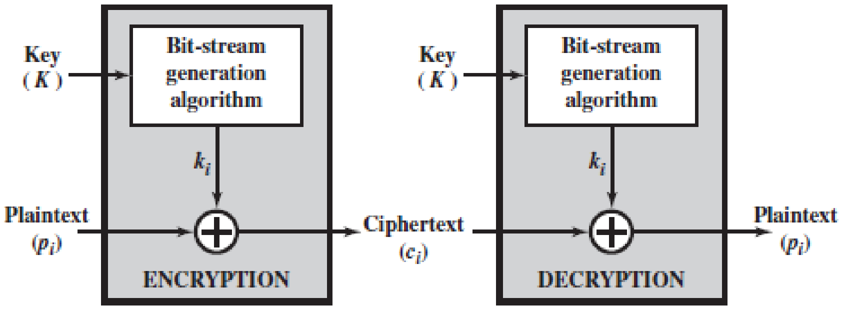
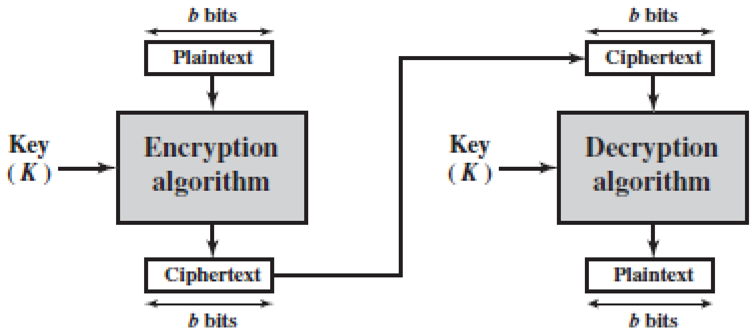
Topic 3
Symmetric Ciphers
Advanced Techniques

- **Stream Cipher and Block Cipher**

Stream Cipher	- Encrypts a digital data stream one bit or one byte at a time. Examples: Autokeyed Vigenère Cipher, Vernam Cipher - In the ideal case, a one-time pad version of the Vernam Cipher, which is unbreakable would be used, in which the keystream is as long as the plaintext bit stream 
Block Cipher	- 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. - The majority of network-based symmetric cryptographic applications make use of block ciphers. 

- **Diffusion and Confusion**

The terms *diffusion* and *confusion* were introduced by Claude Shannon to capture the two building blocks for any cryptographic system. Shannon's concern was to thwart cryptanalysis based on statistical analysis.

Diffusion	- Diffusion means that if we change a single bit of the plaintext, then statistically half of the bits in the ciphertext should change, and similarly, if we change one bit of the ciphertext, then approximately one half of the plaintext bits should change. - The idea of diffusion is to hide the relationship between the ciphertext and the plain text.
Confusion	- Confusion means that each bit of the ciphertext should depend on several parts of the key, obscuring the connections between the two. - The property of confusion hides the relationship between the ciphertext and the key. If a single bit in a key is changed, most or all the bits in the ciphertext will be affected.

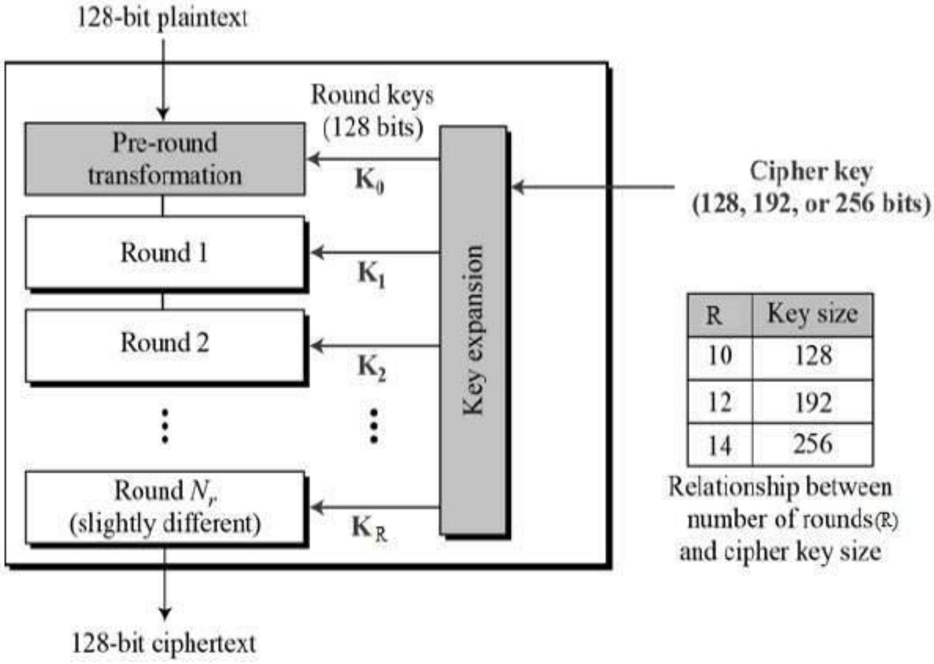
BLOCK CIPHERS

- **Feistel Cipher**

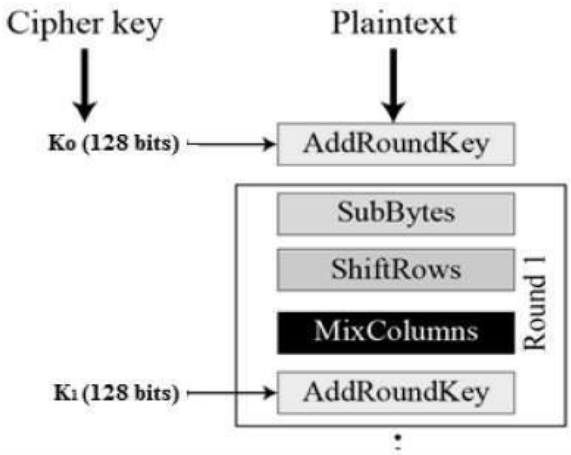
- **Data Encryption Standard (DES)**
  - DES was issued in 1977 by the National Bureau of Standards, now NIST, as Federal Information Processing Standard 46 (FIPS PUB 46).
  - DES was the most widely used encryption scheme until the introduction of the Advanced Encryption Standard (AES) in 2001.
  - Algorithm itself is referred to as the Data Encryption Algorithm (DEA). For DEA, data are encrypted in 64-bit blocks using a 56-bit key. The algorithm transforms 64-bit input in a series of steps into a 64-bit output. The same steps, with the same key, are used to reverse the encryption.
- **Block Cipher Design Principles**
  - **Number of Rounds**
    - The greater the number of rounds, the more difficult it is to perform cryptanalysis.
    - In general, the criterion should be that the number of rounds is chosen so that known cryptanalytic efforts require greater effort than a simple brute-force key search attack.
    - If DES had 15 or fewer rounds, differential cryptanalysis would require less effort than a brute-force key search
  - **Design of Function F**
    - The heart of a Feistel block cipher is the function F.
    - The more nonlinear F, the more difficult any type of cryptanalysis will be.
    - The algorithm should have good avalanche properties. The Strict Avalanche Criterion (SAC) and Bit Independence Criterion (BIC) appear to strengthen the effectiveness of the confusion function.
  - **Key Schedule Algorithm**
    - With any Feistel block cipher, the key is used to generate one subkey for each round.
    - In general, we would like to select subkeys to maximize the difficulty of deducing individual subkeys and the difficulty of working back to the main key.
    - It is suggested that, at a minimum, the key schedule should guarantee key/ciphertext Strict Avalanche Criterion and Bit Independence Criterion.

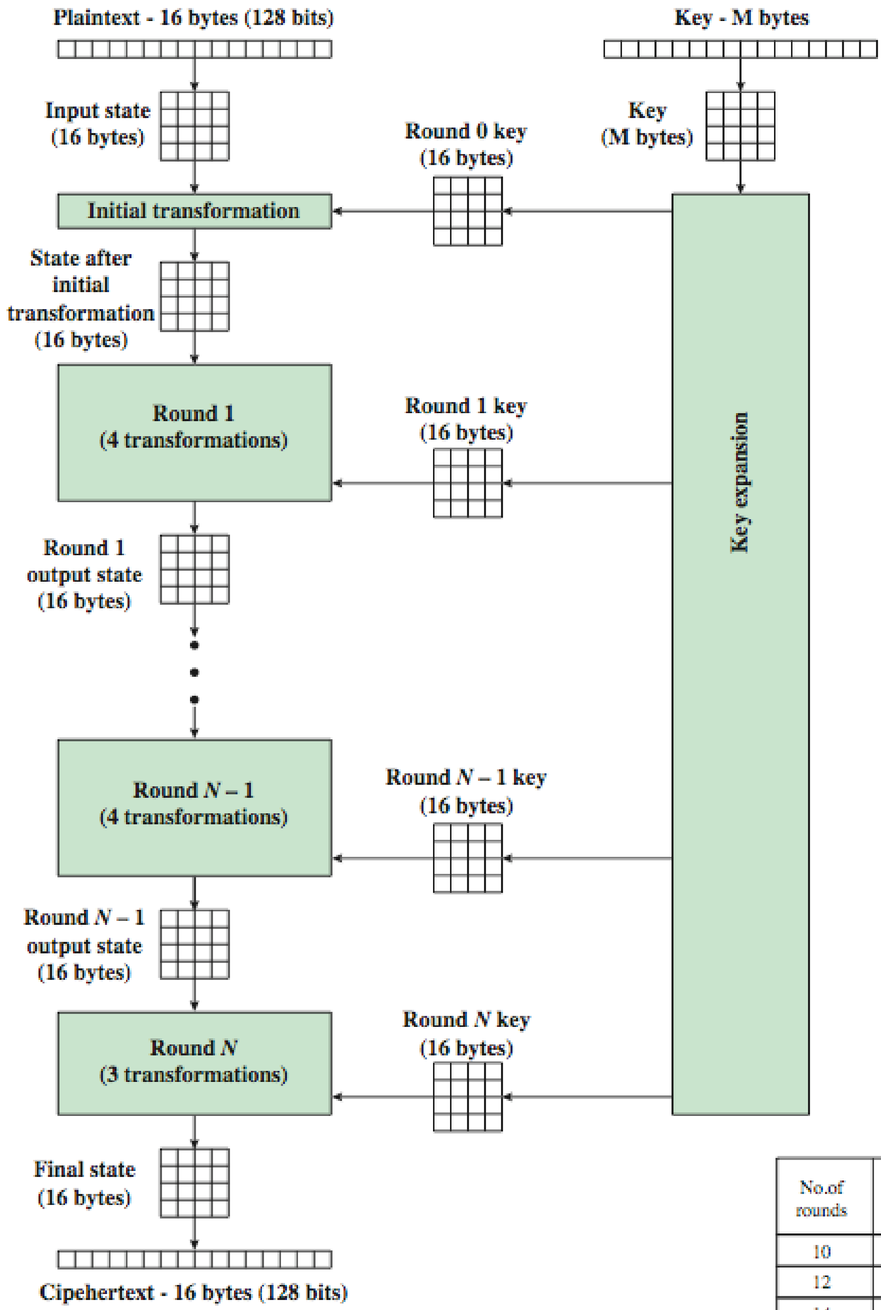
ADVANCED ENCRYPTION STANDARD

- **AES Schematic Diagram**



- **Encryption Process**





No.of rounds	Key Length (bytes)
10	16
12	24
14	32

Example:
Given,
Plaintext : {3F0EBF2BBC1589A33E21B2BDAD71A23C}
Key : {A17067B7C7A2B6045766FD827DD8B3B5}
Both are in Hexadecimal

1. Initial contents of the State Array displayed as a 4 x 4 matrix

Plaintext {3F0EBF2BBC1589A33E21B2BDAD71A23C}

3F	BC	3E	AD
0E	15	21	71
BF	89	B2	A2
2B	A3	BD	3C

Key {A17067B7C7A2B6045766FD827DD8B3B5}

A1	C7	57	7D
70	A2	66	D8
67	B6	FD	B3
B7	04	82	B5

2. Value of State after initial AddRoundKey

We do XOR for 3F $\oplus$ A1, 0E $\oplus$ 70 and so on. For example, 3F $\oplus$ A1, we need to convert those into binary and then do the XOR [If both bits are 1 then XORed bit will be 0, if both bits are 0 then XORed bit will be 0, if one bit is 0 and one bit is 1 then XORed bit will be 1] operation. Then we get the hexadecimal equivalent.

3F = 0011 1111
A1 = 1010 0001
$\oplus$ = 1001 1110

Hexadecimal equivalent of 1001 1110 = 9E

3F	BC	3E	AD
0E	15	21	71
BF	89	B2	A2
2B	A3	BD	3C

$\oplus$

A1	C7	57	7D
70	A2	66	D8
67	B6	FD	B3
B7	04	82	B5

=>

9E	7B	69	D0
7E	B7	47	A9
D8	3F	4F	11
9C	A7	3F	89

3. Value of State after SubBytes

In this step, we use a lookup table called S-box to perform a byte-by-byte substitution of the block. For example,

9E	Row 9 → Column E	0B

So, the value of State after SubBytes:

9E	7B	69	D0
7E	B7	47	A9
D8	3F	4F	11
9C	A7	3F	89

=>

0B	21	F9	70
F3	6C	A0	D3
61	75	84	82
DE	5C	75	A7

	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

Table 1: S-box

4. <u>Value of State after ShiftRows</u>

   In this step, a forward shift row transformation, called ShiftRows, is performed. The first row of State is not altered. For the second row, a 1-byte circular left shift is performed. For the third row, a 2-byte circular left shift is performed. For the fourth row, a 3-byte circular left shift is performed. So, the value of State after ShiftRows:

0B	21	F9	70
F3	6C	A0	D3
61	75	84	82
DE	5C	75	A7

=>

0B	21	F9	70
6C	A0	D3	F3
84	82	61	75
A7	DE	5C	75

5. <u>Value of State after MixColumns</u>

   In this step, a forward mix column transformation, called MixColumns, is performed on each column individually.