\} \oplus \{0D\} * \{94\} \oplus \{09\} * \{ED\}$

$$\begin{array}{cccccccc} X^7 + & X^6 + & X^5 + & X^4 + & \boxed{X^3} + & \boxed{X^2} + & \boxed{X^1} + & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{array}$$

GF(2^3)

0E = 0000 1110 = $X^3 + X^2 + X$

47 = 0100 0111

Inverse Mix Columns

0E	0B	0D	09
09	0E	0B	0D
0D	09	0E	0B
0B	0D	09	0E

*

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

=

?			

$$\boxed{\{0E\} * \{47\}} \oplus \{0B\} * \{37\} \oplus \{0D\} * \{94\} \oplus \{09\} * \{ED\}$$

$$\boxed{X^9 = X^{8+1} = X^8 * X}$$

$$\begin{aligned}
\{0E\} * \{47\} &= (X^3 + X^2 + X) * (X^6 + X^2 + X + 1) \\
&= \boxed{X^9} + X^5 + \cancel{X^4} + \cancel{X^3} + \boxed{X^8} + \cancel{X^4} + \cancel{X^3} + \cancel{X^2} + X^7 + X^3 + \cancel{X^2} + X \\
&= (\boxed{X^8} * X) + X^5 + \boxed{X^8} + X^7 + X^3 + X \\
&= ((X^4 + X^3 + X + 1) * X) + X^5 + X^4 + \cancel{X^3} + \cancel{X} + 1 + X^7 + \cancel{X^3} + \cancel{X} \\
&= \cancel{X^5} + \cancel{X^4} + X^2 + X + \cancel{X^5} + \cancel{X^4} + 1 + X^7 \\
&= X^7 + X^2 + X + 1 \\
&= 1000\ 0111
\end{aligned}$$

Use irreducible Polynomial Theorem, GF (2^3)

$X^8 = X^4 + X^3 + X + 1$

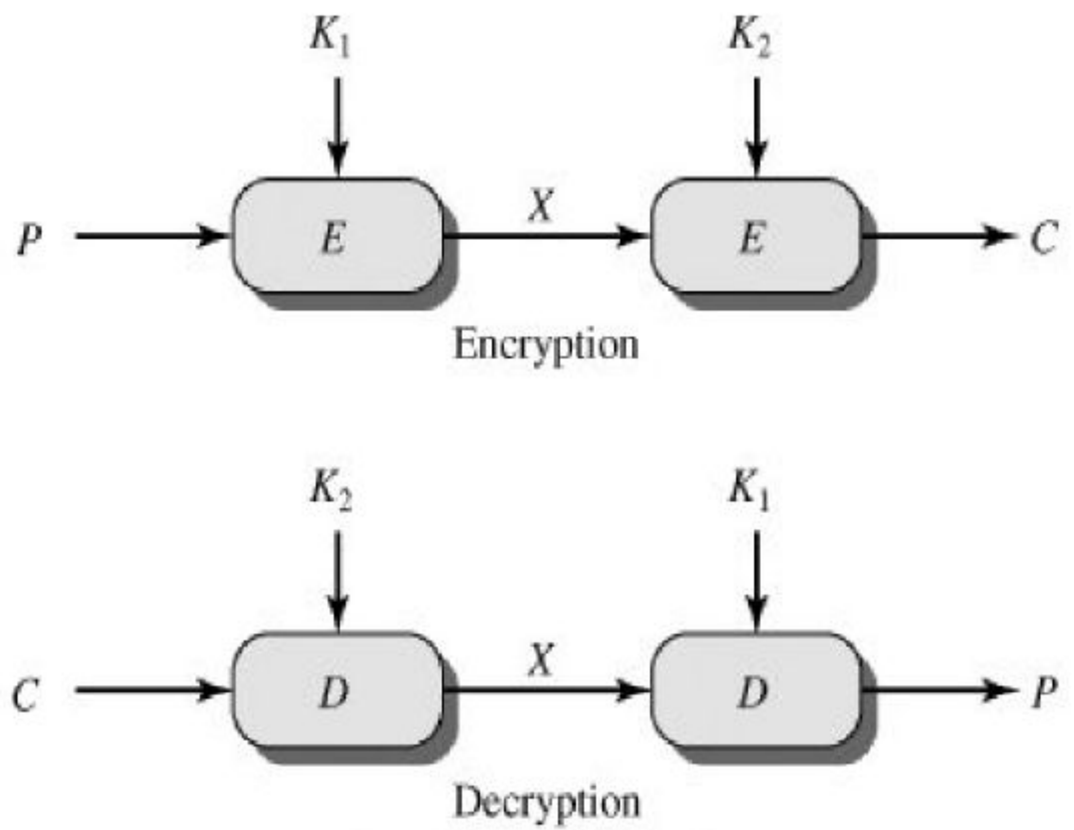
X^7	X^6	X^5	X^4	X^3	X^2	X^1	1
1	0	0	0	0	1	1	1

Difference between AES and DES

AES	DES
AES stands for Advanced Encryption Standard	DES stands for Data Encryption Standard
The date of creation is 2001	The date of creation is 1977
Key length can be 128-bits, 192-bits, and 256-bits.	The key length is 56 bits in DES.
Number of rounds depends on key length: 10(128-bits), 12(192-bits), or 14(256-bits)	DES involves 16 rounds of identical operations
The structure is based on a substitution-permutation network.	The structure is based on a Feistel network.
AES is more secure than the DES cipher and is the de facto world standard.	DES can be broken easily as it has known vulnerabilities. 3DES(Triple DES) is a variation of DES which is secure than the usual DES.
The rounds in AES are: Byte Substitution, Shift Row, Mix Column and Key Addition	The rounds in DES are: Expansion, XOR operation with round key, Substitution and Permutation
AES can encrypt 128 bits of plaintext.	DES can encrypt 64 bits of plaintext.
It can generate Ciphertext of 128, 192, 256 bits.	It generates Ciphertext of 64 bits.
It is faster than DES.	It is slower than AES.
It is efficient with both hardware and software.	It is efficient only with software.

Double DES

Figure 6.1. Multiple Encryption



(a) Double encryption

- Double DES is a encryption technique which uses two instance of DES on same plain text.
- In both instances it uses different keys to encrypt the plain text. Both keys are required at the time of decryption.
- The 64 bit plain text goes into first DES instance which then converted into a 64 bit middle text using the first key and then it goes to second DES instance which gives 64 bit cipher text by using second key.

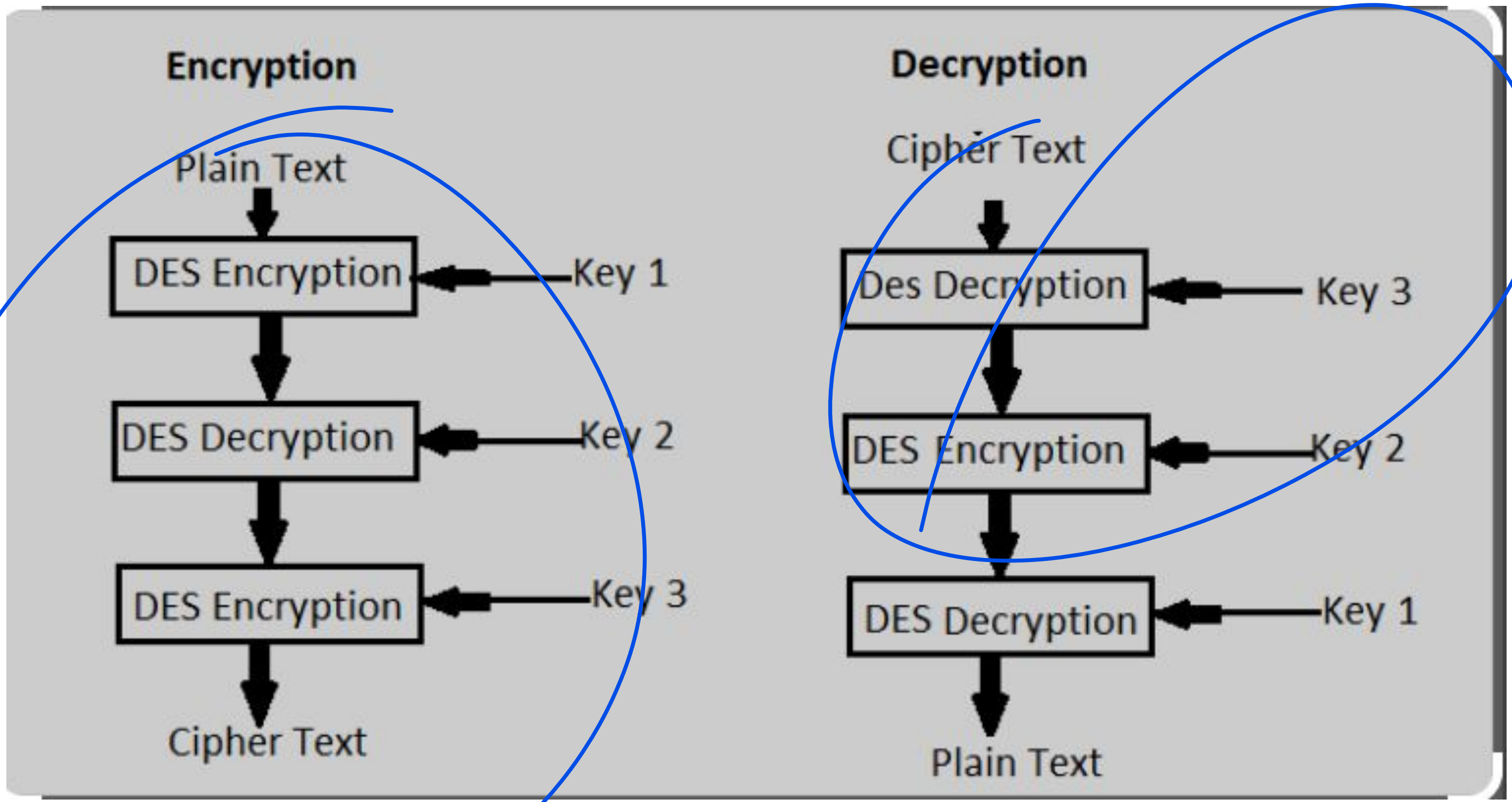
Given a plaintext P, K1 and k2 are keys

$$C = E(K_2, E(K_1, P))$$

Decryption requires that the keys be applied in reverse order:

$$P = D(K_1, D(K_2, C))$$

Triple DES



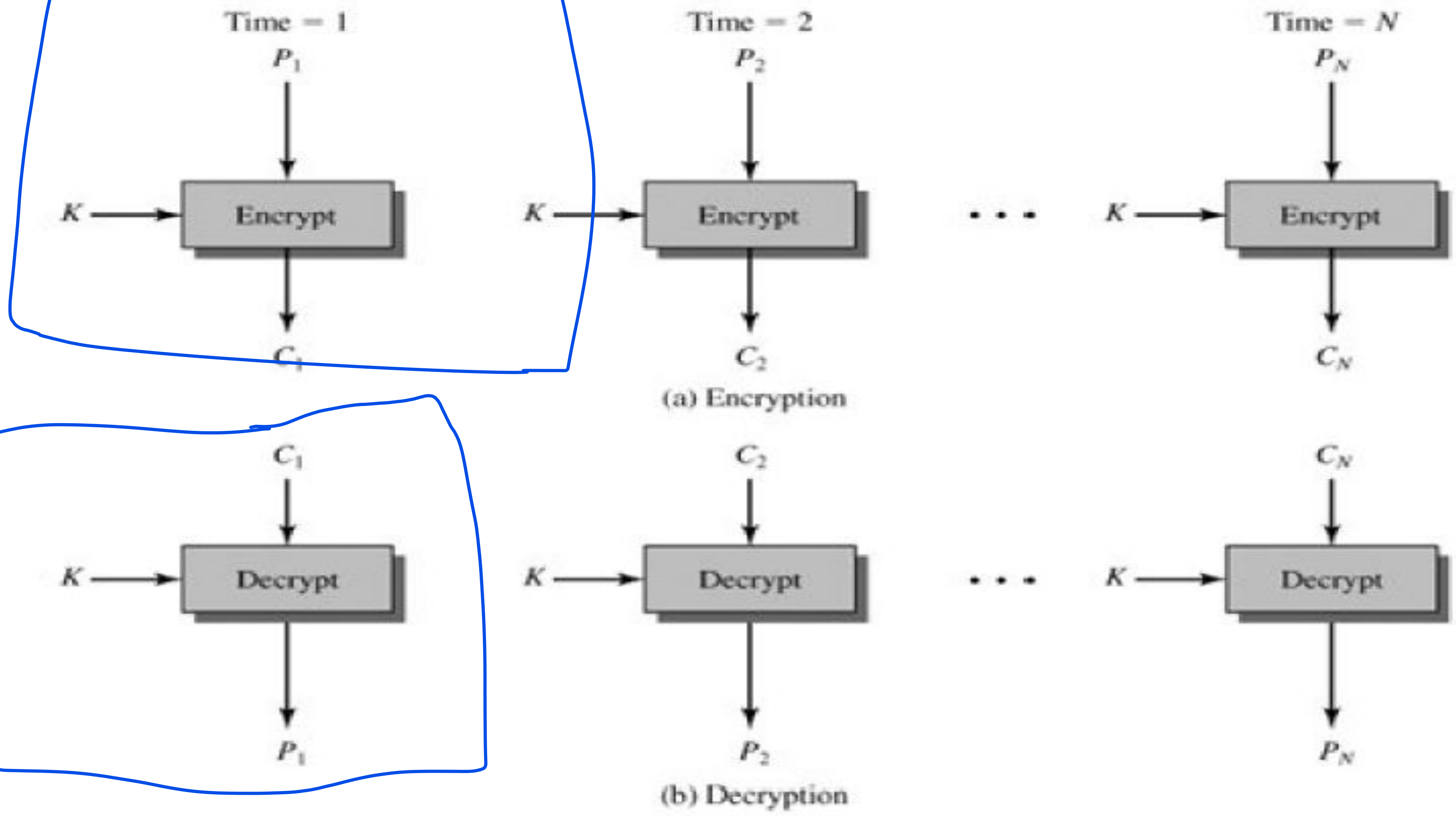
Ciphertext= E(k3, D(k2,(E(K1, Plaintext)))

Plaintext=D(k1,E(k2,D(k3,Ciphertext)))

Block Cipher Modes of Operation

1. ECB mode:- (Electronic Code Block) mode

[View full size image]



Block Cipher Modes of Operation

1. ECB mode:- (Electronic Code Block) mode

Advantages

- Parallel Encryption of Blocks
- Faster Encryption

Disadvantages

- The most significant characteristic of ECB is that the same b-bit block of plaintext, if it appears to be more than once in the message, always produces same ciphertext.
- For lengthy messages, the ECB mode may not be secure. If the message is highly structured, it may be possible for a cryptanalyst to exploit these regularities.

Application

Secure transmission of single values

Block Cipher Modes of Operation

CBC mode:- (Cipher Block Chaining Mode)

Advantages

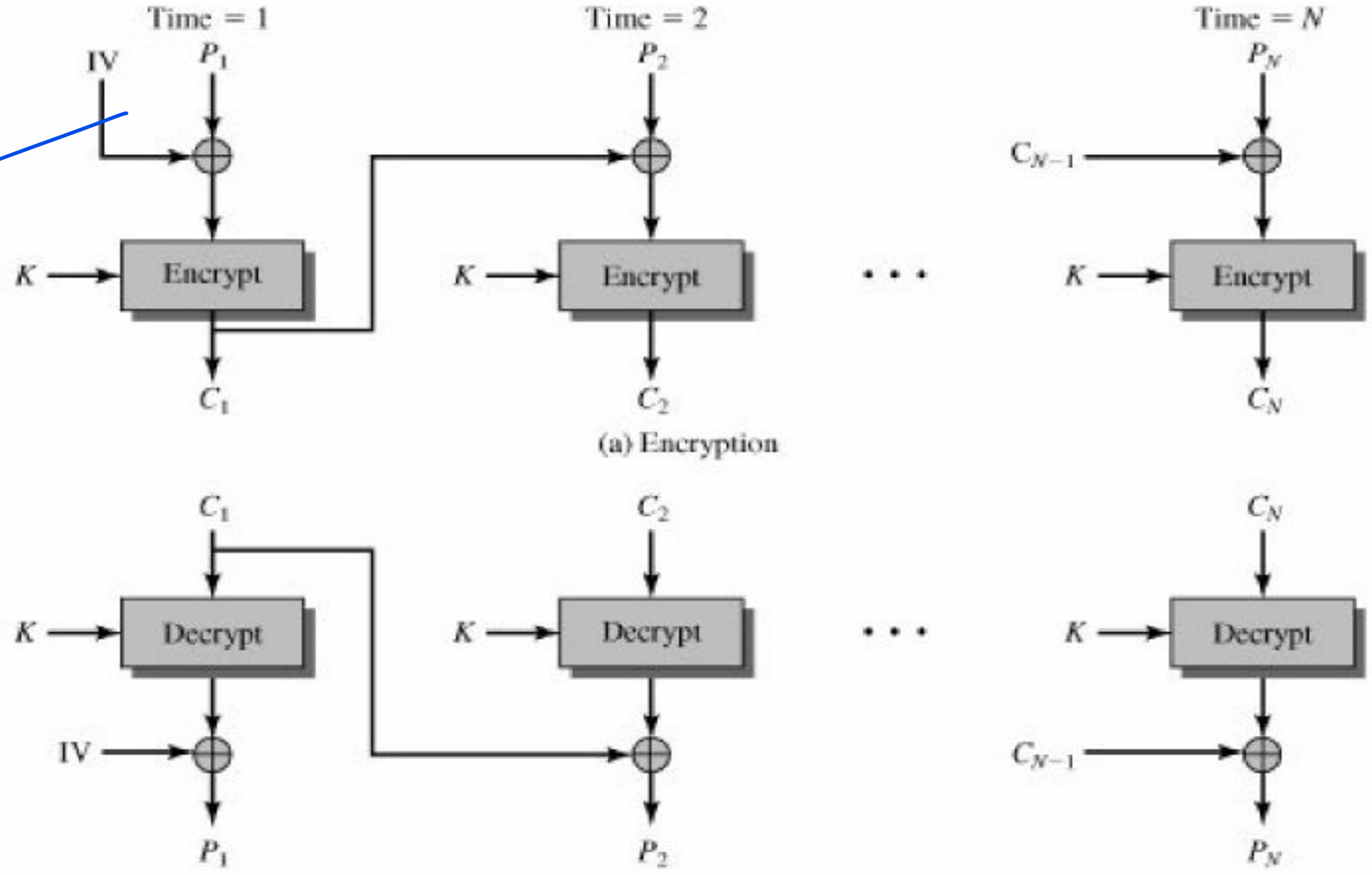
- CBC is used to overcome the security deficiencies of ECB..Same plaintext block produces different ciphertext.
- The IV should be protected against unauthorized changes for maximum security.

Disadvantage

- Parallel Encryption is not possible

Figure 6.4. Cipher Block Chaining (CBC) Mode

View full size image



(a) Encryption

(b) Decryption

Block Cipher Modes of Operation

CTR mode:- (Counter Mode)

Advantage

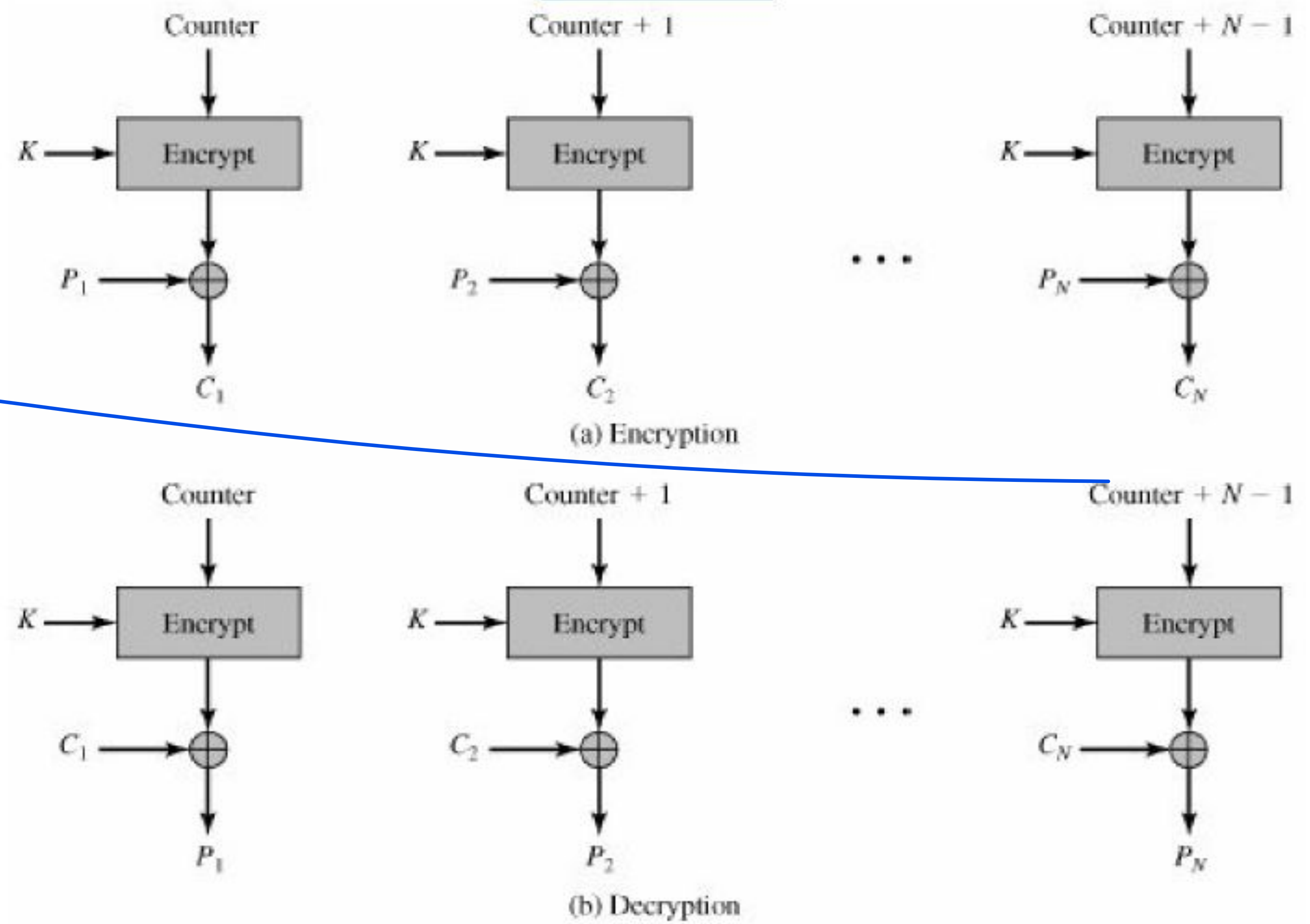
- The disadvantages of both ECB and CBC are overcome by using CTR.
- Efficiency
- Preprocessing

.

.

Figure 6.7. Counter (CTR) Mode

[View full size image]



Block Cipher Modes of Operation

CBC mode:- (Cipher Block Chaining Mode)

$C_1 = E(K, [IV \oplus P_1])$

$P_1 = IV \oplus D(K, C_1)$

Now use the notation that X[i] denotes the ith bit of the b-bit quantity X. Then

$P_1[i] = IV[i] \oplus D(K, C_1)[i]$

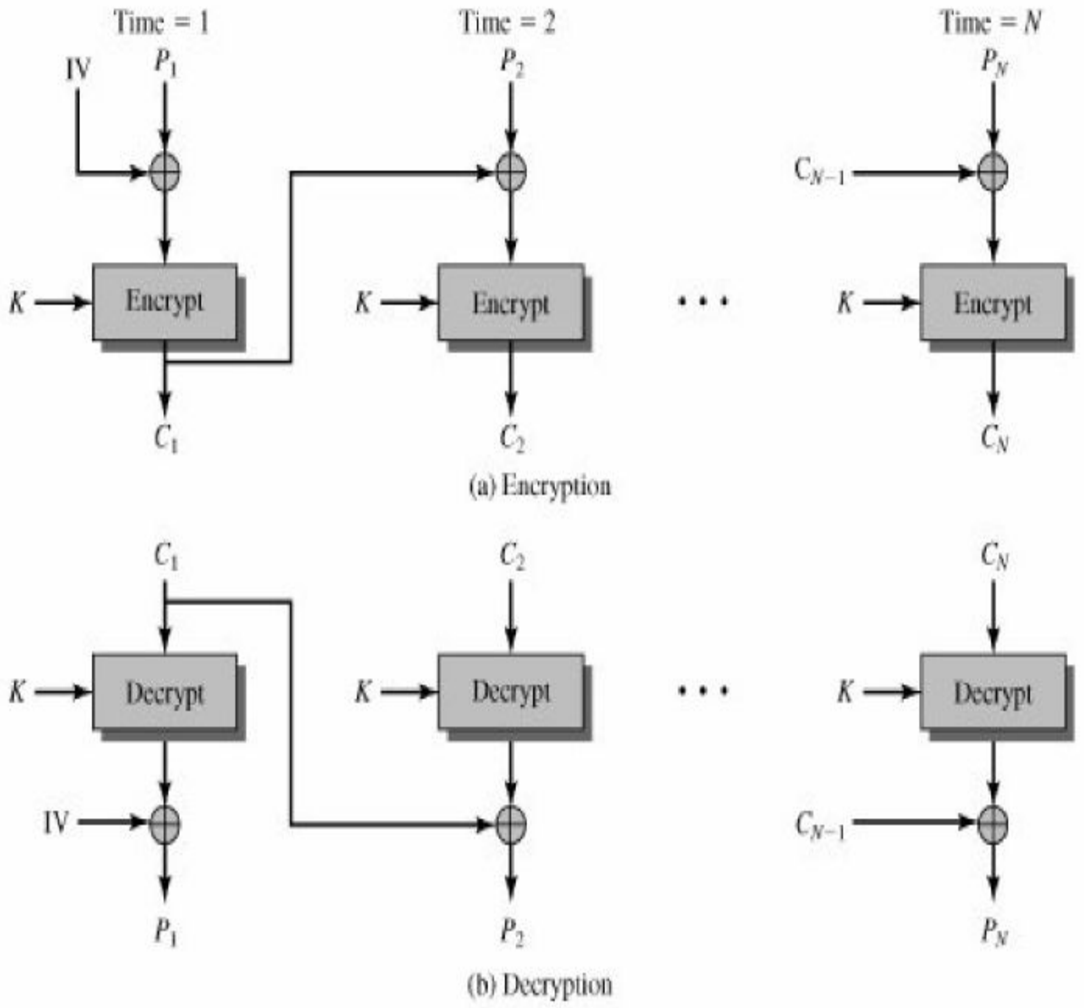
Then, using the properties of XOR, we can state

$P_1[i]' = IV[i]' \oplus D(K, C_1)[i]$

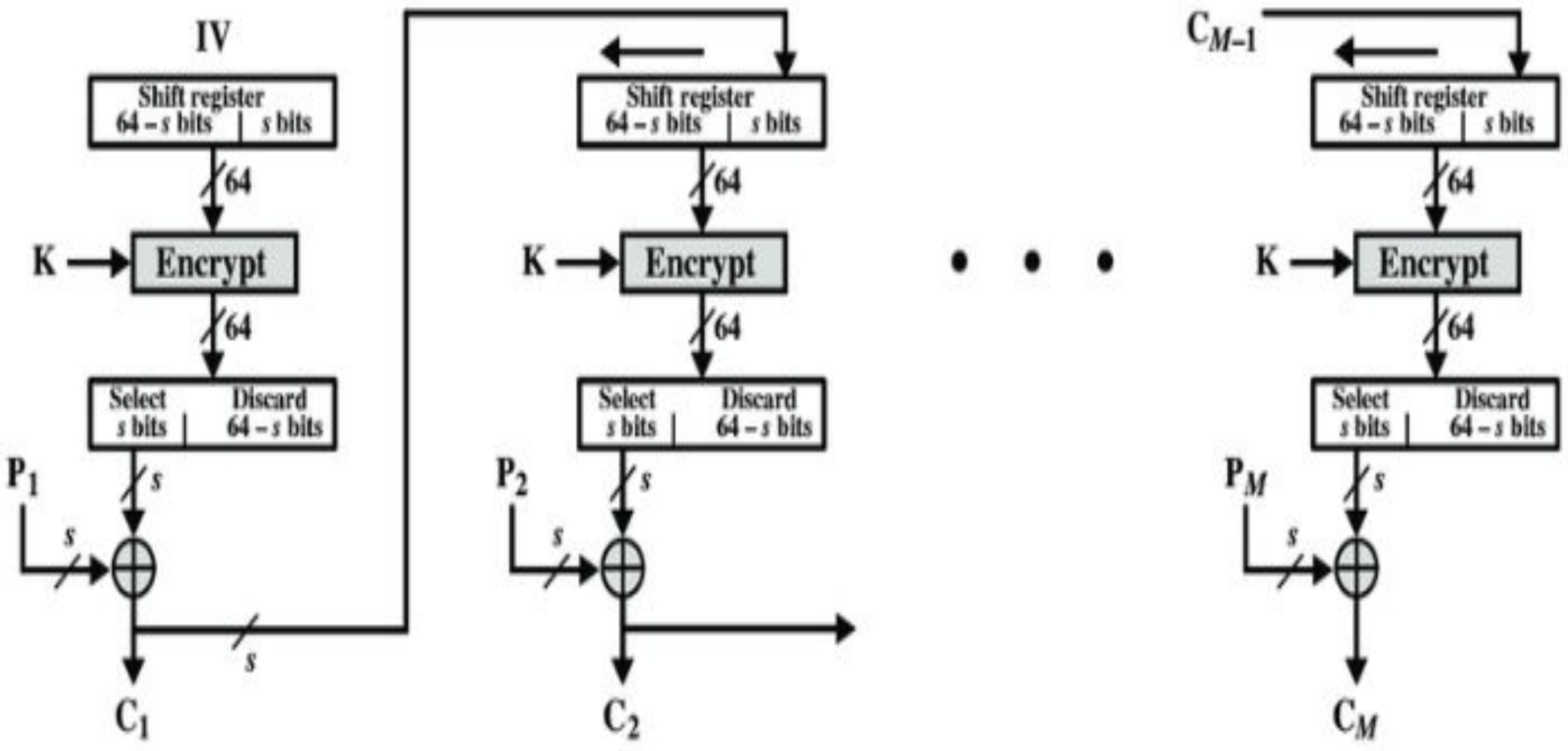
where the prime notation denotes bit complementation. This means that if an opponent can predictably change bits in IV, the corresponding bits of the received value of P_1 can be changed.

Figure 6.4. Cipher Block Chaining (CBC) Mode

[View full size image]



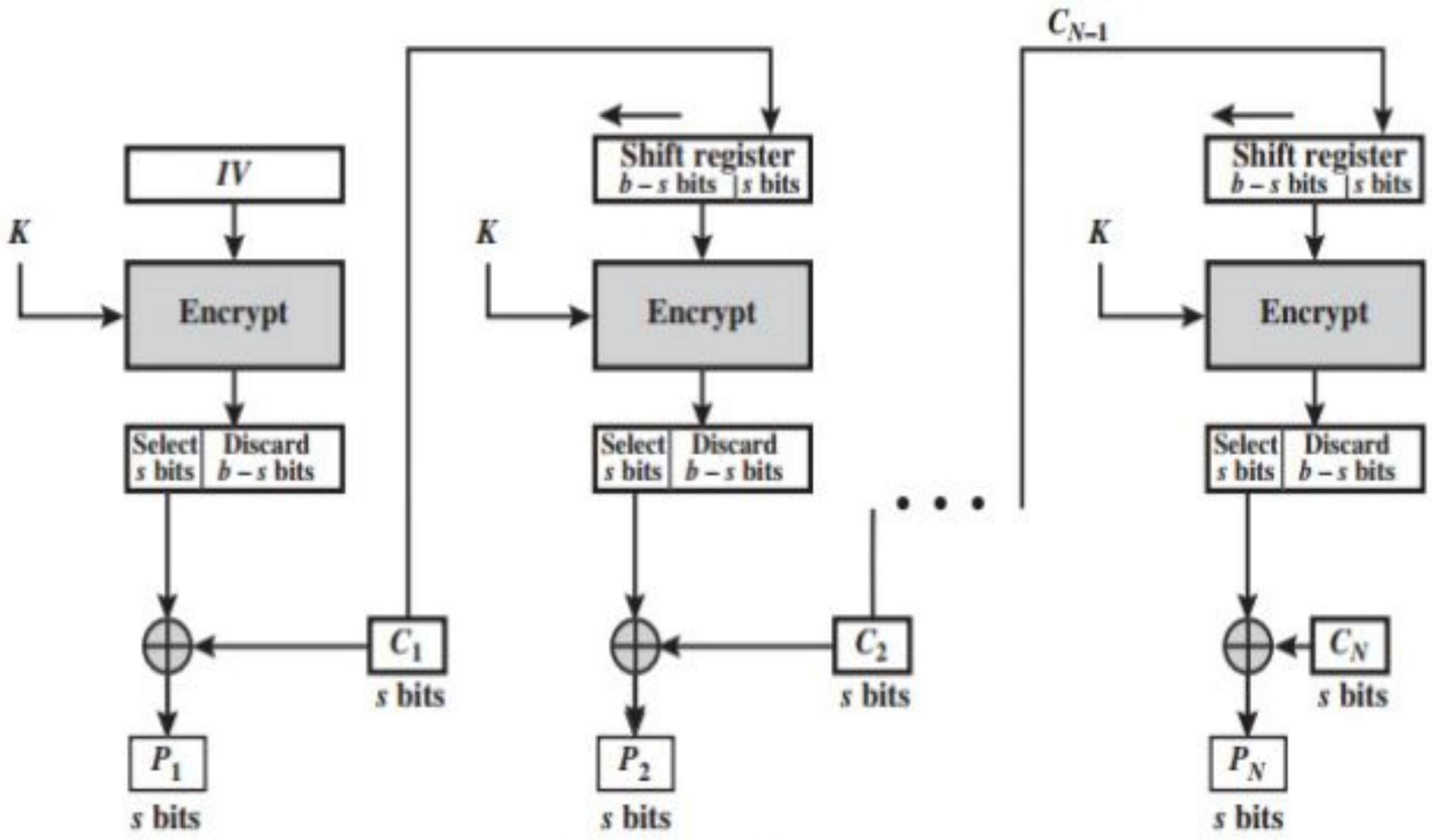
CFB mode:- (Cipher Feedback Mode)



(a) Encryption

Block Cipher Modes of Operation

CFB mode:- (Cipher Feedback Mode)



b) Decryption

Block Cipher Modes of Operation

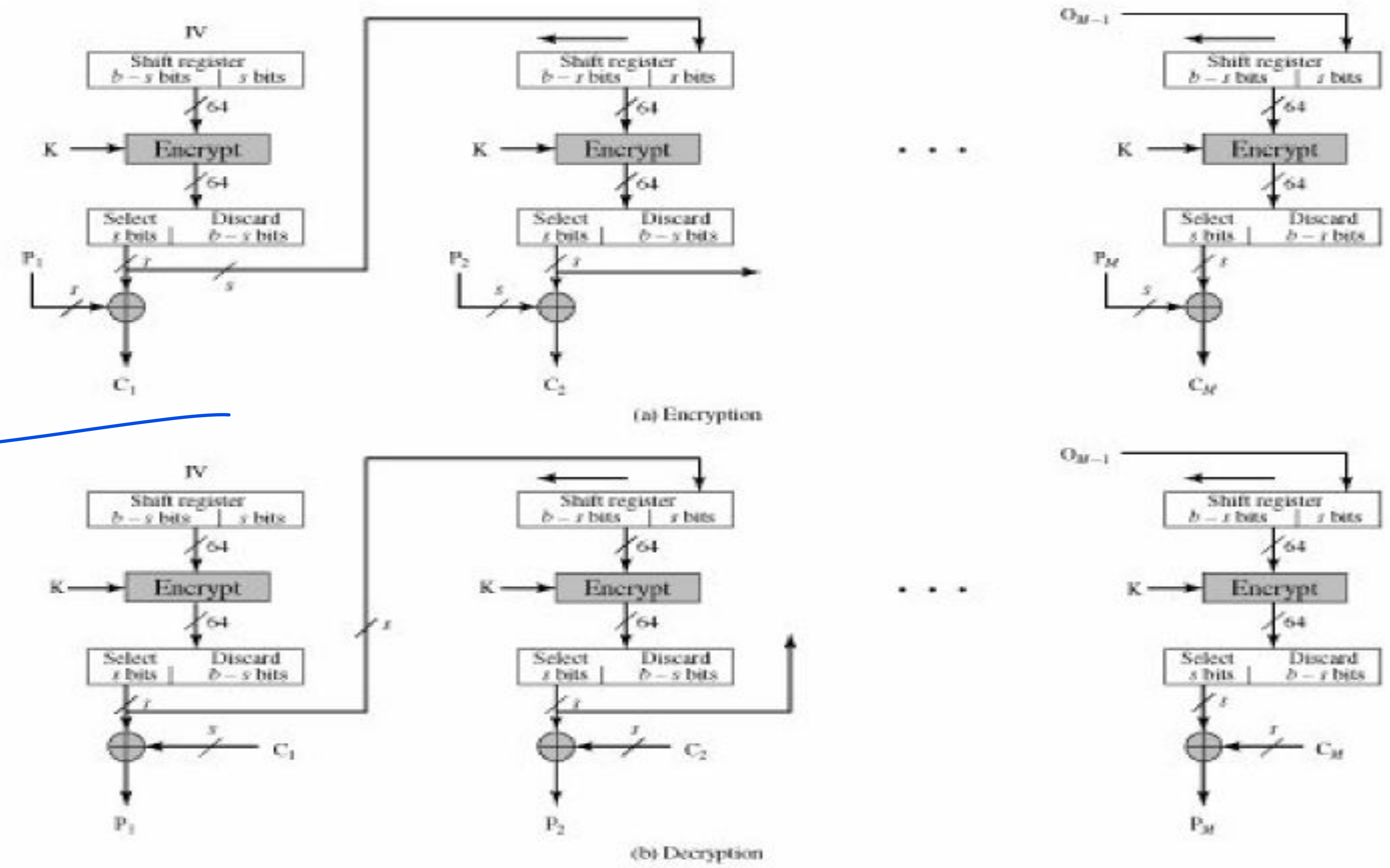
CFB mode:- (Cipher Feedback Mode)

Advantages

- It is convert block cipher into stream cipher/
- A stream cipher eliminates the need to pad a message to be an integral number of blocks.
- Thus if a character stream is being transmitted each character can be encrypted and transmitted immediately using a character oriented stream cipher

Block Cipher Modes of Operation

OFB mode:- (Output Feedback Mode)



Block Cipher Modes of Operation

Table 6.1. Block Cipher Modes of Operation

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 *j* 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