SYMETRIC ENCRYPTION

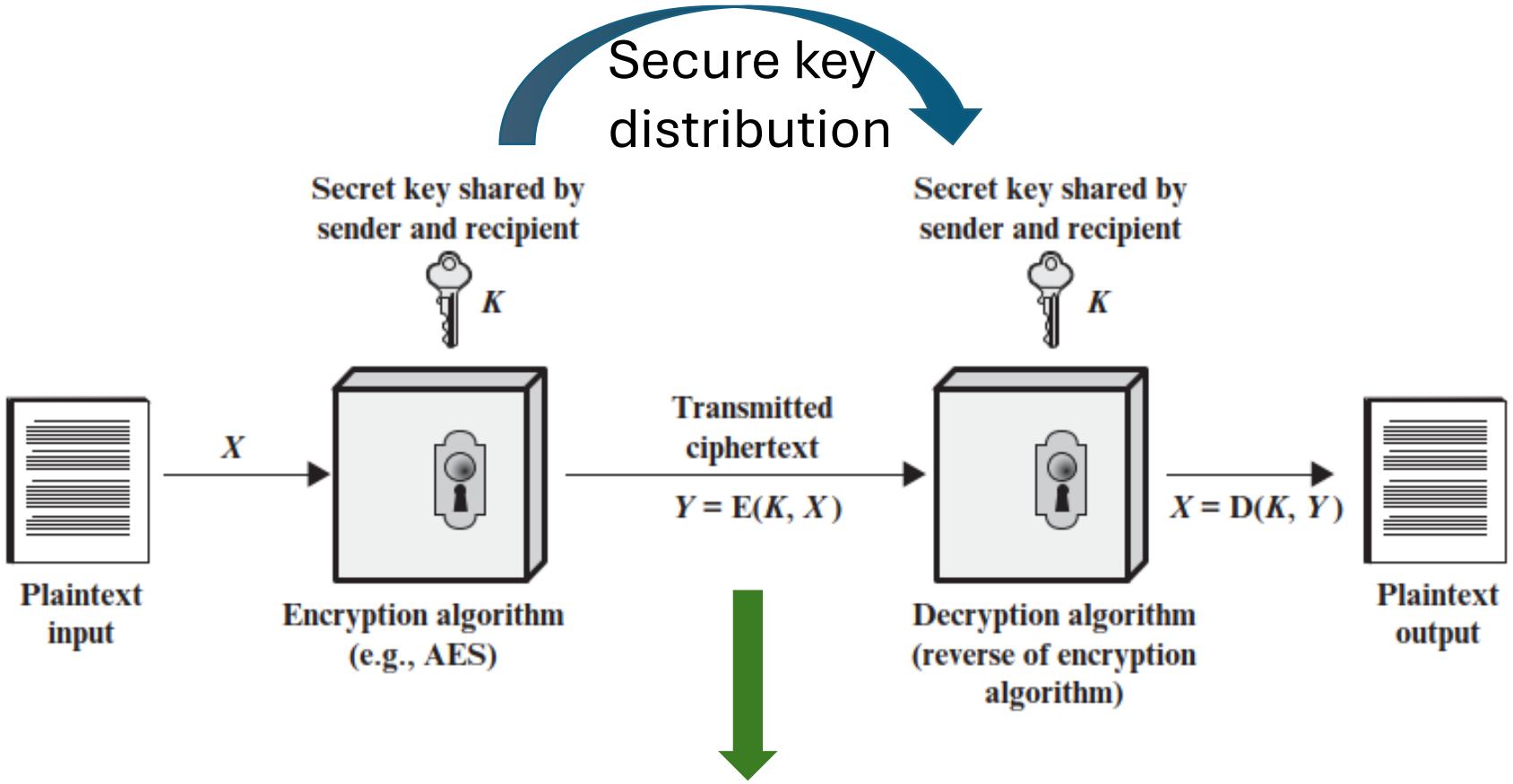
LECTURE CONTENT

- Feistel Cipher
- Data Encryption Standard (DES)
- Multiple Encryption DES (3DES)
- Advanced Encryption Standard (AES)

Learning objectives

- Block ciphers encrypt message in units called blocks
- Classical Cryptography

SYMMETRIC CIPHER MODEL



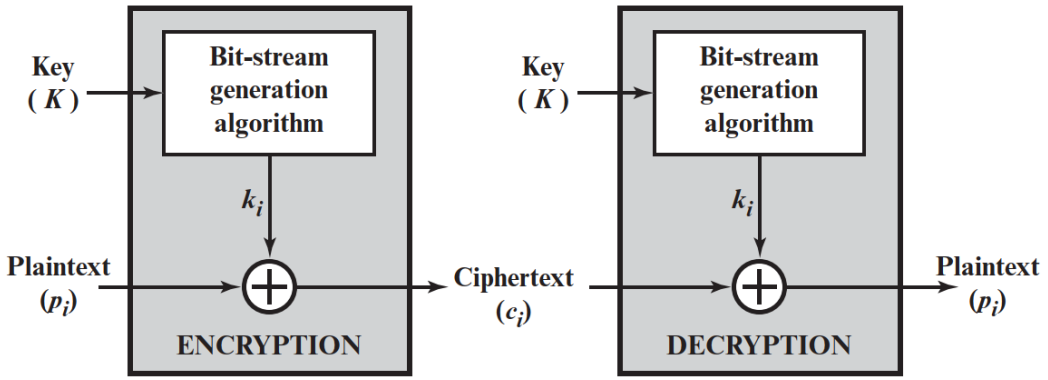
Cryptanalysis and brute-force attack

IMPORTANT FACTS IN THE MODERN CRYPTOGRAPHY

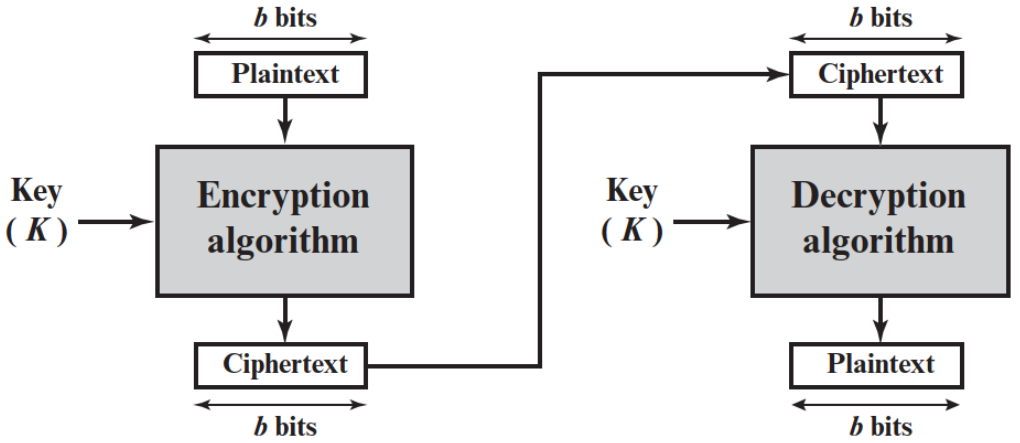
- In 1948, Claude Shannon released his study about Information Theory (Confusion and Difusion)
- In 1973, Feistel implemented Shannon's theory
- In 1977, the symmetric cryptography standard DES appeared
- In 1976, W. Diffie and M. Hellman released the study concerning the mathematical functions that involve two keys, called public key cipher or asymmetric cryptography
- In 1978, the asymmetric cryptography standard (RSA) is released
- In 2001, the new standard of symmetric cryptography (AES) is released

STREAM CIPHER V.S. BLOCK CIPHER

- A stream cipher is one that encrypts a digital data stream one bit or one byte at a time.
- 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.



Stream cipher



Block cipher

MOTIVATION OF BLOCK CIPHER

- A block cipher operates on a plaintext block of n bits to produce a ciphertext block of n bits.
- There are 2^n possible plaintext blocks.
- For the encryption to be reversible, each must produce a unique ciphertext block, leading to $2^n!$ transformations.
- If n is small, the system is vulnerable to a statistical analysis of the plaintext, e.g. the Monoalphabetic cipher.

Reversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	00
11	01

Irreversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	01
11	01

MOTIVATION OF BLOCK CIPHER

- Can we use long blocks with a reversible/simple/arbitrary substitution cipher?

  - Block size = n = 4 bits
    - Key size = n x 2^n bits
    - Number of possible transformations = $2^n!$
  - With a large block size is not practical from an implementation and performance point of view
    - n = 64 bits
    - No. of transformations = $2^{64}!$
  - **Problem:** Key size = 2^{70} bits ≈ 10^{20} bytes!!!!

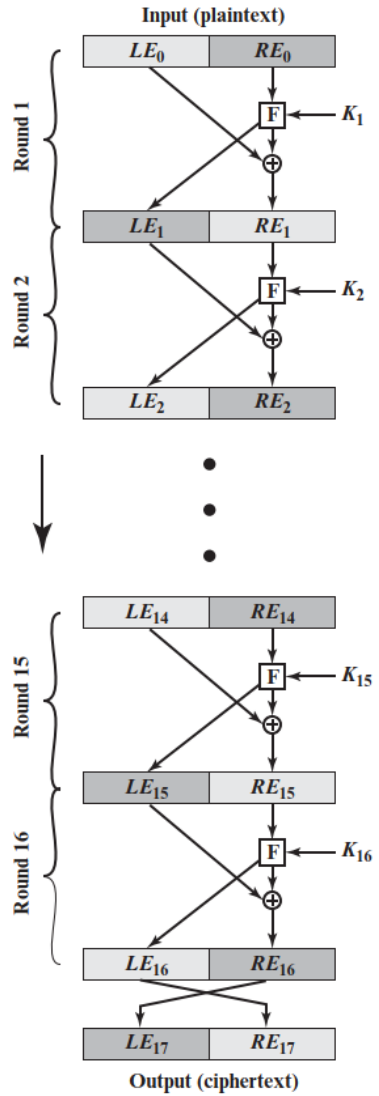
Plaintext	Ciphertext
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

FEISTEL CIPHER

- Feistel proposed to approximate the ideal block cipher by utilizing the concept of a product cipher.
- A product cipher is the execution of two or more simple ciphers in sequence, e.g. rotor machine.
- Feistel proposed the use of alternating substitutions and permutations:
  - **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.
- The purpose is to implement Claude Shannon's proposal to thwart cryptanalysis with *diffusion* and *confusion*.
  - **Confusion** - makes relationship between ciphertext and key as complex as possible
  - **Diffusion** – dissipates statistical structure of plaintext over bulk of ciphertext

FEISTEL CIPHER STRUCTURE

- Block size:
  - Larger block size → greater security
  - Larger block size → reduced encryption/decryption speed
  - Typical sizes → 64/128 bits
- Key:
  - Key is regenerated in each round
- Number of rounds:
  - Single round is not enough
  - Multiple rounds offer increasing security
  - Typical value is 16 rounds
- Subkey generation algorithm
  - Increasing complexity
- Round function F
  - Further increasing complexity

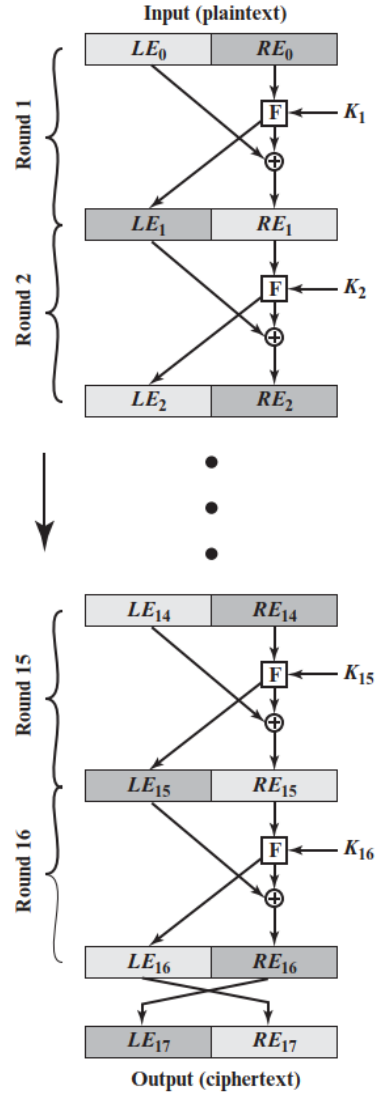


FEISTEL CIPHER STRUCTURE

- Formula of encryption:

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

  - i – round index, [1,16]
  - LE_i – left half block of round i
  - RE_i – right half block of round i
  - $F(RE_{i-1},K_i)$ – round function



FEISTEL CIPHER STRUCTURE

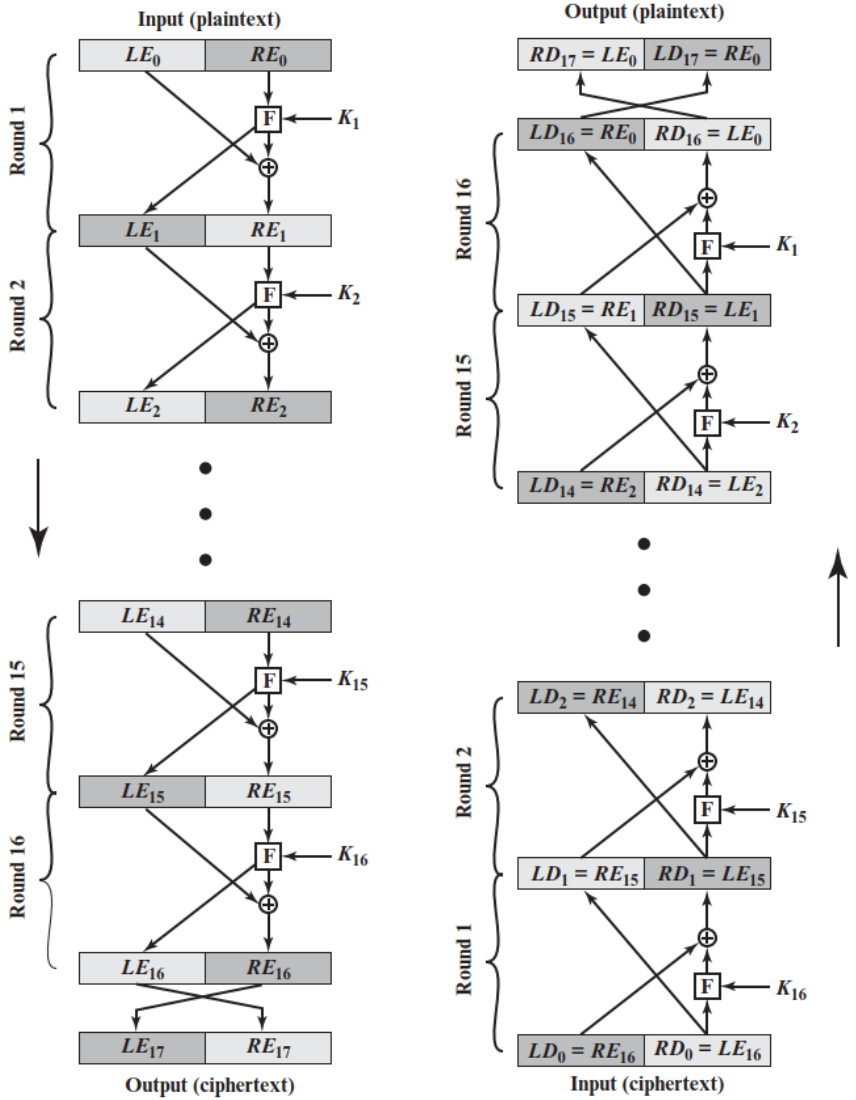
- Formula of encryption :

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

- Formula of decryption :

$$LD_i = RD_{i-1}$$
$$RD_i = LD_{i-1} \oplus F(RD_{i-1}, K_{17-i})$$

- Encryption and decryption can share the same implementation!



AVALANCHE EFFECT

- A property of the Feistel Cipher Structure is Avalanche Effect
- A change of **one** input bit or key bit should result in changing approximately **half** of output bits!
- Making attempts to guess the key by using different Plaintext – Ciphertext pairs should be impossible

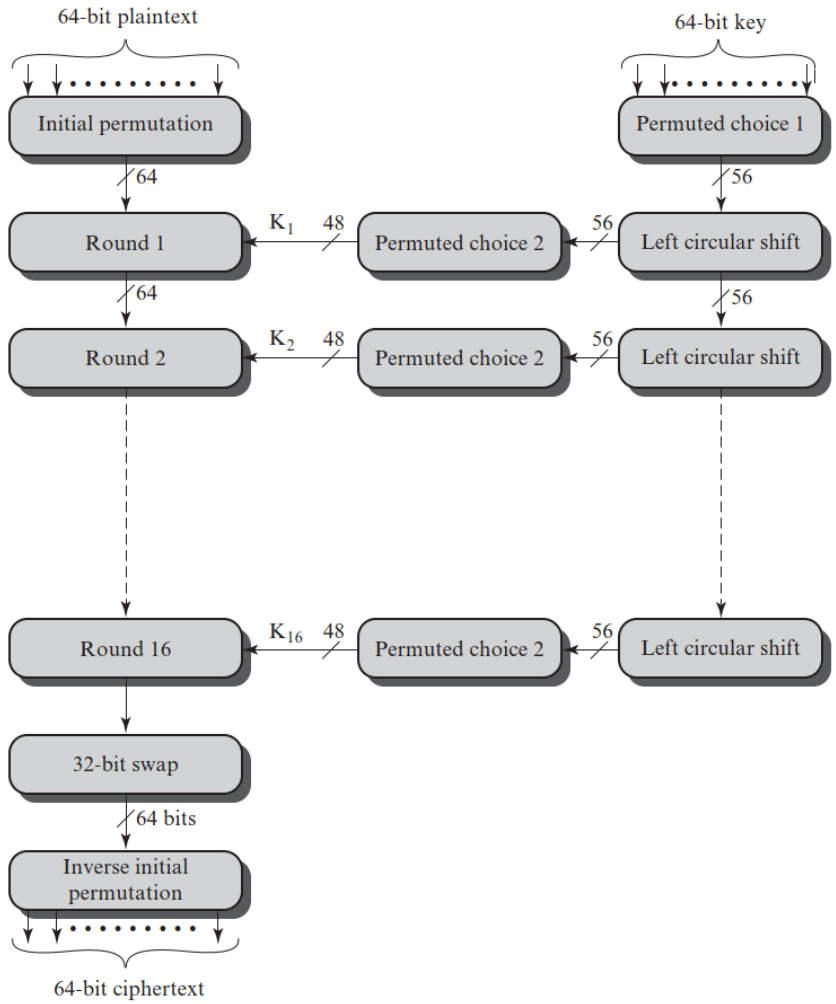
DATA ENCRYPTION STANDARD (DES)

- DES is based on the Feistel Cipher Structure
- One of the most widely used block cipher in world
- Adopted in 1977 by NIST
- Encrypts 64-bit data using 56-bit key
- DES has become widely used, especially in financial applications

DES ENCRYPTION

Break message into 8-byte (64-bit) blocks

- Each block broken into 32-bit halves
- Initial permutation
- 16 rounds of scrambling
- Final (reverse) permutation



Encryption algorithm structure:

- Initial and final permutation
- Round
  - Scrambling *F* function
- Key schedule

INITIAL AND FINAL PERMUTATION

- First and final steps of the data computation
- IP reorders the input data bits and IP^{-1} is the inverse

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

IP^{-1}

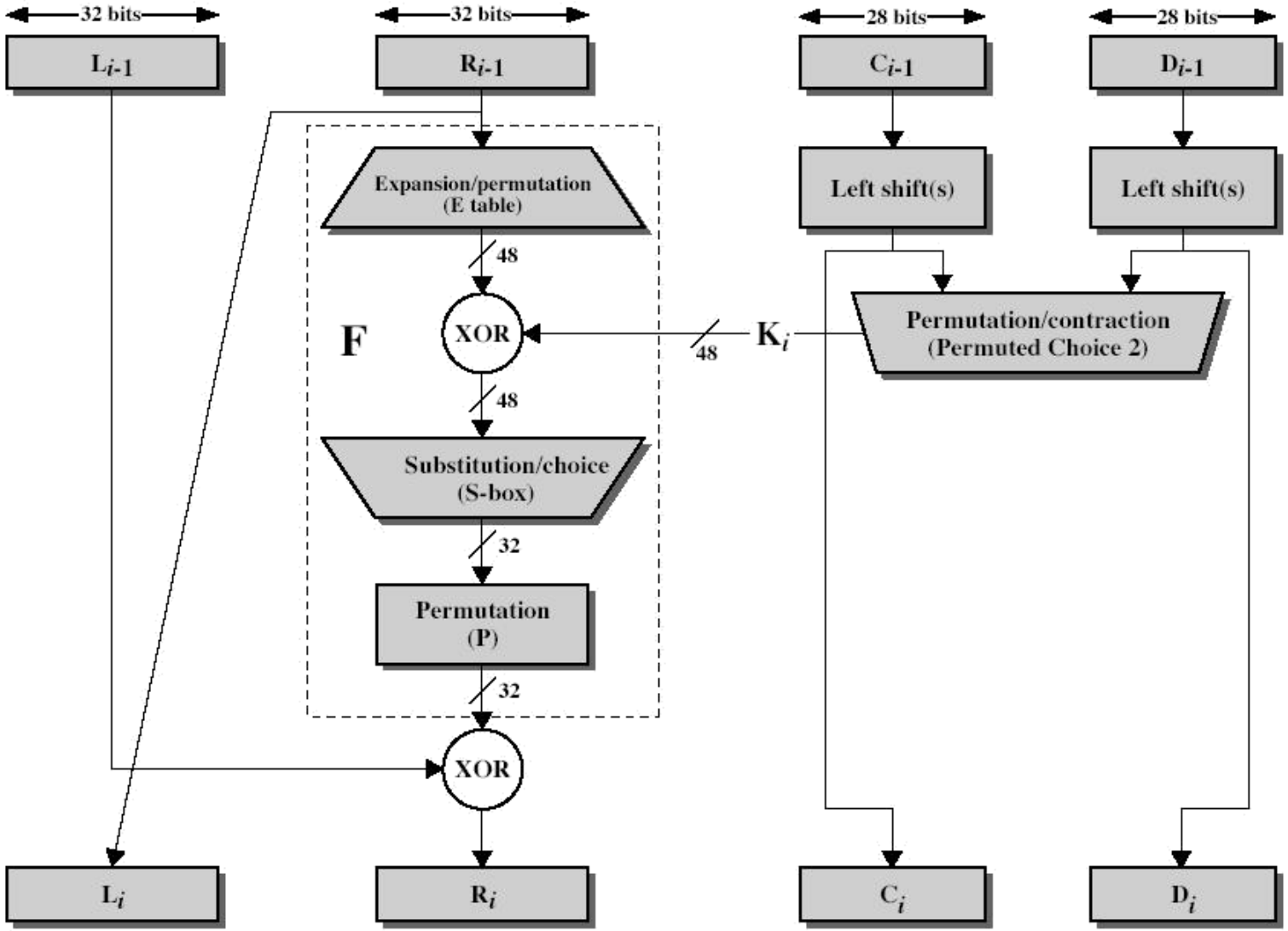
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

- Example:

IP(675a6967 5e5a6b5a) = (ffb2194d 004df6fb)

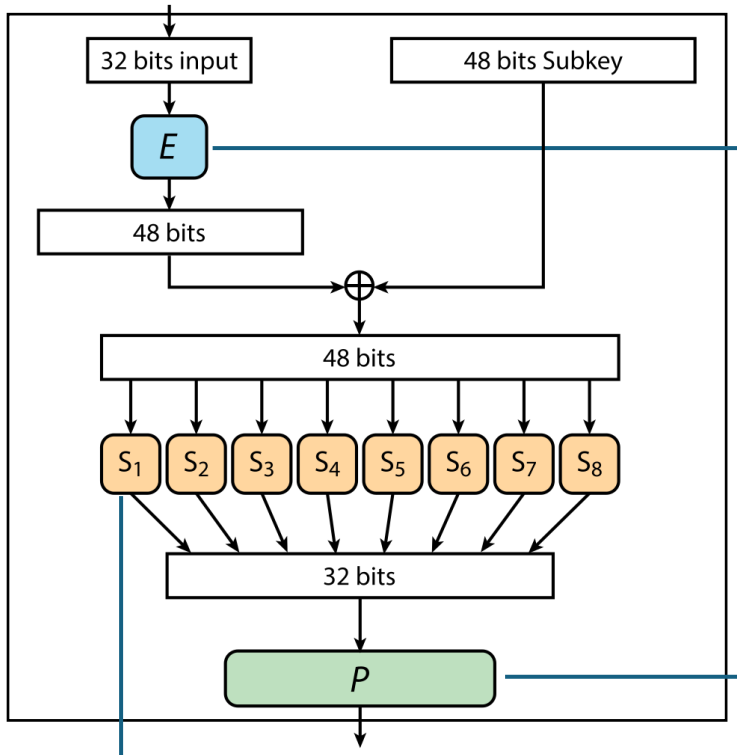
1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61
0110	0111	0101	1010	0110	1001	0110	0111	0101	1110	0101	1010	0110	1011	0101	1010
1111	1111	1011	0010	0001	1001	0100	1101	0000	0000	0100	1101	1111	0110	1111	1011

SINGLE ROUND OF DES ALGORITHM



SINGLE ROUND OF DES ALGORITHM

F function



E

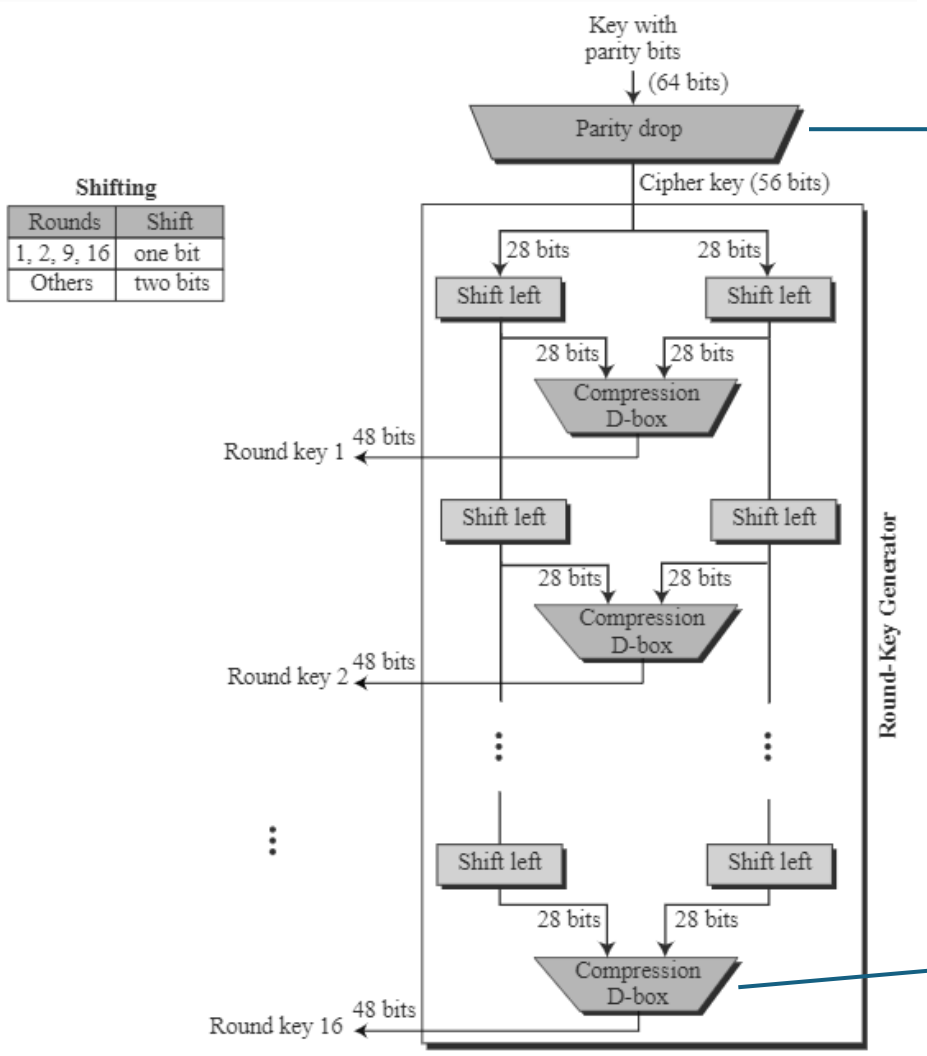
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

P

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

S_1	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0yyyy1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
1yyyy0	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
1yyyy1	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

DES KEY SCHEDULE



Shifting

Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits

Parity Drop Table

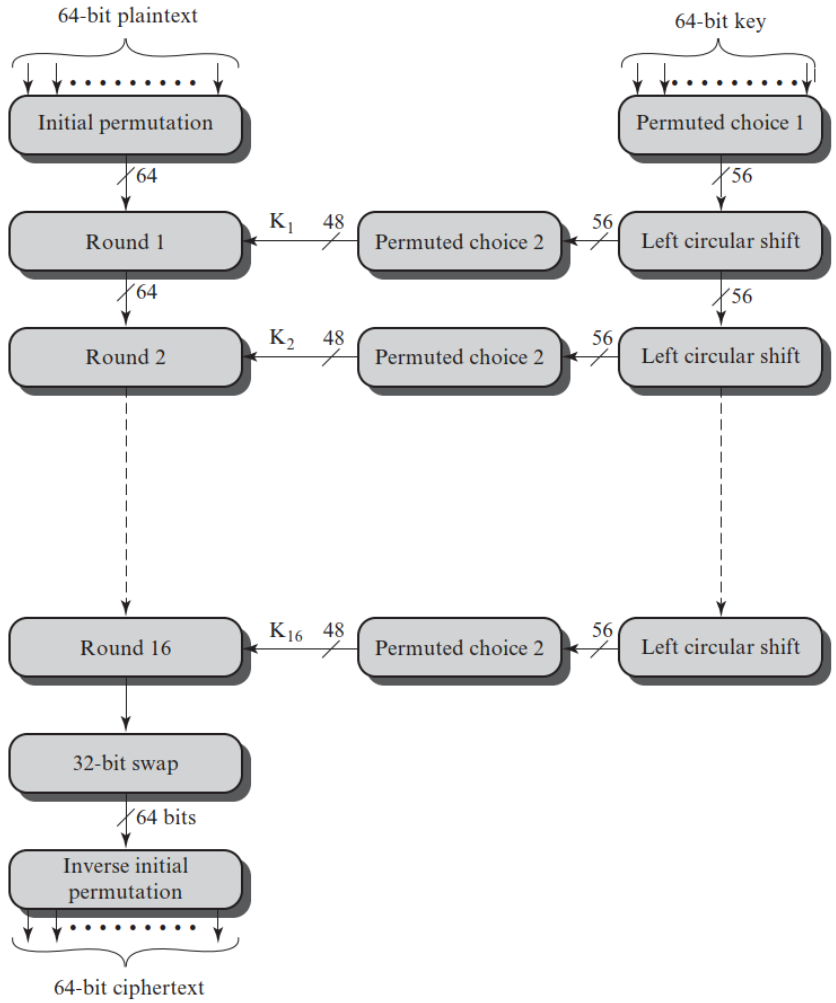
1	2	3	4	5	6	7	**8**
9	10	11	12	13	14	15	**16**
17	18	19	20	21	22	23	**24**
25	26	27	28	29	30	31	**32**
33	34	35	36	37	38	39	**40**
41	42	43	44	45	46	47	**48**
49	50	51	52	53	54	55	**56**
57	58	59	60	61	62	63	**64**

Key Compression Table

	1	2	3	4	5	6	7	8
1	14	17	11	24	01	05	03	28
2	15	06	21	10	23	19	12	04
3	26	08	16	07	27	20	13	02
4	41	52	31	37	47	55	30	40
5	51	45	33	48	44	49	39	56
6	34	53	46	42	50	36	29	32

DES DECRYPTION

- Same algorithm is used for decryption.
- The application of subkeys is reversed
- The initial and final permutations are reversed.



AVALANCHE EFFECT OF DES

1-bit change in plaintext

Round		δ
	02468aceeca86420 12468aceeca86420	1
1	3cf03c0fbad22845 3cf03c0fbad32845	1
2	bad2284599e9b723 bad3284539a9b7a3	5
3	99e9b7230bae3b9e 39a9b7a3171cb8b3	18
4	0bae3b9e42415649 171cb8b3ccaca55e	34
5	4241564918b3fa41 ccaca55ed16c3653	37
6	18b3fa419616fe23 d16c3653cf402c68	33
7	9616fe2367117cf2 cf402c682b2cefbc	32
8	67117cf2c11bfc09 2b2cefbc99f91153	33

Round		δ
9	c11bfc09887fbc6c 99f911532eed7d94	32
10	887fbc6c600f7e8b 2eed7d94d0f23094	34
11	600f7e8bf596506e d0f23094455da9c4	37
12	f596506e738538b8 455da9c47f6e3cf3	31
13	738538b8c6a62c4e 7f6e3cf34bc1a8d9	29
14	c6a62c4e56b0bd75 4bc1a8d91e07d409	33
15	56b0bd7575e8fd8f 1e07d4091ce2e6dc	31
16	75e8fd8f25896490 1ce2e6dc365e5f59	32
IP^{-1}	da02ce3a89ecac3b 057cde97d7683f2a	32

Key:12468aceeca86420

1-bit change in key

Round		δ
	02468aceeca86420 02468aceeca86420	0
1	3cf03c0fbad22845 3cf03c0f9ad628c5	3
2	bad2284599e9b723 9ad628c59939136b	11
3	99e9b7230bae3b9e 9939136b768067b7	25
4	0bae3b9e42415649 768067b75a8807c5	29
5	4241564918b3fa41 5a8807c5488dbe94	26
6	18b3fa419616fe23 488dbe94aba7fe53	26
7	9616fe2367117cf2 aba7fe53177d21e4	27
8	67117cf2c11bfc09 177d21e4548f1de4	32

Round		δ
9	c11bfc09887fbc6c 548f1de471f64dfd	34
10	887fbc6c600f7e8b 71f64dfd4279876c	36
11	600f7e8bf596506e 4279876c399fdc0d	32
12	f596506e738538b8 399fdc0d6d208dbb	28
13	738538b8c6a62c4e 6d208dbbb9bdeeaa	33
14	c6a62c4e56b0bd75 b9bdeeaad2c3a56f	30
15	56b0bd7575e8fd8f d2c3a56f2765c1fb	27
16	75e8fd8f25896490 2765c1fb01263dc4	30
IP^{-1}	da02ce3a89ecac3b ee92b50606b62b0b	30

Key1: 0f1571c947d9e859

Key2: 1f1571c947d9e859

STRENGTH OF DES

- **Time to break DES**
  - Number of keys: 2^{56} = 7.2×10^{16} keys
    - On the *average* you need to search through 2^{55} keys (half of all possible keys must be tried to achieve success.)
    - In the worst case you need to search all 2^{56} keys
  - If you can do one encryption/decryption in 1 clock cycle @ 500 MHz
    - Time taken to check ONE key = $1/(500 \times 10^6)$ s
    - Time taken to check 2^{55} keys = $2^{55}/(500 \times 10^6)$ s = 72,057,594.04 s /3600 = 20016 hours /24 = 834 days
- The **hertz** (symbol: **Hz**) is defined as the number of cycles per second (MHz = 10^6 Hz)
- **Nowadays technology**
  - A single PC can break DES in about a year
  - If 100 PCs work in parallel, it only takes 3-4 days.

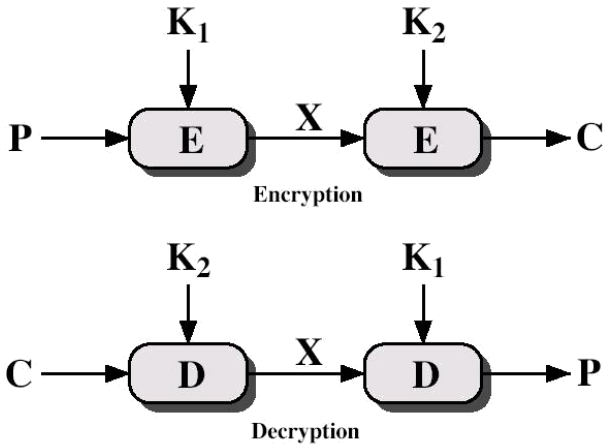
REPLACEMENT OF DES

- It is necessary to design a replacement for DES, leading to two solutions:
  - Triple-DES (3DES)
  - Advanced Encryption Standard (AES)

WHY TRIPLE-DES?

- Why not Double-DES?
  - Key length = 112 bits

$$C = E_{K2}[E_{K1}[P]]$$

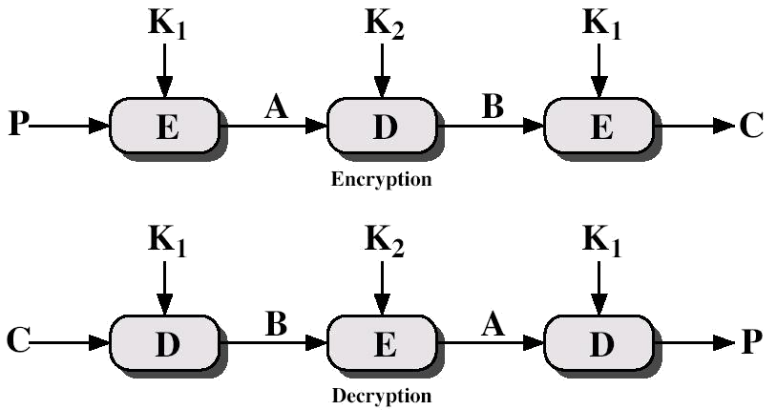


- Meet-in-the-middle attack
  - Since $X = E_{K1}[P] = D_{K2}[C]$
  - Attack by encrypting P with all keys and store
  - Then decrypt C with keys and match X value
  - It takes $O(2^{56})$ steps

TRIPLE-DES WITH TWO-KEYS

- Use 2 keys with E-D-E sequence
  - Key length = 112 bits

$$C = E_{K1}[D_{K2}[E_{K1}[P]]]$$

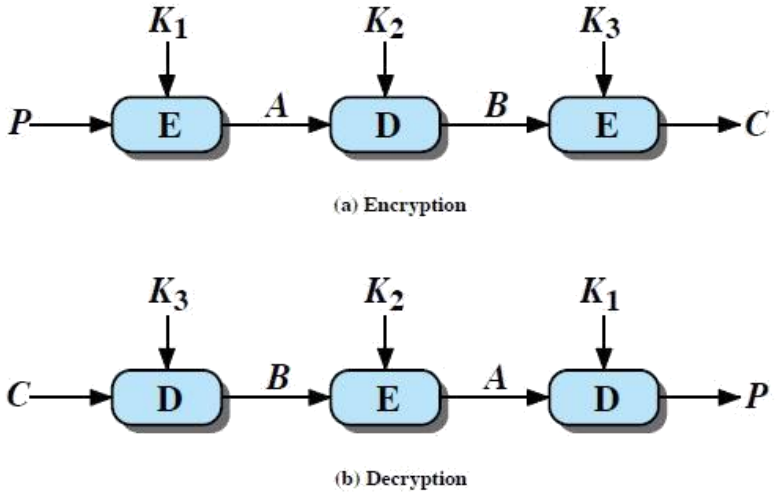


- If K1=K2 then can work with single DES, no new hardware is required for single DES.
- No current known practical attacks for 2-key 3DES

TRIPLE-DES WITH THREE-KEYS

- Although there are no practical attacks on two-key Triple-DES, there are some theoretical ones
- Triple-DES with Three-Keys can be used to avoid even these
  - Key length = 168 bits

$$C = E_{K3}[D_{K2}[E_{K1}[P]]]$$



- Backward compatibility with DES ($K_3=K_2=K_1$)
- Has been adopted by some Internet applications

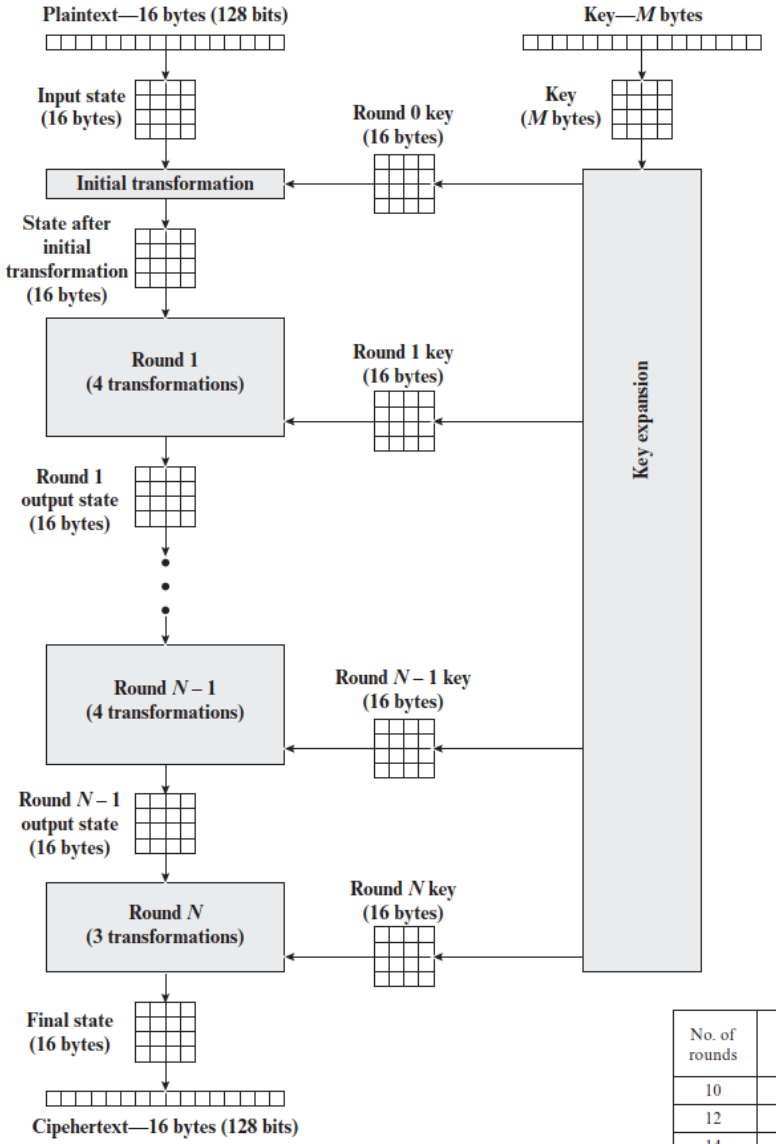
ADVANCED ENCRYPTION STANDARD (AES)

- It was clearly needed a replacement for DES
  - Theoretical attacks that can break it
  - Have demonstrated exhaustive key search attacks
- It can be used Triple-DES – but slow with small blocks
- US NIST: call for candidates for Advanced Encryption Standard (AES) in 1997
- 15 candidates accepted in Jun 98, and 5 were shortlisted in Aug-99
  - MARS (IBM) - complex, fast, high security margin
  - RC6 (USA) - v. simple, v. fast, low security margin
  - Rijndael (Belgium) - clean, fast, good security margin
  - Serpent (Euro) - slow, clean, v. high security margin
  - Twofish (USA) - complex, v. fast, high security margin
- Rijndael was selected as the AES in Oct-2000
- Issued as FIPS PUB 197 standard in Nov-2001

FEATURES OF AES

- Designed by Rijmen-Daemen in Belgium
- Block size: 128 bits
- Key sizes: 128/192/256 bits
- Variable rounds: 10/12/14 rounds
- Resistant against known attacks
- Speed and code compactness on many CPUs

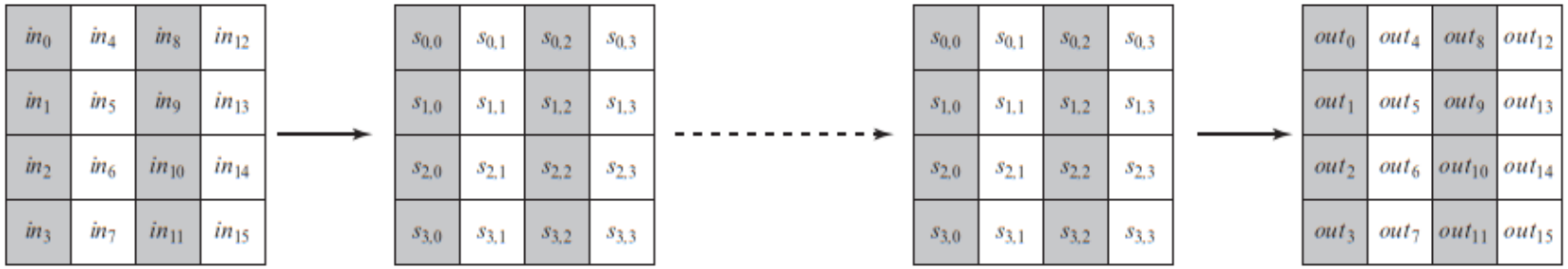
STRUCTURE OF AES



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

DATA STRUCTURE OF AES

- Processes data as 4 groups of 4 bytes (128 bits) or 4x4 matrix state
- Key expansion: takes 128-bit (16-byte) key and expands into an array of 44 32-bit words

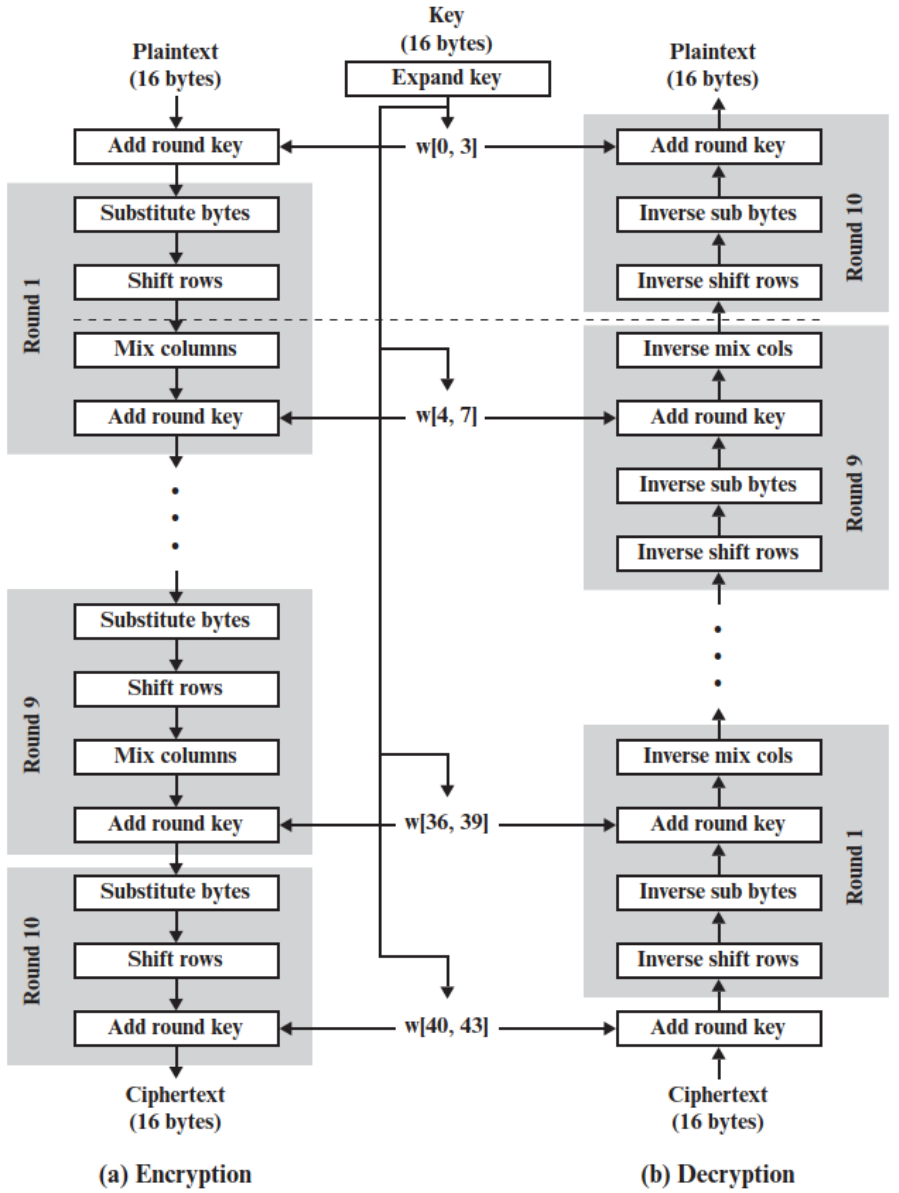


(a) Input, state array, and output

(b) Key and expanded key

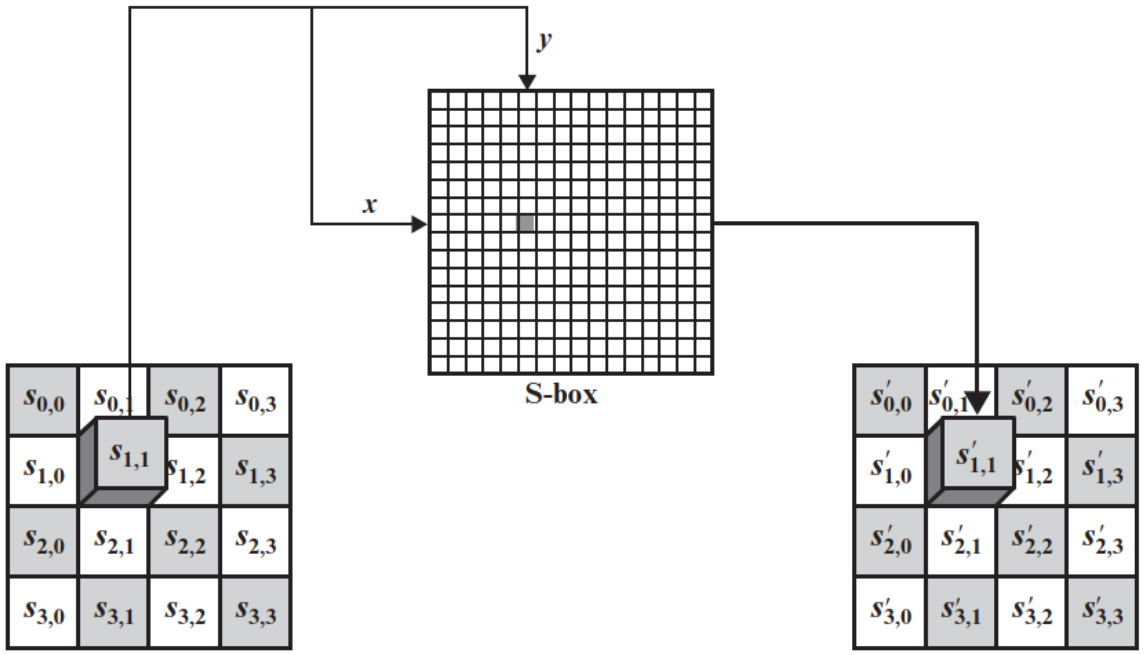
AES ENCRYPTION AND DECRYPTION

- In AES, each round is not Feistel network
- Each round has four operations:
  - Substitute
  - Shift rows
  - Mix columns
  - Add round key



AES ROUND (BYTE SUBSTITUTION)

- Byte substitution (1 S-box of 16x16 used on every byte)



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
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

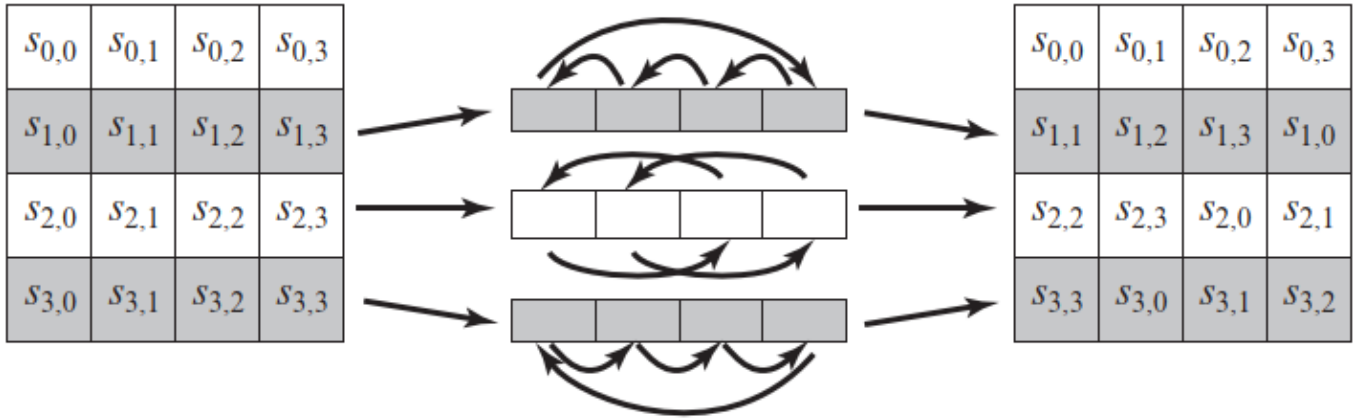
Inverse S-box

x \ y	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

- Inverse Byte substitution: Inverse S-box
  - IS-box(S-box(a)) = a

AES ROUND (SHIFT ROWS)

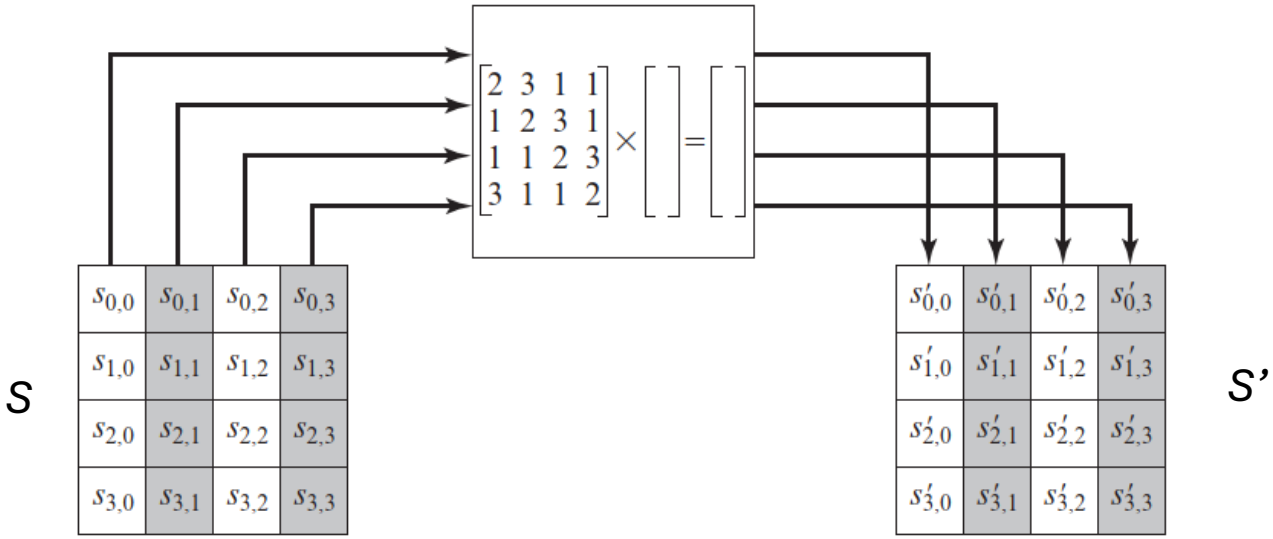
- Shift rows (permute bytes in each row)
  - Circular left shift



  - Inverse shift rows: circular right shift

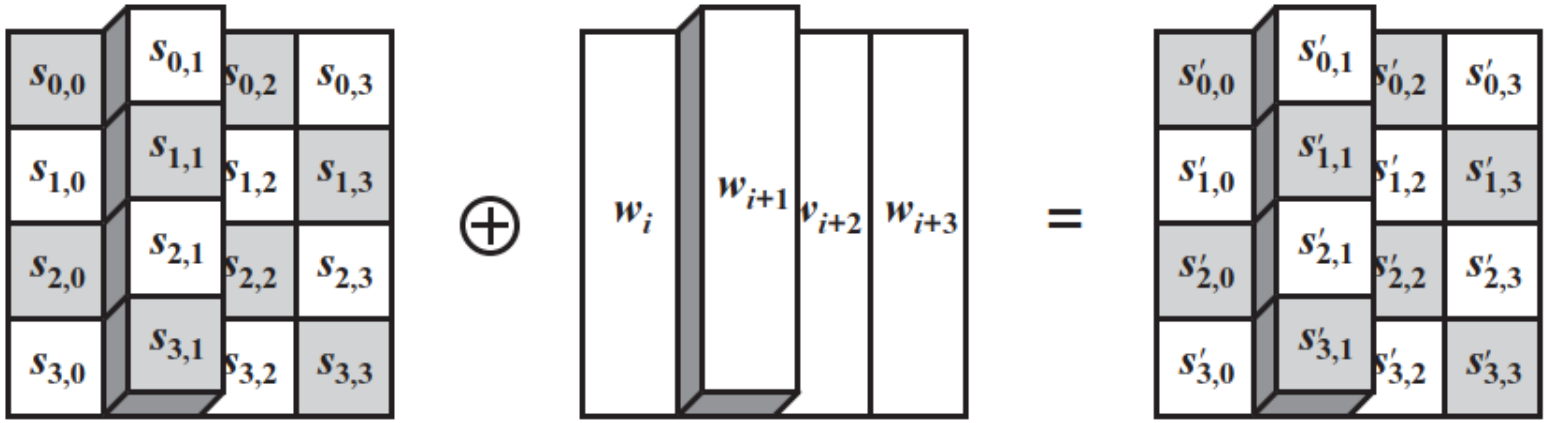
AES ROUND (MIX COLUMNS)

- Mix columns (subs using matrices multiplication)
  - $M \cdot S = S'$
  - Example: $S'_{0,0} = 2 \oplus S_{0,0} + 3 \oplus S_{1,0} + S_{2,0} + S_{3,0}$



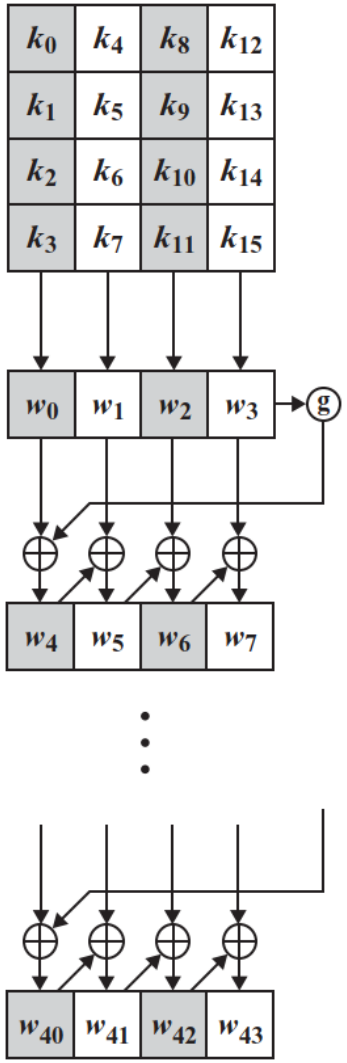
  - Inverse mix columns: $\exists M^{-1} \mid M^{-1} \cdot M = I$

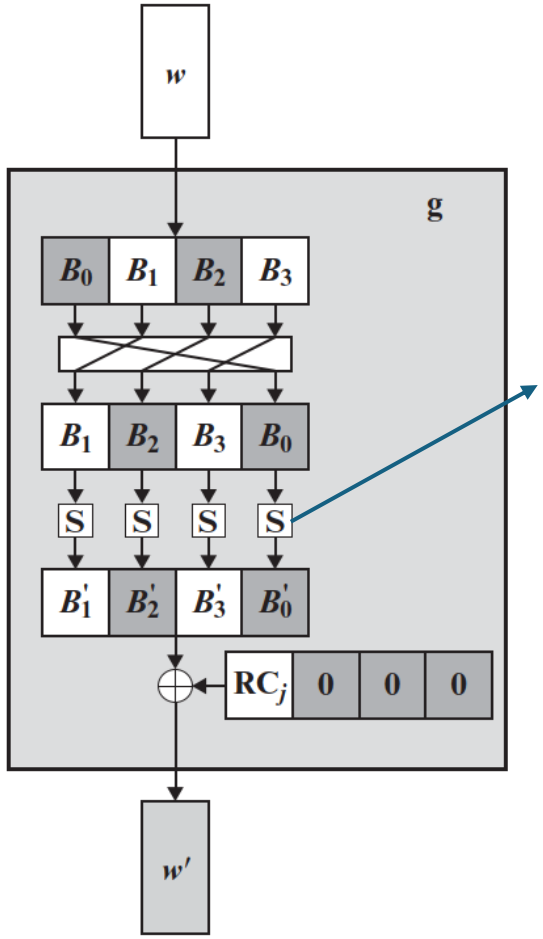
AES ROUND (ADD ROUND KEY)



(Add Round Key)

AES KEY GENERATION





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
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Same S-box is used here

j	1	2	3	4	5	6	7	8	9	10
RC[j]	01	02	04	08	10	20	40	80	1B	36

AVALANCHE EFFECT IN AES

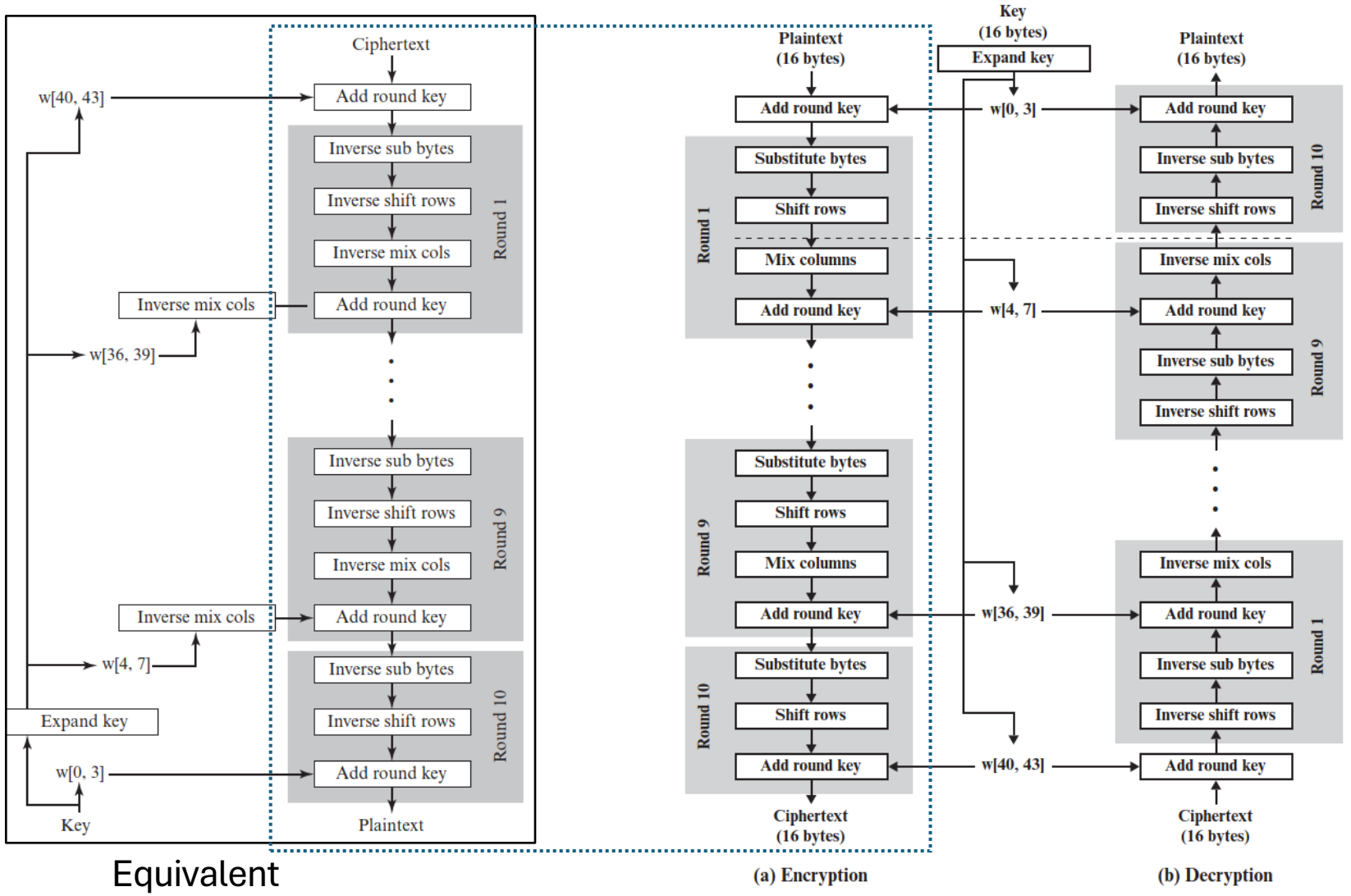
Round		Number of Bits that Differ
	0123456789abcdeffedcba9876543210 0023456789abcdeffedcba9876543210	1
0	0e3634aece7225b6f26b174ed92b5588 0f3634aece7225b6f26b174ed92b5588	1
1	657470750fc7ff3fc0e8e8ca4dd02a9c c4a9ad090fc7ff3fc0e8e8ca4dd02a9c	20
2	5c7bb49a6b72349b05a2317ff46d1294 fe2ae569f7ee8bb8c1f5a2bb37ef53d5	58
3	7115262448dc747e5cdac7227da9bd9c ec093dfb7c45343d689017507d485e62	59
4	f867aee8b437a5210c24c1974cffeabc 43efdb697244df808e8d9364ee0ae6f5	61
5	721eb200ba06206dcbd4bce704fa654e 7b28a5d5ed643287e006c099bb375302	68
6	0ad9d85689f9f77bc1c5f71185e5fb14 3bc2d8b6798d8ac4fe36a1d891ac181a	64
7	db18a8ffa16d30d5f88b08d777ba4eaa 9fb8b5452023c70280e5c4bb9e555a4b	67
8	f91b4fbfe934c9bf8f2f85812b084989 20264e1126b219aef7feb3f9b2d6de40	65
9	cca104a13e678500ff59025f3bafaa34 b56a0341b2290ba7dfdfbddcd8578205	61
10	ff0b844a0853bf7c6934ab4364148fb9 612b89398d0600cde116227ce72433f0	58

1-bit change in plaintext

Round		Number of Bits that Differ
	0123456789abcdeffedcba9876543210 0123456789abcdeffedcba9876543210	0
0	0e3634aece7225b6f26b174ed92b5588 0f3634aece7225b6f26b174ed92b5588	1
1	657470750fc7ff3fc0e8e8ca4dd02a9c c5a9ad090ec7ff3fc1e8e8ca4cd02a9c	22
2	5c7bb49a6b72349b05a2317ff46d1294 90905fa9563356d15f3760f3b8259985	58
3	7115262448dc747e5cdac7227da9bd9c 18aeb7aa794b3b66629448d575c7cebf	67
4	f867aee8b437a5210c24c1974cffeabc f81015f993c978a876ae017cb49e7eec	63
5	721eb200ba06206dcbd4bce704fa654e 5955c91b4e769f3cb4a94768e98d5267	81
6	0ad9d85689f9f77bc1c5f71185e5fb14 dc60a24d137662181e45b8d3726b2920	70
7	db18a8ffa16d30d5f88b08d777ba4eaa fe8343b8f88bef66cab7e977d005a03c	74
8	f91b4fbfe934c9bf8f2f85812b084989 da7dad581d1725c5b72fa0f9d9d1366a	67
9	cca104a13e678500ff59025f3bafaa34 0ccb4c66bbfd912f4b511d72996345e0	59
10	ff0b844a0853bf7c6934ab4364148fb9 fc8923ee501a7d207ab670686839996b	53

1-bit change in key

AES IMPROVEMENT FOR IMPLEMENTATION



Equivalent

(a) Encryption

(b) Decryption

BLOCK CIPHER OPERATION

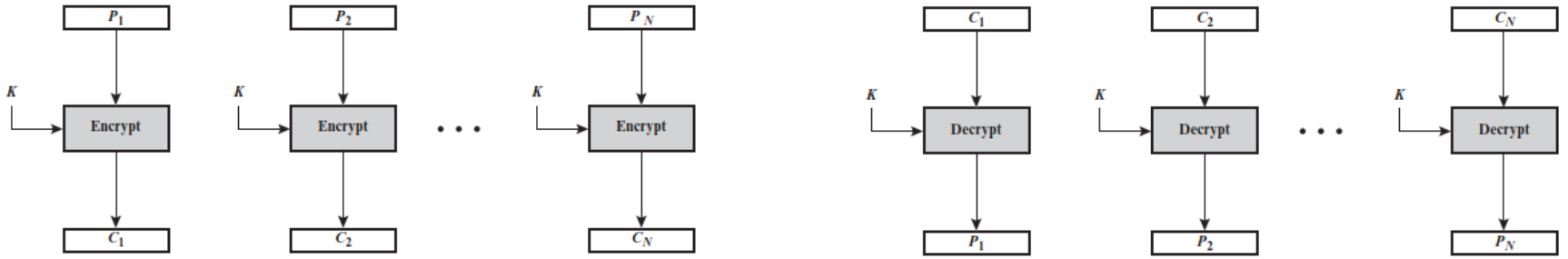
- We have discussed encryption for a single block, but the plaintext normally consists of multiblock
- There are five block cipher modes of operation:
  - Electronic Codebook (ECB)
  - Cipher Block Chaining (CBC)
  - Cipher Feedback (CFB)
  - Output Feedback (OFB)
  - Counter (CTR)

ELECTRONIC CODEBOOK (ECB)

- Message is broken into independent blocks which are encrypted
- Each block is a value which is substituted, like a codebook, hence name
- Each block is encoded independently of the other blocks

$$C_i = E_K(P_i)$$

- ECB is used for secure transmission of single block

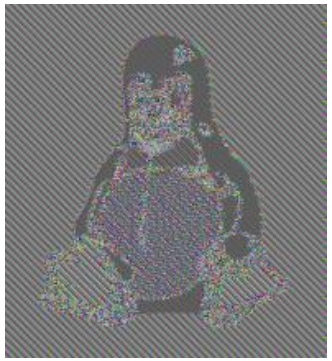


(a) Encryption (b) Decryption

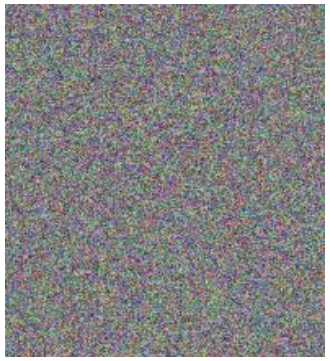
LIMITATIONS OF ECB

- Limitations
  - If the same block of plaintext appears more than once in the message, it always produces the same ciphertext.
  - Weakness due to encrypted message blocks being independent

Original



ECB



No ECB

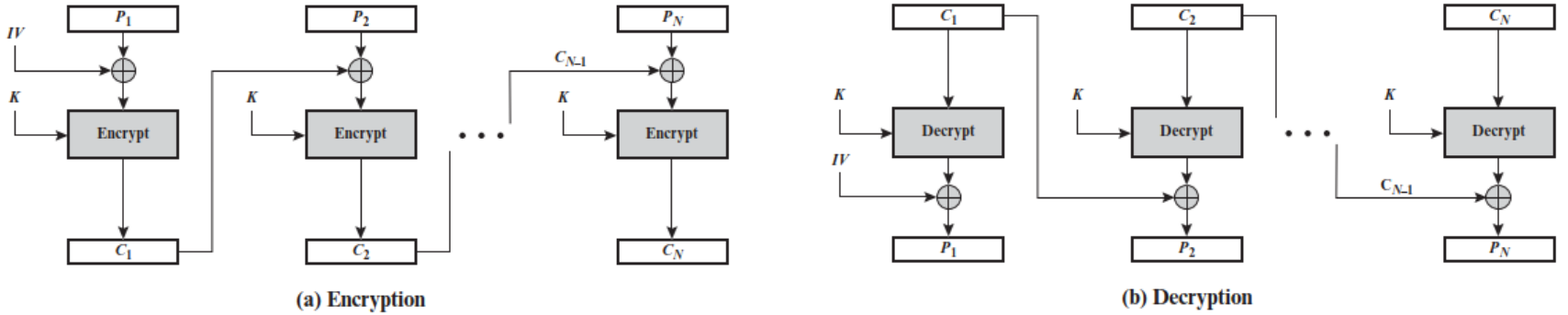
CIPHER BLOCK CHAINING (CBC)

- Message is broken into blocks
- But these are linked together in the encryption operation
- Each previous cipher blocks is chained with current plaintext block
- Use Initial Vector (IV) to start process

$$C_i = E_K(P_i \oplus C_{i-1})$$

$$C_{-1} = IV$$

- CBC is used for bulk data encryption, authentication

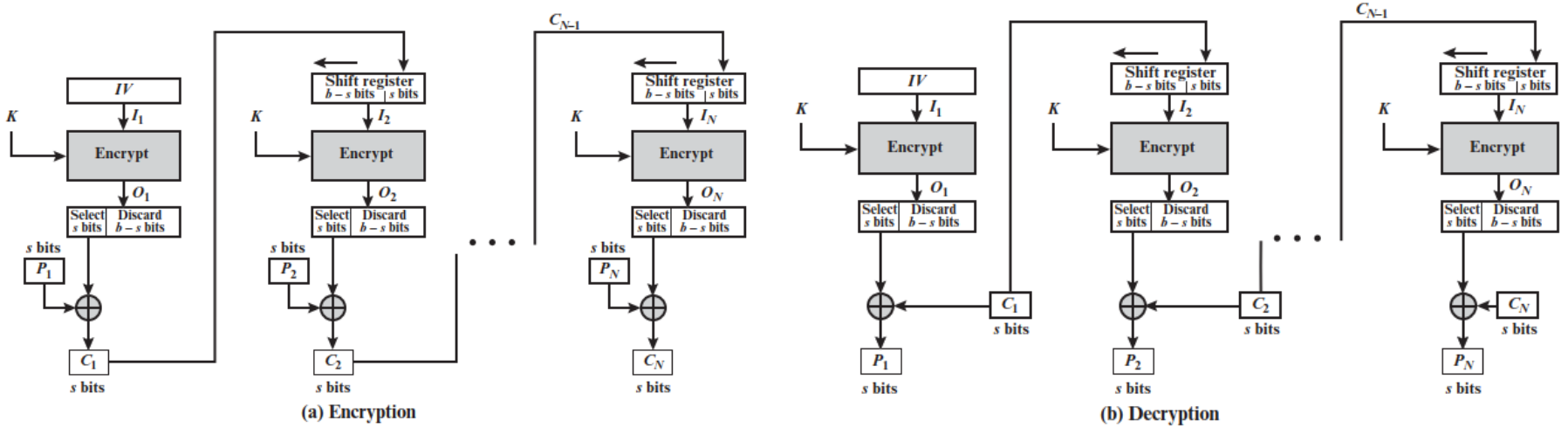


CIPHER FEEDBACK (CFB)

- Message is treated as a stream of bits
- Result is feedback for next stage
- Standard allows any number of bit (1,8 or 64 or whatever) to be fed back, namely CFB-1, CFB-8, CFB-64, etc
- A common value is s=8 (CFB-8)

$C_i = P_i \oplus S_s(E_K(C_{i-1}))$, $S_s(x)$ are the "s" bits of x

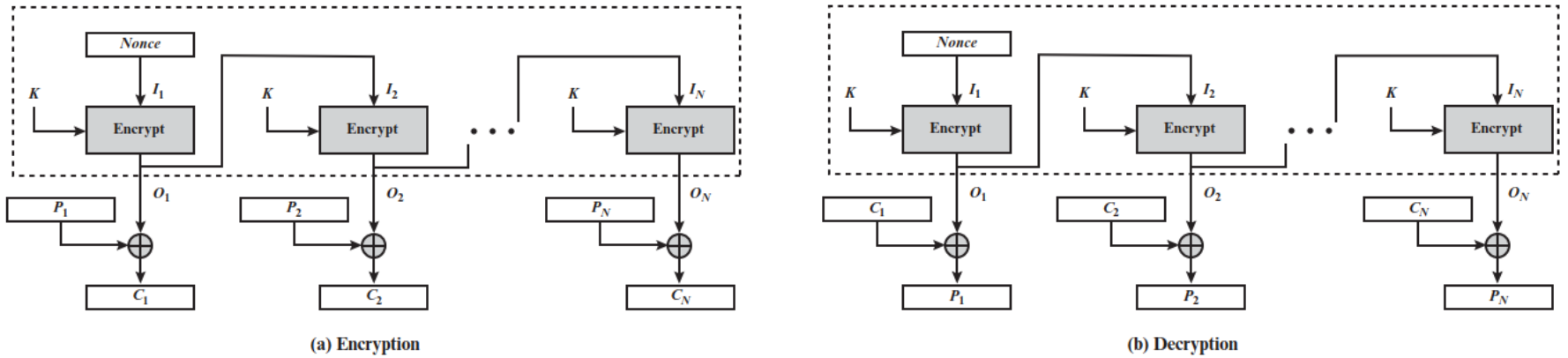
$C_{-1} = IV$



OUTPUT FEEDBACK (OFB)

- Message is treated as a stream of bits
- Output of cipher is added to message
- Output is then fed back
- Feedback is independent of message
- Can be computed in advance

$$C_i = P_i \oplus O_i \;,\; O_i = E_K(O_{i-1}),\; O_{-1} = IV$$

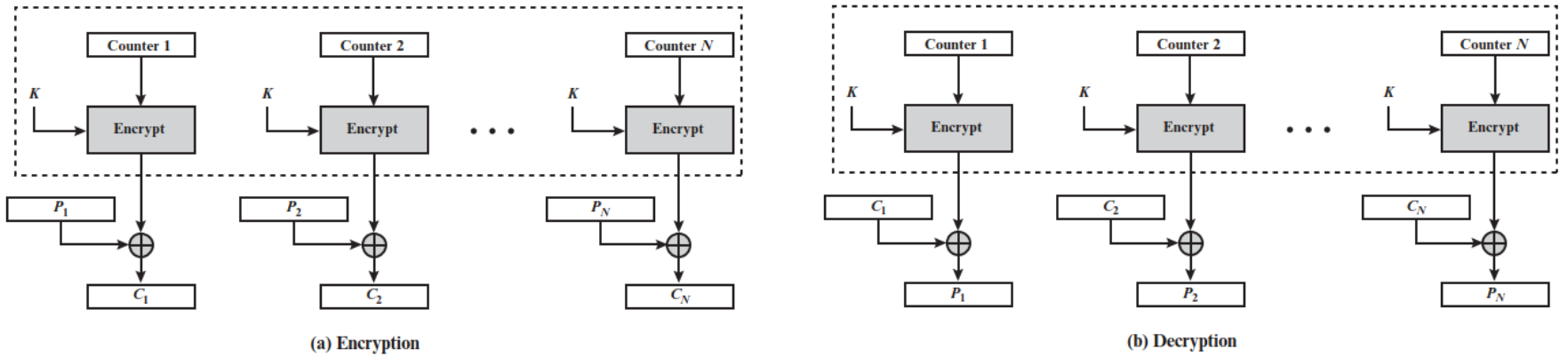


(a) Encryption (b) Decryption

COUNTER (CTR)

- A “new” mode, though proposed early on
- Similar to output feedback but encrypts counter value rather than any feedback value
- Must have a different key & counter value for every plaintext block (never reused)

$$C_i = P_i \oplus O_i \quad , \quad O_i = E_K(\text{Counter}+i-1)$$



(a) Encryption

(b) Decryption

LABORATORY

- Laboratory_01: Python Encryption AES