Lecture Outline

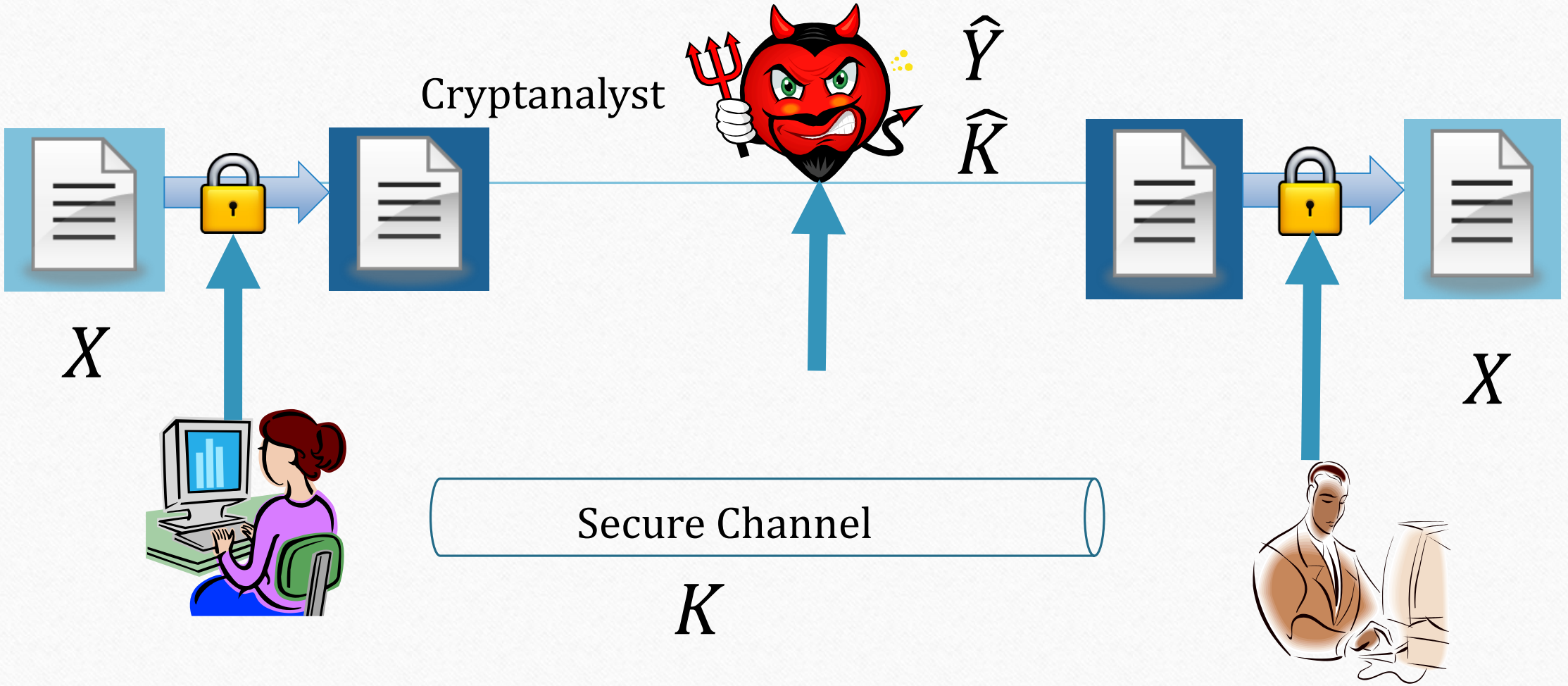
Symmetric ciphers

- Block cipher principles
- Feistel Cipher
- Data Encryption Standard (DES)
- Triple DES
- AES

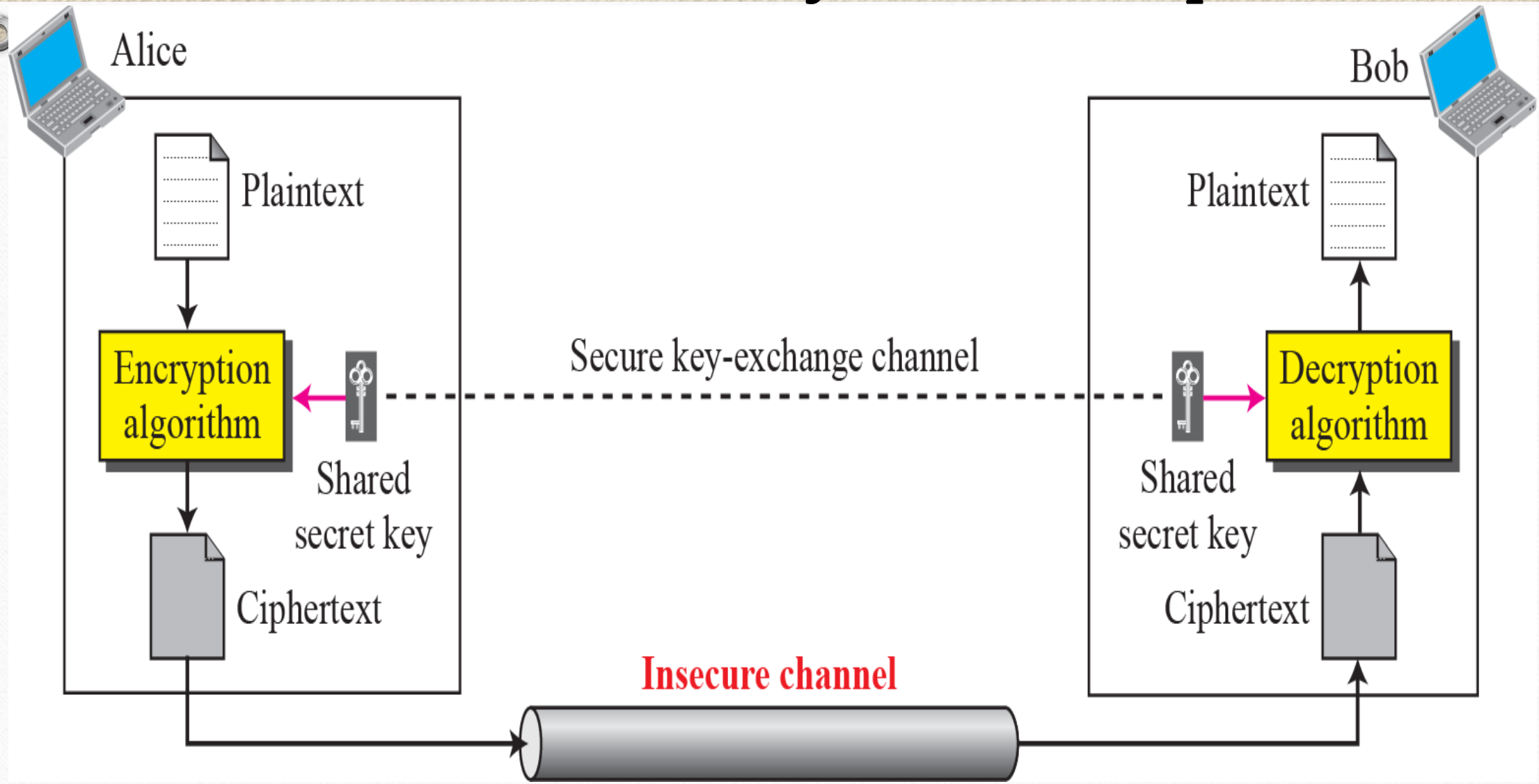
General Idea of Symmetric Ciphers

- Both sender and receiver keys are the same: $K_A = K_B$
- The keys must be kept secret and securely distributed
  - Thus, also called "Secret Key Cryptography"
- Data Encryption Standard (DES), 3DES , IDEA etc

Symmetric Cipher Model



General Idea of Symmetric Ciphers



Modern Symmetric Ciphers

- The traditional symmetric-key ciphers are character-oriented ciphers.
- Currently, bit-oriented ciphers are now more commonly used.
- This is because the information to be encrypted is not just text; it can also consist of numbers, graphics, audio, and video data.
- It is convenient to convert these types of data into a stream of bits, to encrypt the stream, and then to send the encrypted stream.
- A modern block cipher can be either a block cipher or a stream cipher.

Modern Ciphers

Stream Ciphers

- Stream ciphers, like block ciphers, break message into fixed length blocks, but use a sequence of keys to encrypt the blocks.
- The Vigenère cipher is an example of a stream cipher.
- Let E be an encipherment algorithm, and let $E_k(b)$ be the encipherment of the message b with key k.
- Let a message $m = b_1 b_2 \ldots$where each b_i is of a fixed length, and let $k = k_1 k_2 \ldots .$
- A **stream cipher** is a cipher for which $E_k(m) = Ek_1(b_1)Ek_2(b_2) \ldots$

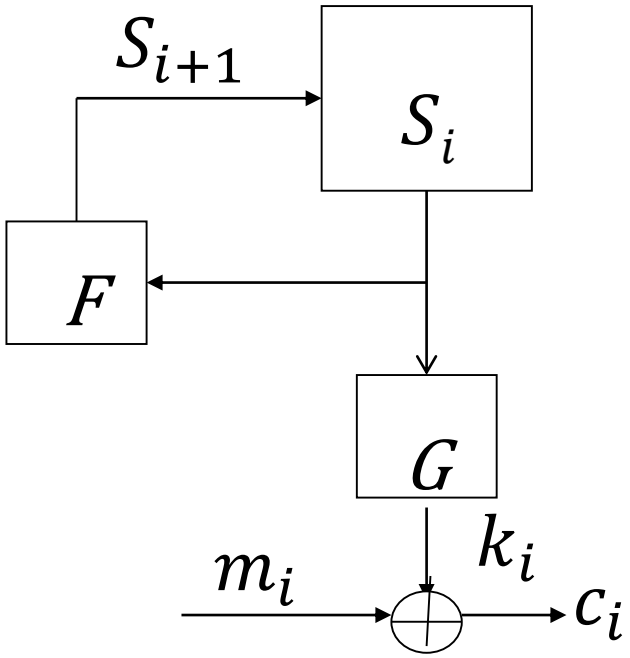
Modern Ciphers

Stream Ciphers

- Idea of a stream cipher: partition the text into small (e.g. 1 bit) blocks and let the encoding of each block depend on many previous blocks.
- For each block, a different “key” is generated.

Modern Ciphers

Stream Cipher Model



S_i : state of the cipher
at time $t = i$.

F : state function.

G : output function.

- Initial state, output and state functions are controlled by the secret key.

Stream Ciphers

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 would be used, in which the keystream is as long as the plaintext bit stream

If the cryptographic keystream is random, then this cipher is unbreakable by any means other than acquiring the keystream

- Keystream must be provided to both users in advance via some independent and secure channel
- This introduces insurmountable logistical problems if the intended data traffic is very large

Stream Ciphers

For practical reasons the bit-stream generator must be implemented as an algorithmic procedure so that the cryptographic bit stream can be produced by both users

It must be computationally impractical to predict future portions of the bit stream based on previous portions of the bit stream

The two users need only share the generating key and each can produce the keystream

Modern Ciphers

Block Ciphers

- 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 , 128, 256 bits is used. As with a stream cipher, the two users share a symmetric encryption key. A block cipher can be used to achieve the same effect as a stream cipher.
- Far more effort has gone into analyzing block ciphers.
- They are applicable to a broader range of applications than stream ciphers.
- The vast majority of network-based symmetric cryptographic applications make use of block ciphers.

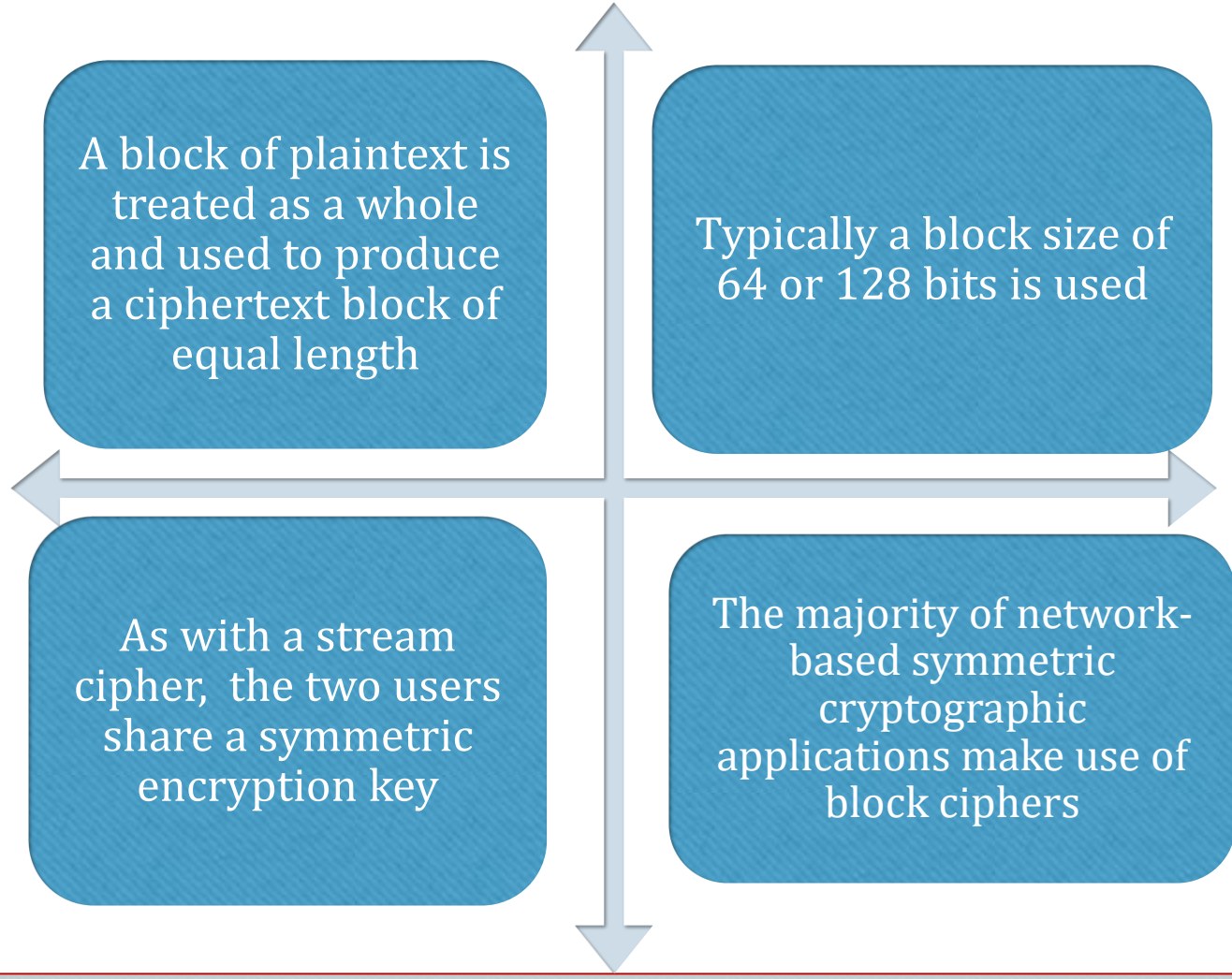
Modern Ciphers

Block Ciphers

- Idea of a block cipher: partition the text into relatively large (e.g. 128 bits) blocks and encode each block separately.
- The encoding of each block generally depends on at most one of the previous blocks.
- The same “key” is used at each block.

Modern Ciphers

Block Ciphers



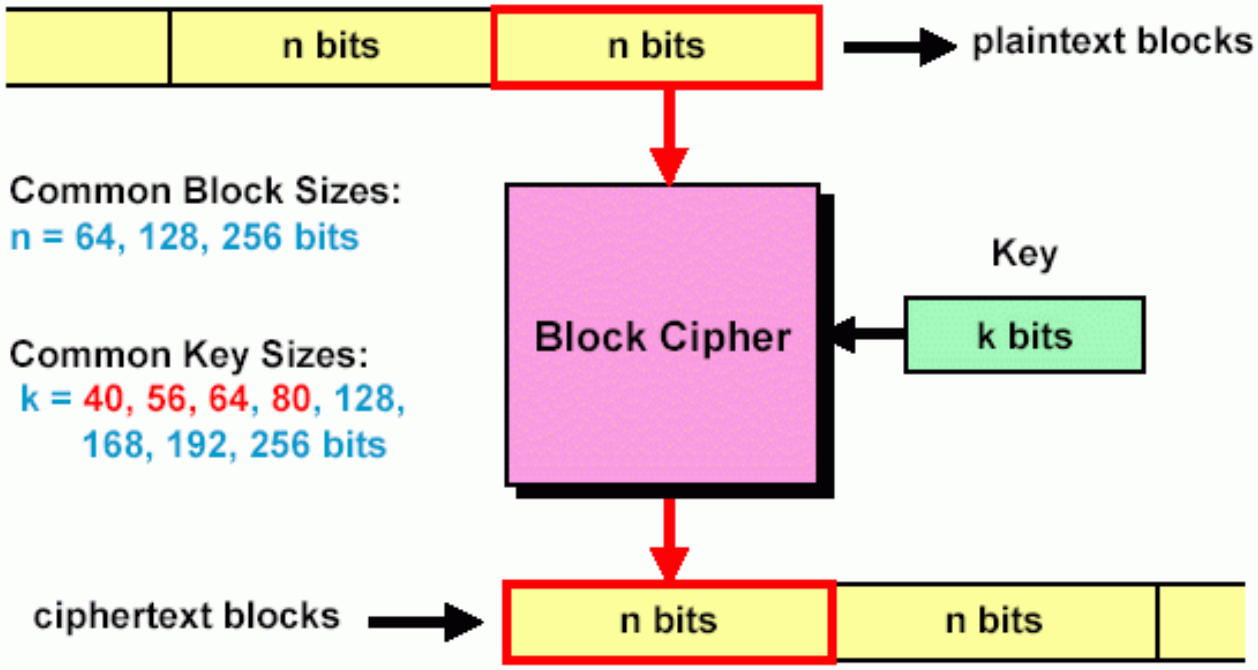
Modern Ciphers

Block Ciphers: Formal Definition

- Let E be an encipherment algorithm, and let $E_k(b)$ be the encipherment of the message b with key k.
- Let a message $m = b_1 b_2$ …where each b_i is of a fixed length.
- A **block cipher** is a cipher for which $E_k(m) = E_k(b_1)E_k(b_2)$ …

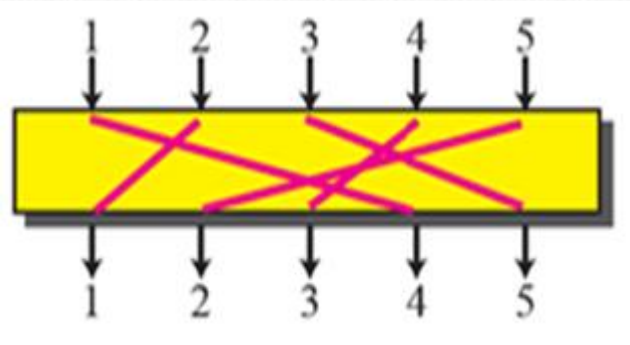
A Modern Block Cipher

Divide input bit stream into n-bit sections, encrypt only that section, no dependency/history between sections

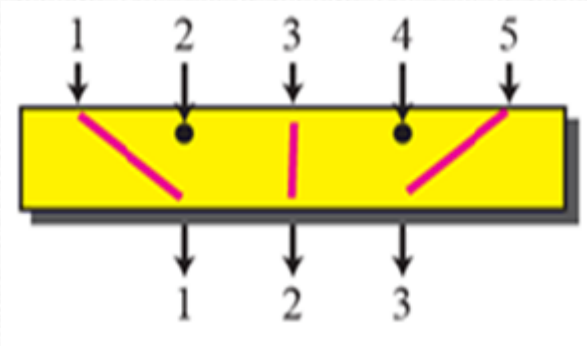


In a good block cipher, each output bit is a function of all n input bits and all k key bits

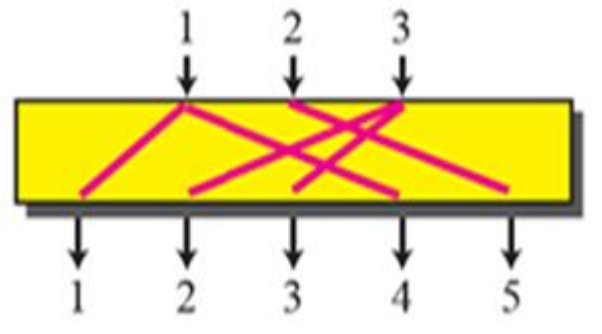
Components of a modern block cipher



Straight Permutation

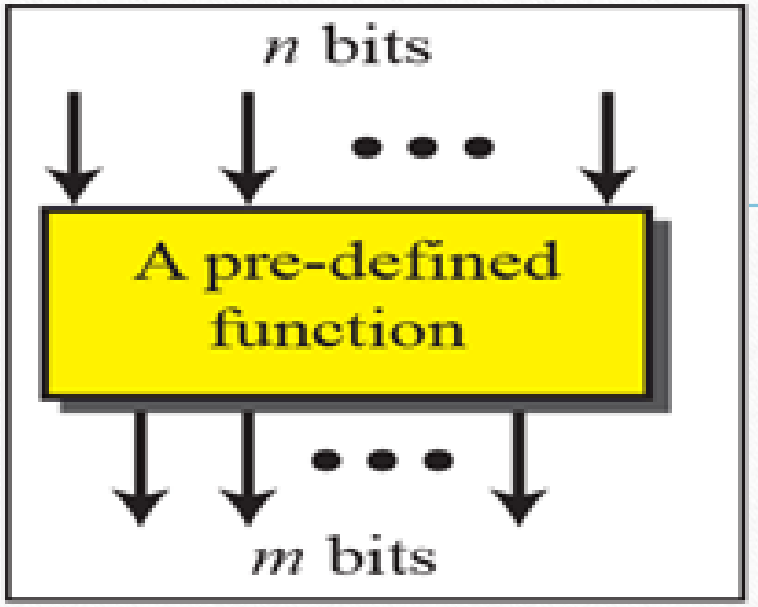


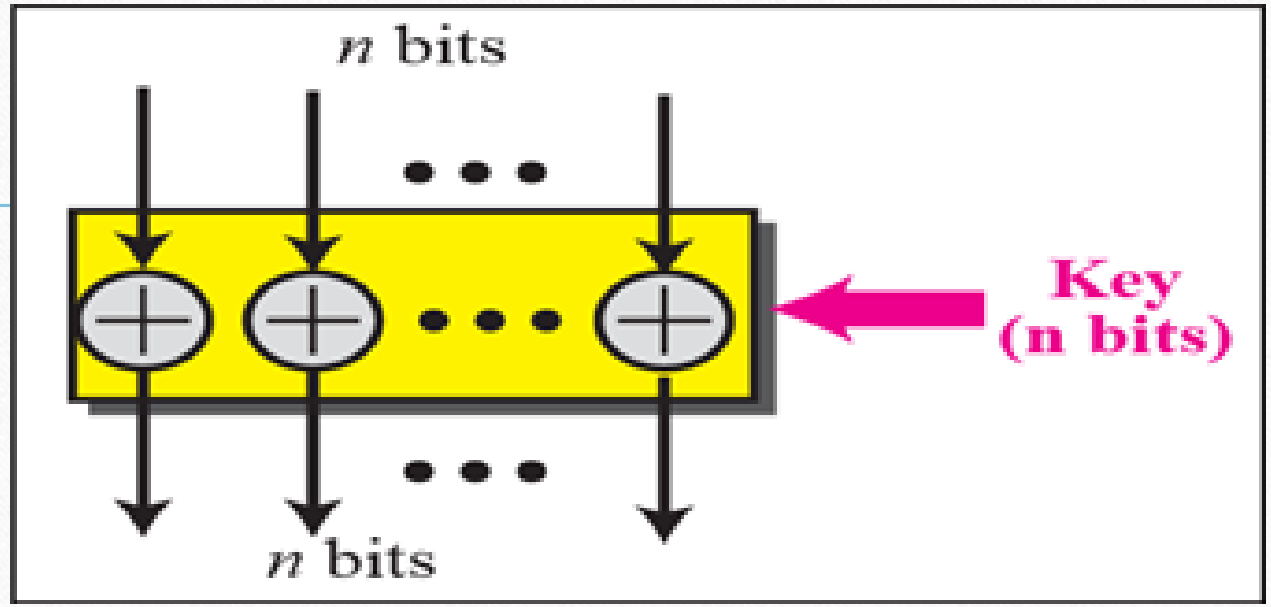
Compression Permutation



Expansion Permutation

Components of a modern block cipher

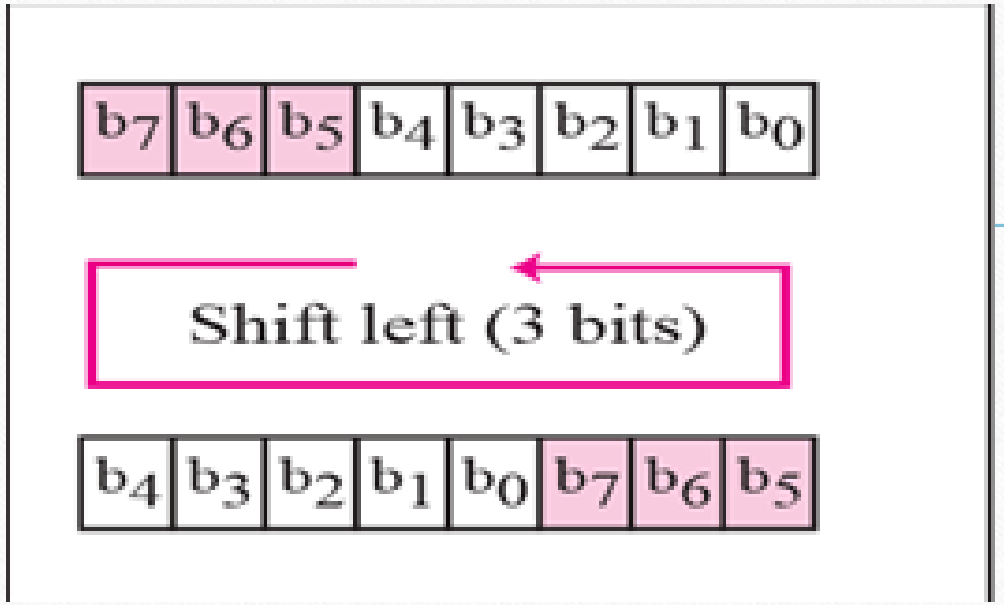




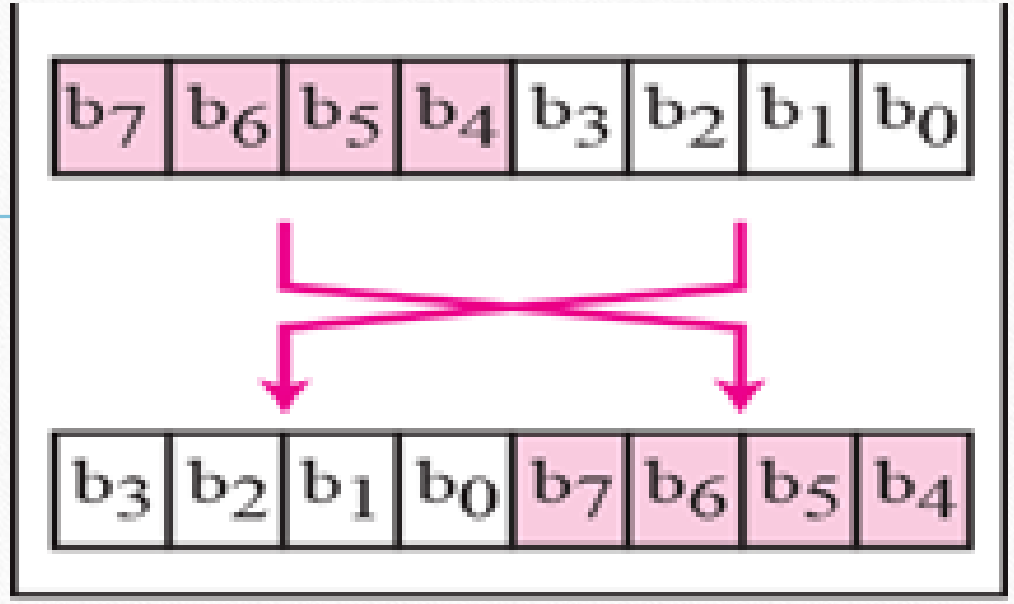
Substitution

Exclusive OR

Components of a modern block cipher

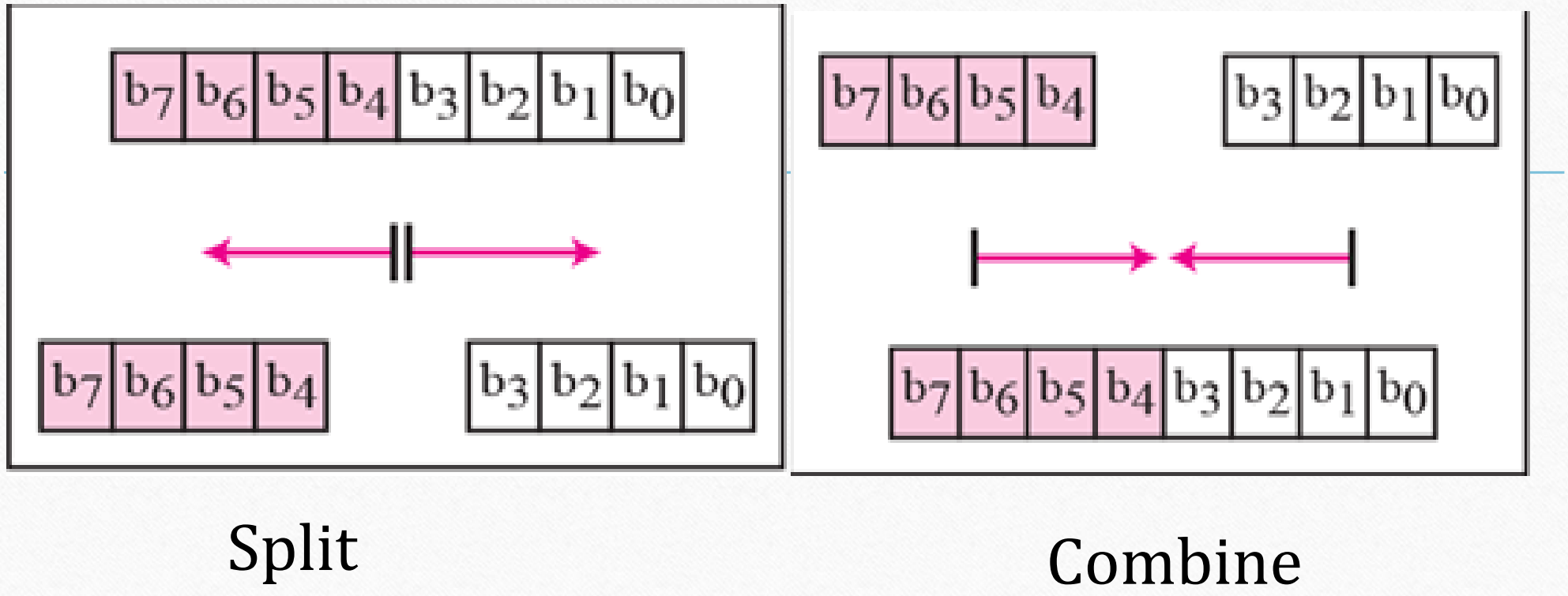


Shift



Swap

Components of a modern block cipher



Split

Combine

Lecture Outline

Symmetric ciphers

- Block cipher principles
- Feistel Cipher
- Data Encryption Standard (DES)
- Triple DES
- AES

Feistel Cipher

- Fiestal proposed the use of a cipher that alternates substitutions and permutations

Substitutions

- Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements

Permutation

- No elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence is changed

- Is a practical application of a proposal by Claude Shannon to develop a product cipher that alternates confusion and diffusion functions

- Its the structure used by some significant symmetric block ciphers currently in use

Feistel Cipher

- Terms introduced by Claude Shannon to capture the two basic building blocks for any cryptographic system
  - Shannon's concern was to thwart cryptanalysis based on statistical analysis

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

Confusion

- Seeks to make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible
- Even if the attacker can get some handle on the statistics of the ciphertext, the way in which the key was used to produce that ciphertext is so complex as to make it difficult to deduce the key

Feistel Cipher

Feistel Cipher Design Features

- The exact realization of a Feistel network depends on the choice of certain parameters and design features:

1. Block size: Larger block sizes mean greater security but reduce encryption/decryption speed for a given algorithm. 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.
2. Key size: Larger key size mean greater security but may decrease encryption/decryption speed. 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.

Feistel Cipher

Feistel Cipher Design Features

- The exact realization of a Feistel network depends on the choice of the following parameters and design features:

3. 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.
4. Subkey generation algorithm: Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.
5. Round function F: Again, greater complexity generally means greater resistance to cryptanalysis.

Feistel Cipher

Feistel Cipher Design Features

There are two other considerations in the design of a Feistel cipher:

i. Fast software encryption/decryption: In many cases, encryption is embedded in applications or utility functions. Accordingly, the speed of execution of the algorithm becomes a concern.

ii. 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

Lecture Outline

Symmetric ciphers

- Block cipher principles
- Feistel Cipher
- Data Encryption Standard (DES)
- Triple DES
- AES
- RC4

Data Encryption Standard (DES)

- DES is a block cipher; it encrypts data in 64-bit blocks.
- The fundamental building block of DES is called a round and it (DES) has 16 rounds
- The same algorithm and key are used for encryption and decryption (except for minor differences in key schedule).

$$\mathcal{M} = \mathrm{C} = \{0,1\}^{64}$$

- The key length is 56 bits. $k = \{0,1\}^{56}$

(The key is usually expressed as a 64-bit number, but every eight bit is used for parity checking and ignored.)

- A handful of numbers are considered weak keys, e.g.,

$$E_k\big(E_k(\mathcal{M})\big) = \mathcal{M}$$

$$E_{k_1}(E_{k_2}(\mathcal{M}) = \mathcal{M}$$

Data Encryption Standard (DES)

- Encodes plaintext in 64-bit chunks using a 64-bit key (56 bits + 8 bits parity)
- Uses a combination of diffusion and confusion to achieve security
- Was cracked in 1997
  - Parallel attack – exhaustively search key space
- Decryption in DES – it's symmetric! Use K_A again as input and then the same keys except in reverse order

Data Encryption Standard (DES)

1. 64-bit input is permuted
2. 16 stages of identical operation
   - Differ in the 48-bit key extracted from 56-bit key - complex
   - $R_I = R_{I-1}$ encrypted with K_I and XOR'd with L_{I-1}
3. Final inverse permutation stage

Data Encryption Standard (DES)

1. First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the *permuted input.*

2. This is followed by a phase consisting of sixteen rounds of the same function, which involves both permutation and substitution functions.

3. The output of the last (sixteenth) round consists of 64 bits that are a function of the input plaintext and the key. The left and right halves of the output are swapped to produce the pre-output.

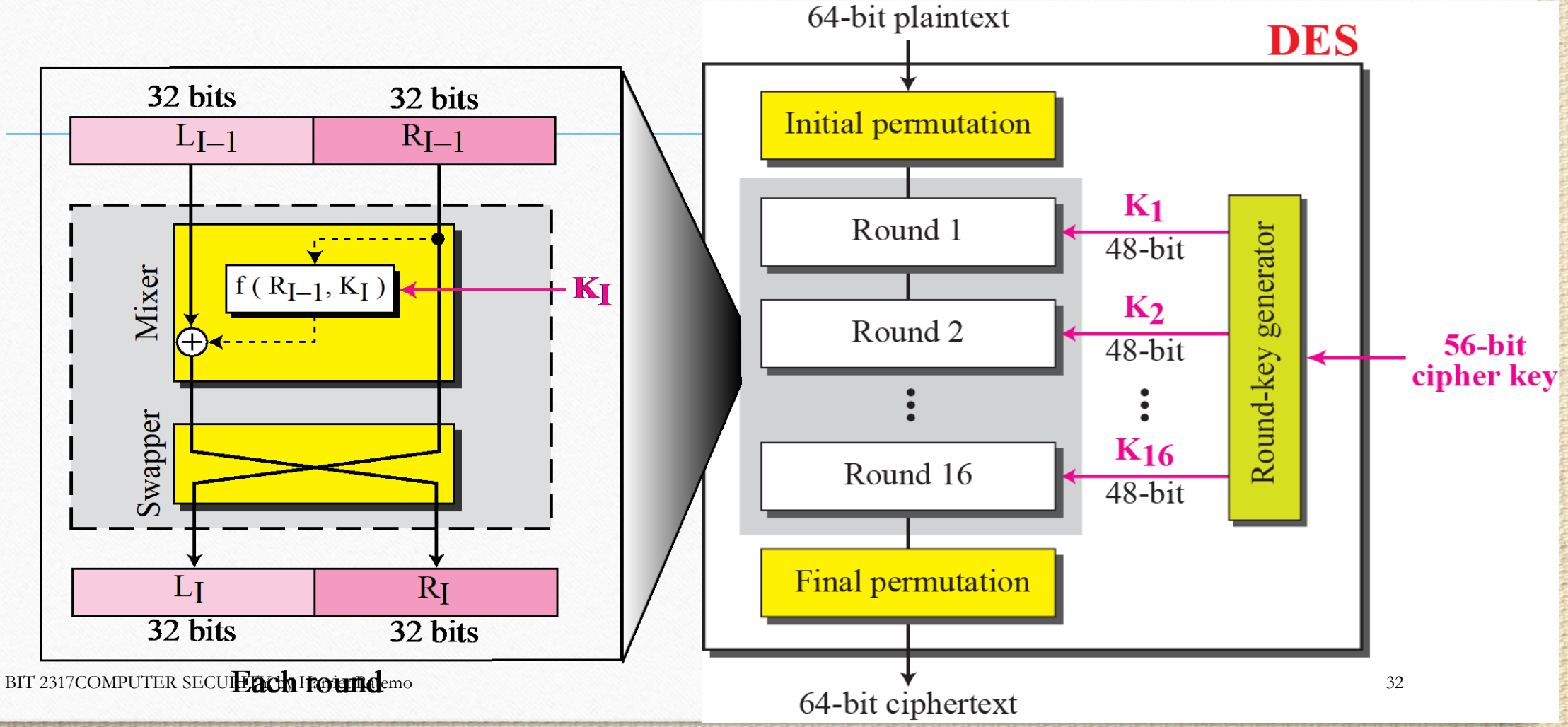
Data Encryption Standard (DES)

4. Finally, the pre-output is passed through a permutation that is the inverse of the initial permutation function, to produce the 64-bit ciphertext.

5. With the exception of the initial and final permutations, DES has the exact structure of a Feistel cipher.

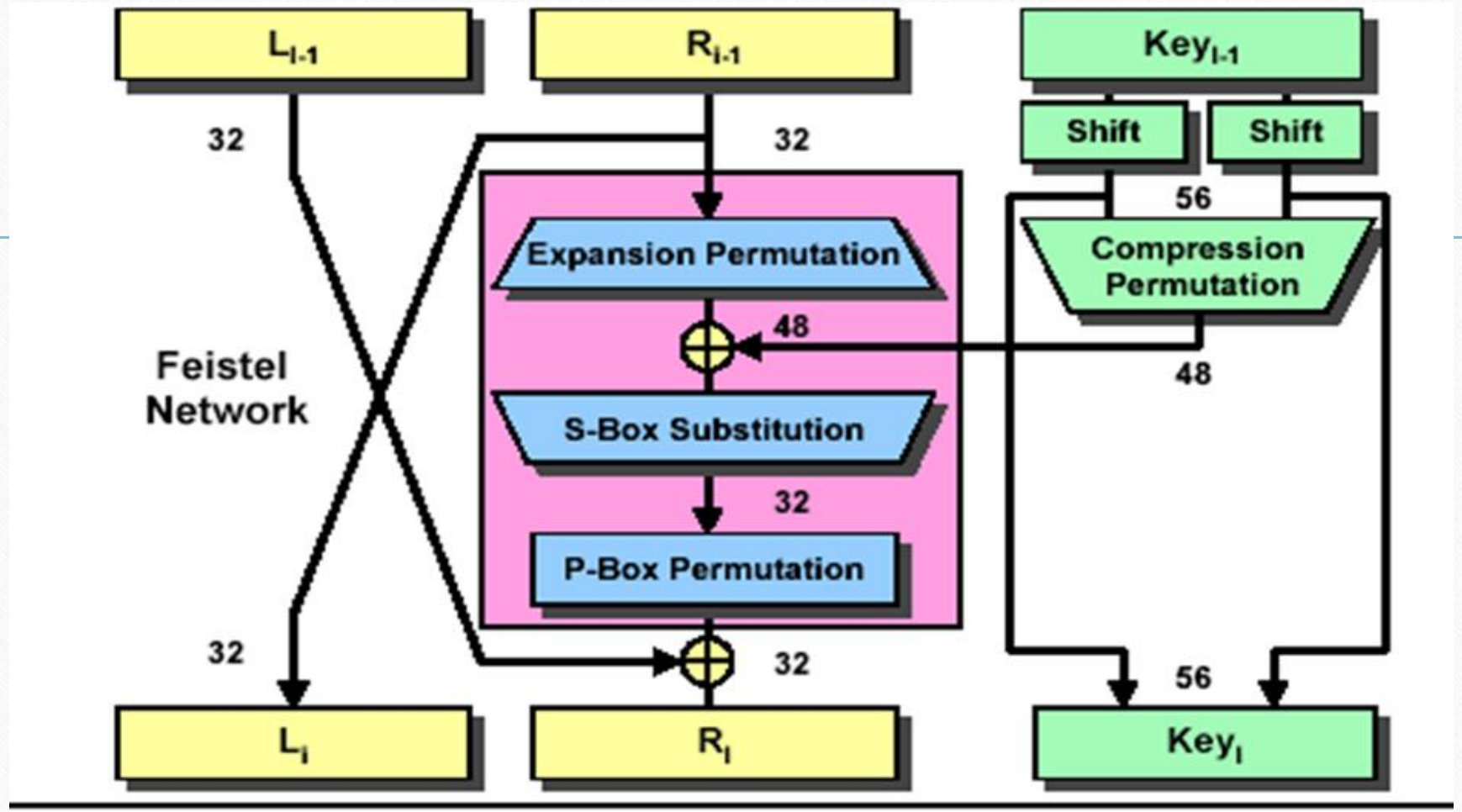
DES Inner Working

General Structure of DES



Data Encryption Standard (DES)

General Structure of DES



Data Encryption Standard (DES)

Feistel Network

- The network has the F-functions as the main component

The F-functions' (Feistel Function) basic operation

- It operates on half a block (32-bit)at a time and consists of 4 stages:-

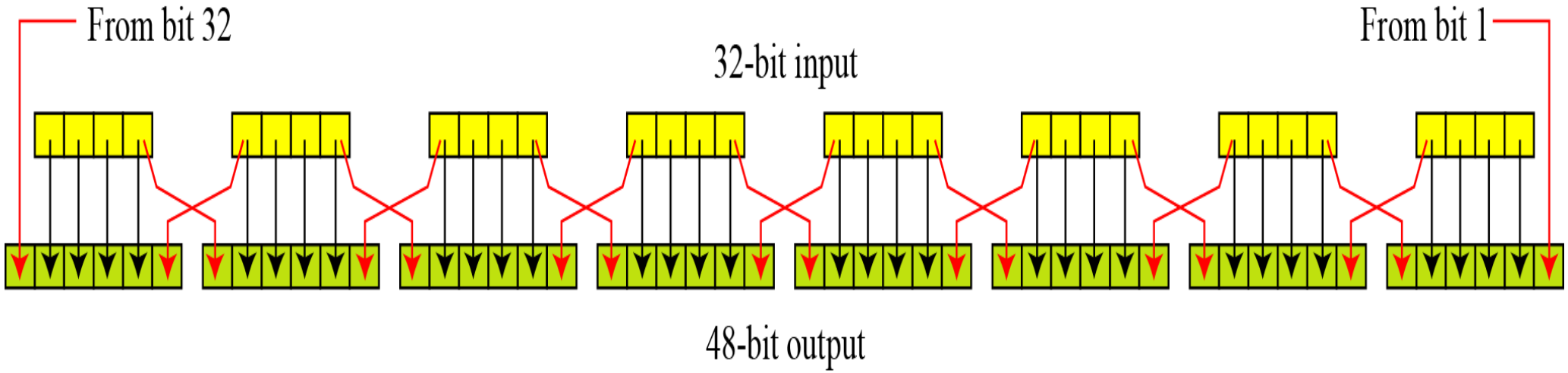
1. Expansion: The 32-bit block is expanded to 48 bits using the expansion permutation, by duplicating half of the bits. The o/p consists of eight 6-bit (8*6 = 48-bits) pieces, each containing a copy of 4 corresponding input bits, plus a copy of the immediately adjacent bit from each of the input pieces to either side.

- The security offered by this operation comes from one bit affecting two substitutions in the S-boxes.
- This causes the dependency of the output bits on the input bits to spread faster, creating what is called the avalanche affect.

Data Encryption Standard (DES)

Feistel Network

- Expansion permutation



Data Encryption Standard (DES)

Feistel Network

- The network has the F-functions as the main component

The F-functions' (Feistel Function) basic operation

- It operates on half a block (32-bit)at a time and consists of 4 stages:-

2. Key Mixing: The result is combined with a subkey using an XOR operation. Sixteen 48-bits subkeys, one for each round, are derived from the main key using the key schedule.

Data Encryption Standard (DES)

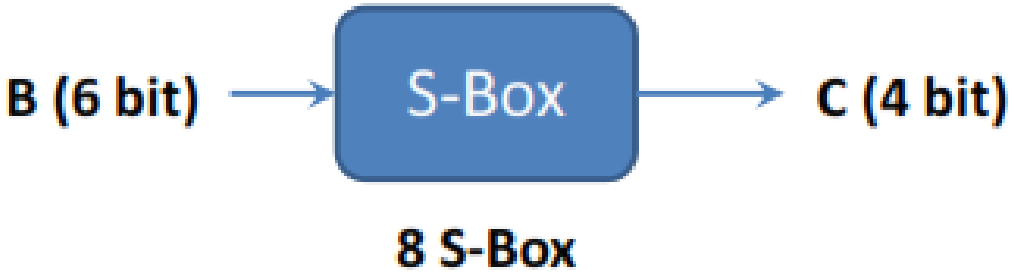
- Feistel Network

3. Substitution: The above block (48-bits) is then divided to eight 6-bit pieces before processing by the S-boxes (substitution boxes).

- Each of the eight S-boxes replaces its 6 input bits with 4 output bits according to a non linear transformation, provided in the form of a lookup table.
- The S-boxes provide the core of the security of DES, without them the cipher would be linear and trivially breakable.

Data Encryption Standard (DES)

- Feistel Network



S= matrix 4×16, values form 0 to 15

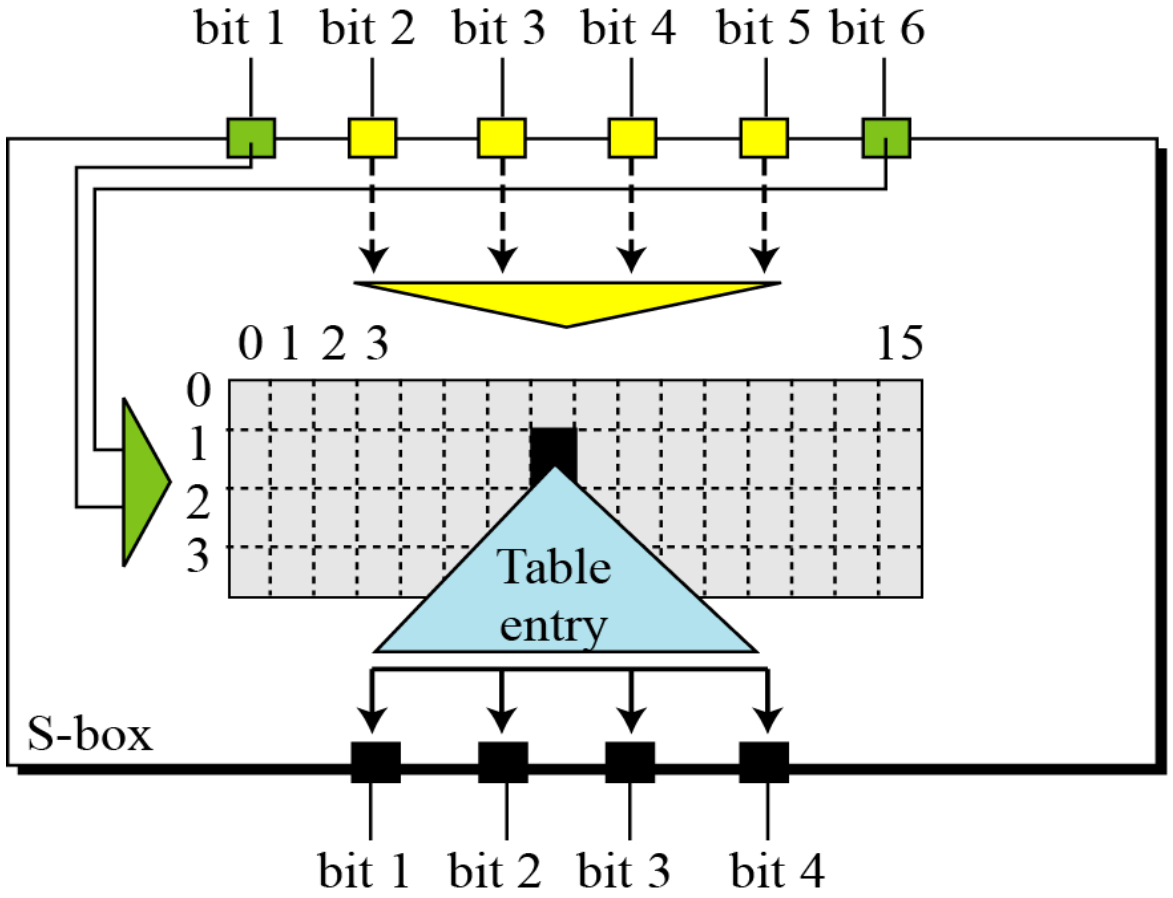
B(6 bit long)= $b_1 b_2 b_3 b_4 b_5 b_6$

$b_1 b_6 \implies r$ =row of the matrix (2 bits: 0,1,2,3)

$b_2 b_3 b_4 b_5 \implies c$= column of the matrix (4 bits: 0,1,2,3,…,15)

C(4 bit long)= Binary representation of S(r,c)

Data Encryption Standard (DES)



Data Encryption Standard (DES)

- Example

Row #	S_1	1	2	3	...	7							15	Column #
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S(i, j) <16, can be represented with 4 bits

Example 1: B=101111

$b_1 b_6 = 11$ =row 3

$b_2 b_3 b_4 b_5$= 0111=column 7

C(4 bit long)= Binary representation of S(r,c)=0111

Example 2: B=011011 ?

Example 3: B= 101011? 9 in binary this is 1001

Data Encryption Standard (DES)

Feistel Network

S_5	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

Data Encryption Standard (DES)

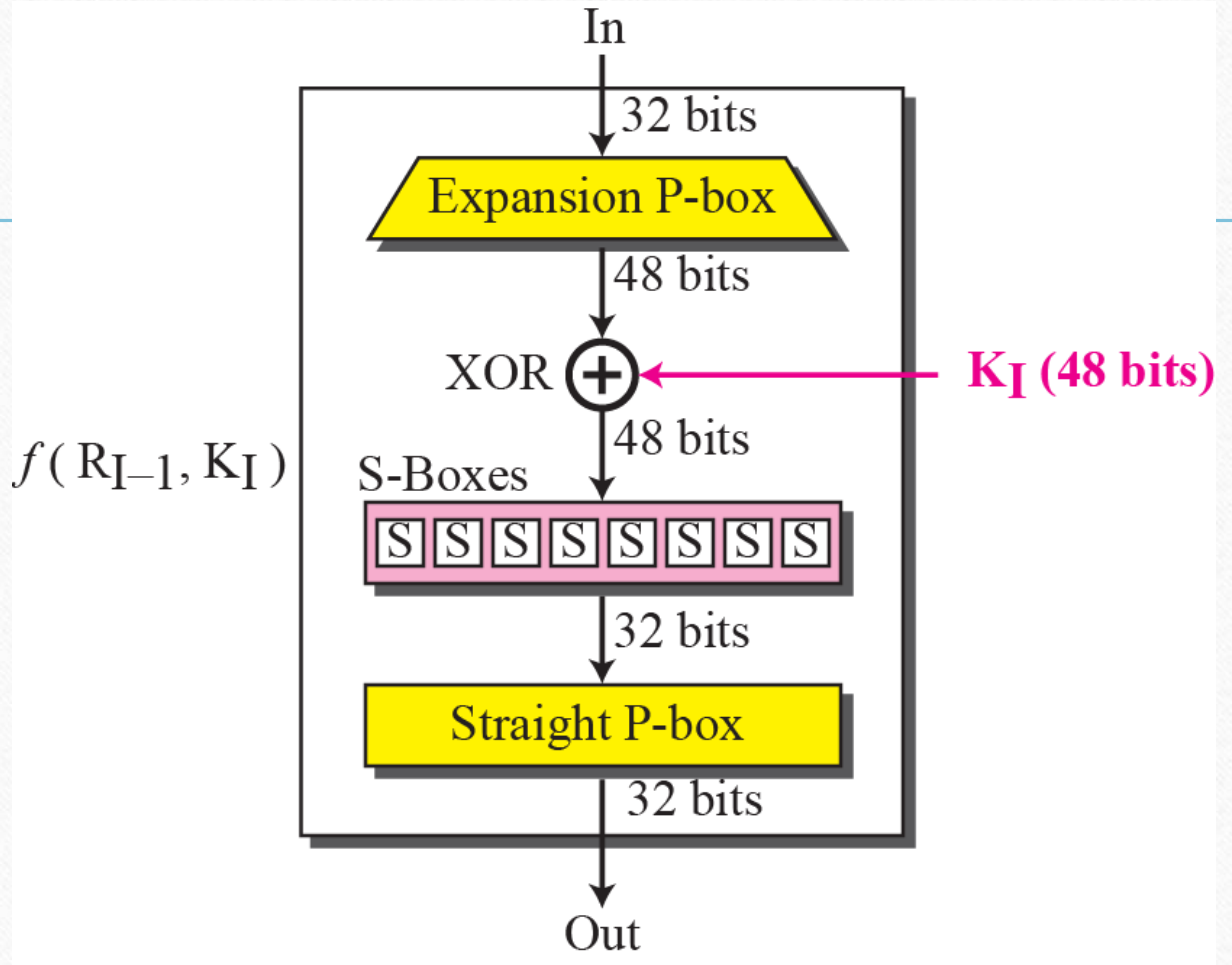
- Feistel Network

4. Permutation: Finally, the 32 outputs from the S-boxes are rearranged according to a fixed permutation, the P-box.

- This is designed so that after expansion, each S-box's output bits are spread across 6 different S-box in the next round.

Data Encryption Standard (DES)

DES Function

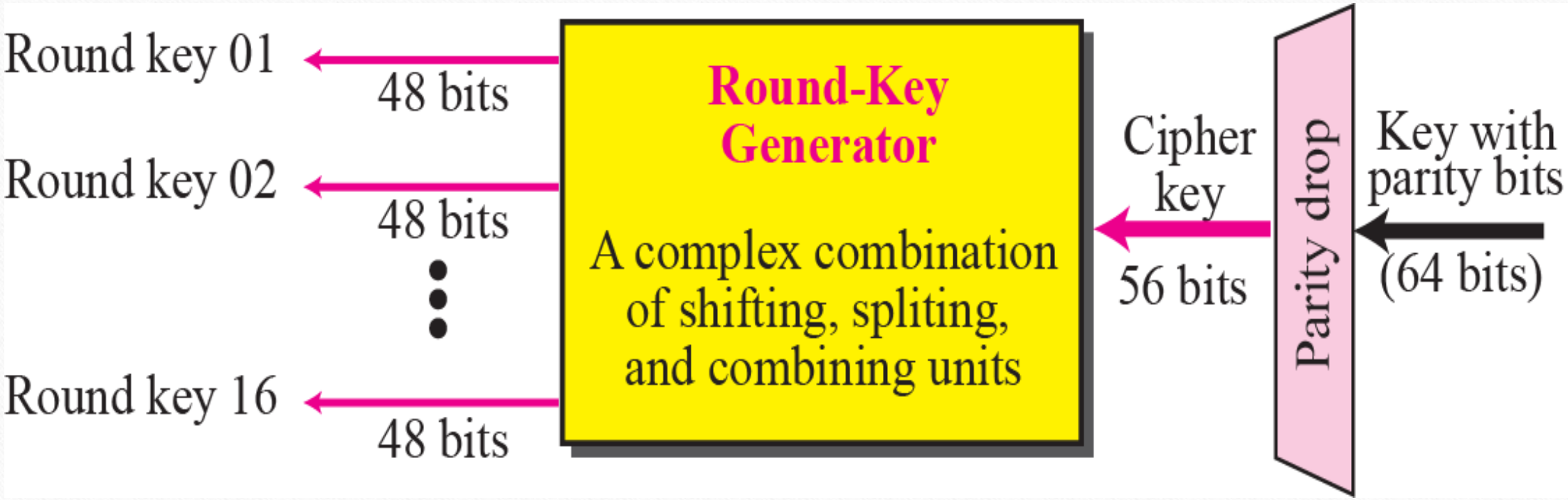


Data Encryption Standard (DES)

Key Generation

- How the 56-bit key is used.
- Initially, the key is passed through a permutation function.
- Then, for each of the sixteen rounds, a *subkey* (K_i) *is produced by the combination of a left* circular shift and a permutation.
- The permutation function is the same for each round, but a different subkey is produced because of the repeated shifts of the key bits.

Data Encryption Standard (DES)

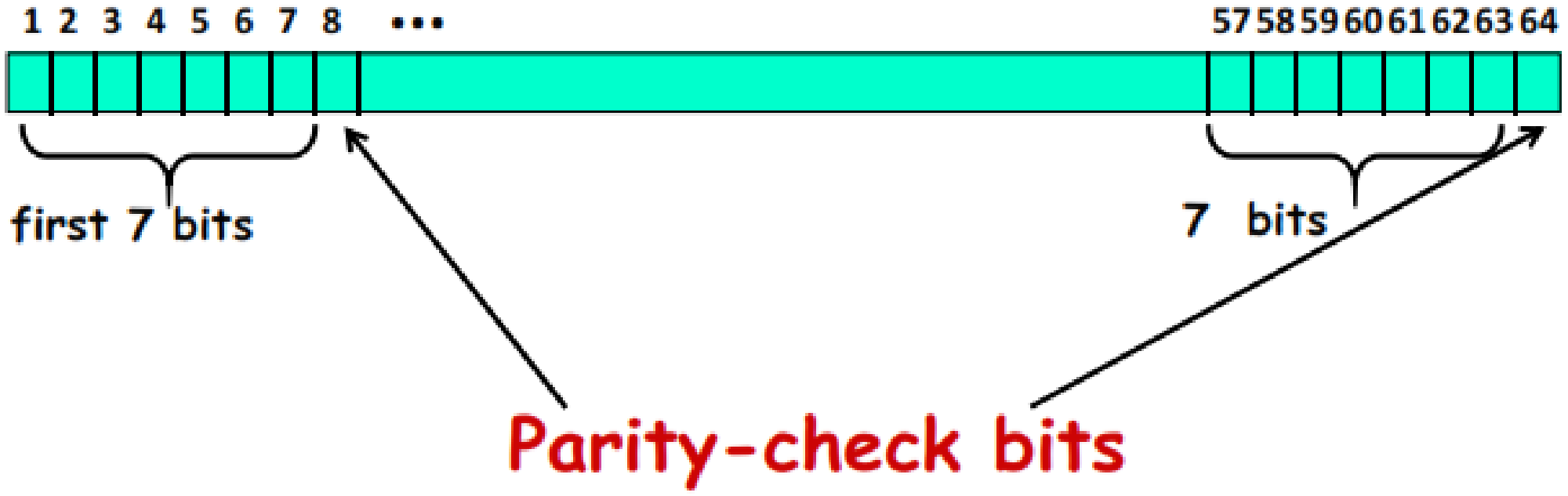


Key Generation

Key Generation

In the DES specification, the key length is 64 bit:

- 8 bytes; in each byte, the 8th bit is a parity-check bit



Each parity-check bit is the XOR of the previous 7 bits

DES Decryption

The decryption process with DES is essentially the same as the encryption process and is as follows:

- Use the ciphertext as the input to the DES algorithm but use the keys K_i in reverse order. That is, use K_{16} on the first iteration, K_{15} on the second until K_1 which is used on the 16th and last iteration.

Data Encryption Standard (DES)

Summary

- The basic operation is as follows:
- Split the plaintext block into two equal pieces, (L_0, R_0)

1. Apply a fixed Initial permutation "IP" to the input block

$$(L_0, R_0) \leftarrow IP(Input\ block)$$

2. Iterate the following 16 rounds

$$L_i \leftarrow R_{i-1}$$

$$\mathrm{R_i} \leftarrow L_{i-1} \oplus f(R_{i-1}, k_i)$$

3. The results from round 16 (L_{16}, R_{16}) is input into the inverse of IP to cancel the effects of the IP. The o/p from this step is the o/p of the DES algorithm i.e.

$$Output\ block \leftarrow IP^{-1}(L_{16}, R_{16})$$

Avalanche Effect

- Desirable property of any encryption algorithm is that a small change in either plaintext or key should produce significant changes in the ciphertext.
- DES exhibits a strong avalanche effect.

Avalanche Effect

(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

Lecture Outline

Symmetric ciphers

- Block cipher principles
- Feistel Cipher
- Data Encryption Standard (DES)
- Triple DES
- AES

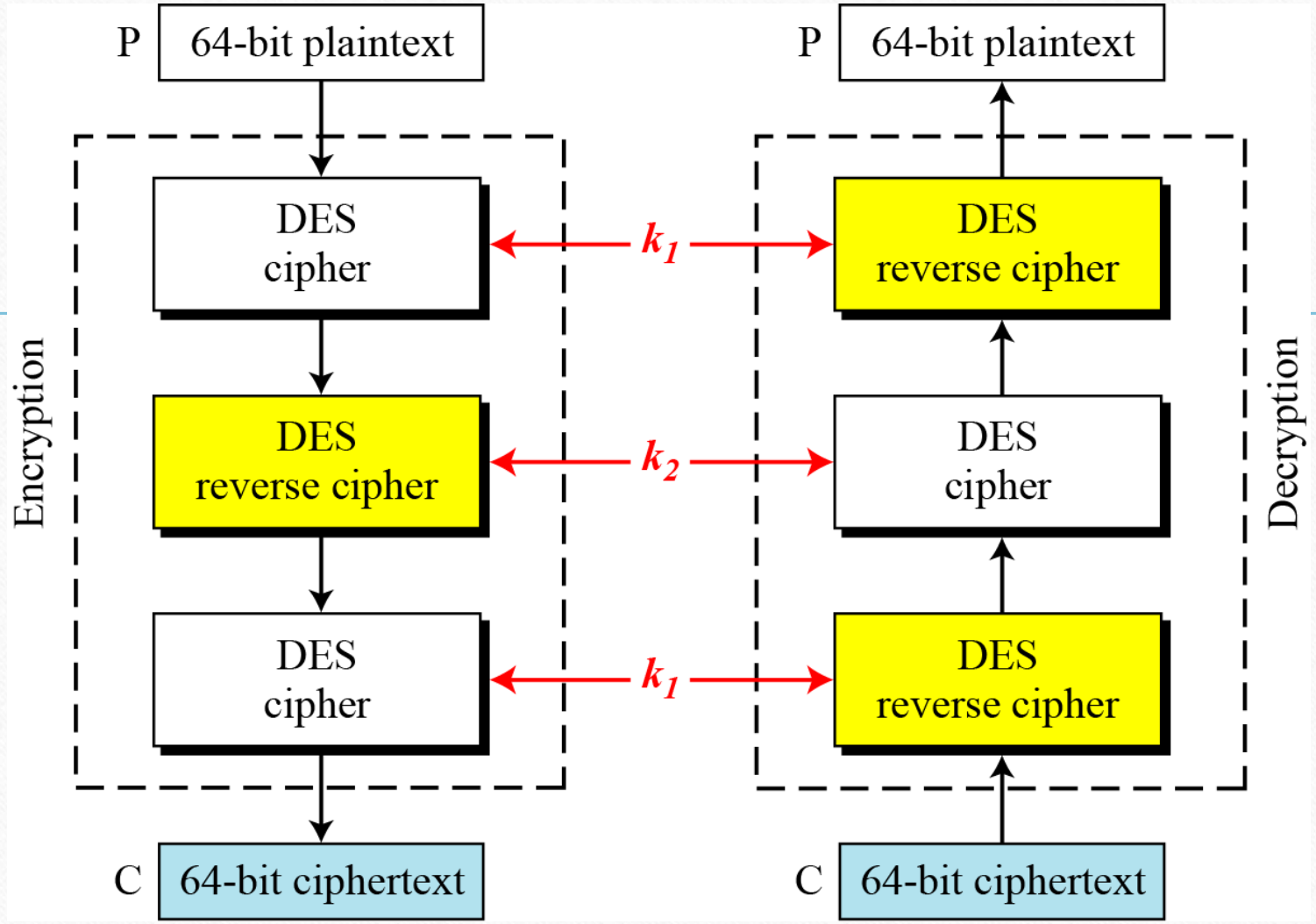
Triple DES

- The main disadvantage of DES is that it has a short key making it more prone to brute force attack.
- A soln. to overcoming this limitation is to run the DES algorithm a multiple number of times using different keys in a proposal called triple DES (encryption –decryption-encryption)

$$C \leftarrow E_{k_1}(D_{k_2}\left(E_{k_1}(m)\right))$$

$$m \leftarrow D_{k_1}\left(E_{k_2}\left(D_{k_1}(c)\right)\right)$$

Triple DES


P
64-bit plaintext
DES cipher
DES reverse cipher
DES cipher
C
64-bit ciphertext
Encryption
k_1
k_2
k_1
P
64-bit plaintext
DES reverse cipher
DES cipher
DES reverse cipher
C
64-bit ciphertext
Decryption

Triple DES

Result:

- Achieves the effect of enlarging the key space.
- The scheme also achieves easy compatibility with single DES if $k_1 = k_2$ is used.
- It can also use three different keys but this way is not compatible with single key DES.

Lecture Outline

Symmetric ciphers

- Block cipher principles
- Feistel Cipher
- Data Encryption Standard (DES)
- Triple DES
- AES

Advanced Encryption Scheme (AES)

- Published by NIST in Nov 2001: FIPS PUB 197
- Based on a competition won by Rijmen and Daemen (Rijndael) from Belgium
- 22 submissions, 7 did not satisfy all requirements
- 15 submissions, 5 finalists: Mars, RC6, Rijndael, Serpent, Twofish. Winner: Rijndael.

Advanced Encryption Scheme (AES)

- Rijndael allows many block sizes and key sizes
- AES restricts it to:
    - Block Size: 128 bits
    - Key sizes: 128, 192, 256 (AES-128, AES-192, AES-256)
- An iterative rather than Feistel cipher
    - Operates on entire data block in every round
- Byte operations: Easy to implement in software

Advanced Encryption Scheme (AES)

- The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data.
- The AES algorithm is a symmetric block cipher that can encrypt and decrypt information and is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits.
- AES was introduced to replace the Triple DES (3DES) algorithm used for a good amount of time universally.
  - Though, if security were the only consideration, then 3DES would be an appropriate choice for a standardized encryption algorithm for decades to come.

Advanced Encryption Scheme (AES)

- The main drawback (3DES) was its slow software implementation .
- For reasons of both efficiency and security, a larger block size is desirable.
- Due to its

1. High level security
2. Speed
3. Ease of implementation
4. Flexibility

Rijndael was chosen for AES standard in the year 2001.

Advanced Encryption Scheme (AES)

- Two principles of a good cipher, as defined by Claude Shannon are:

1. 'Confusion' which stands for Substitution operations
2. 'Diffusion' which stands for transposition or permutation operations

S-P Network Model (Shannon)

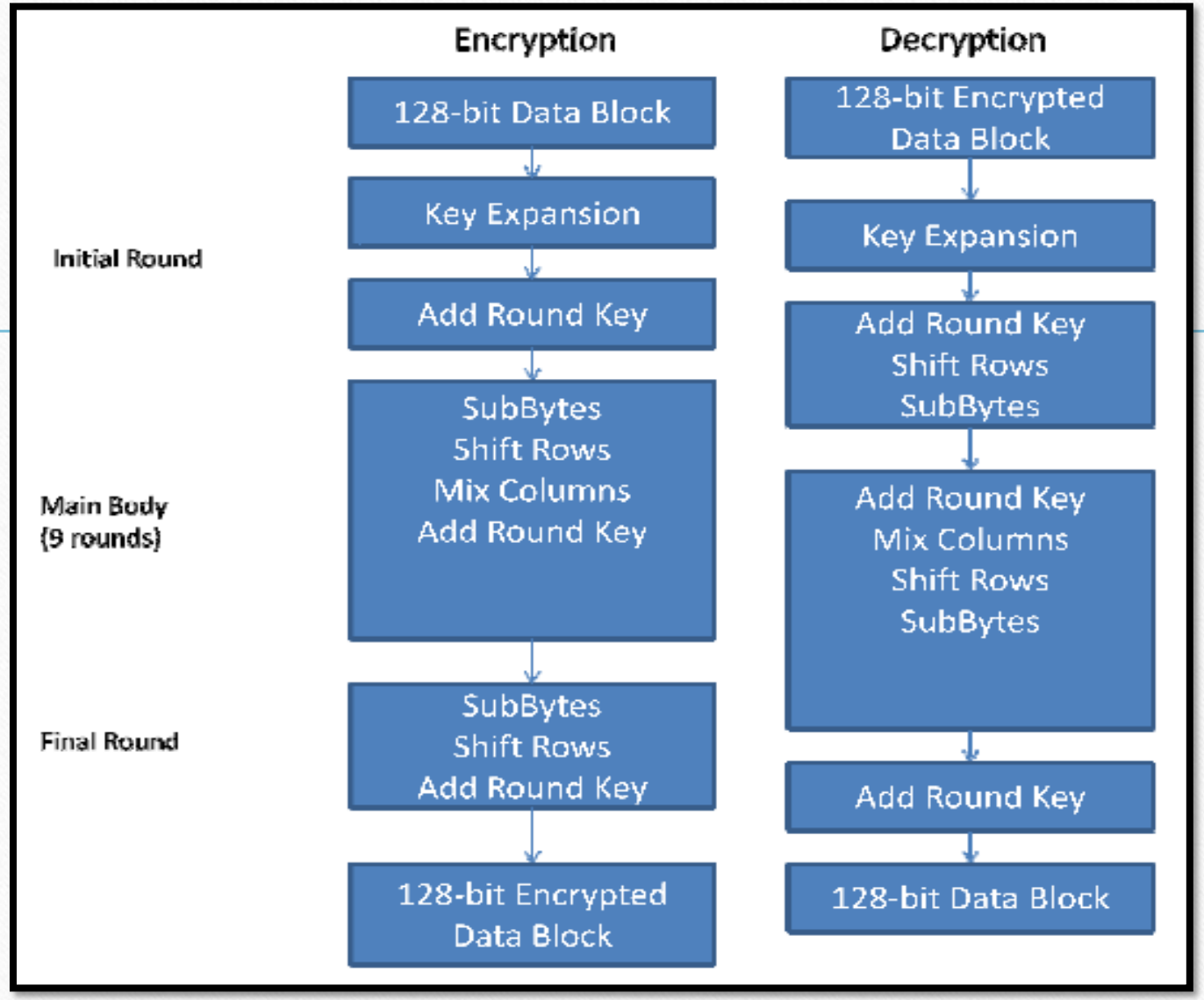
- Divide each block of data into smaller manageable pieces of same length
- In parallel, each block goes through:

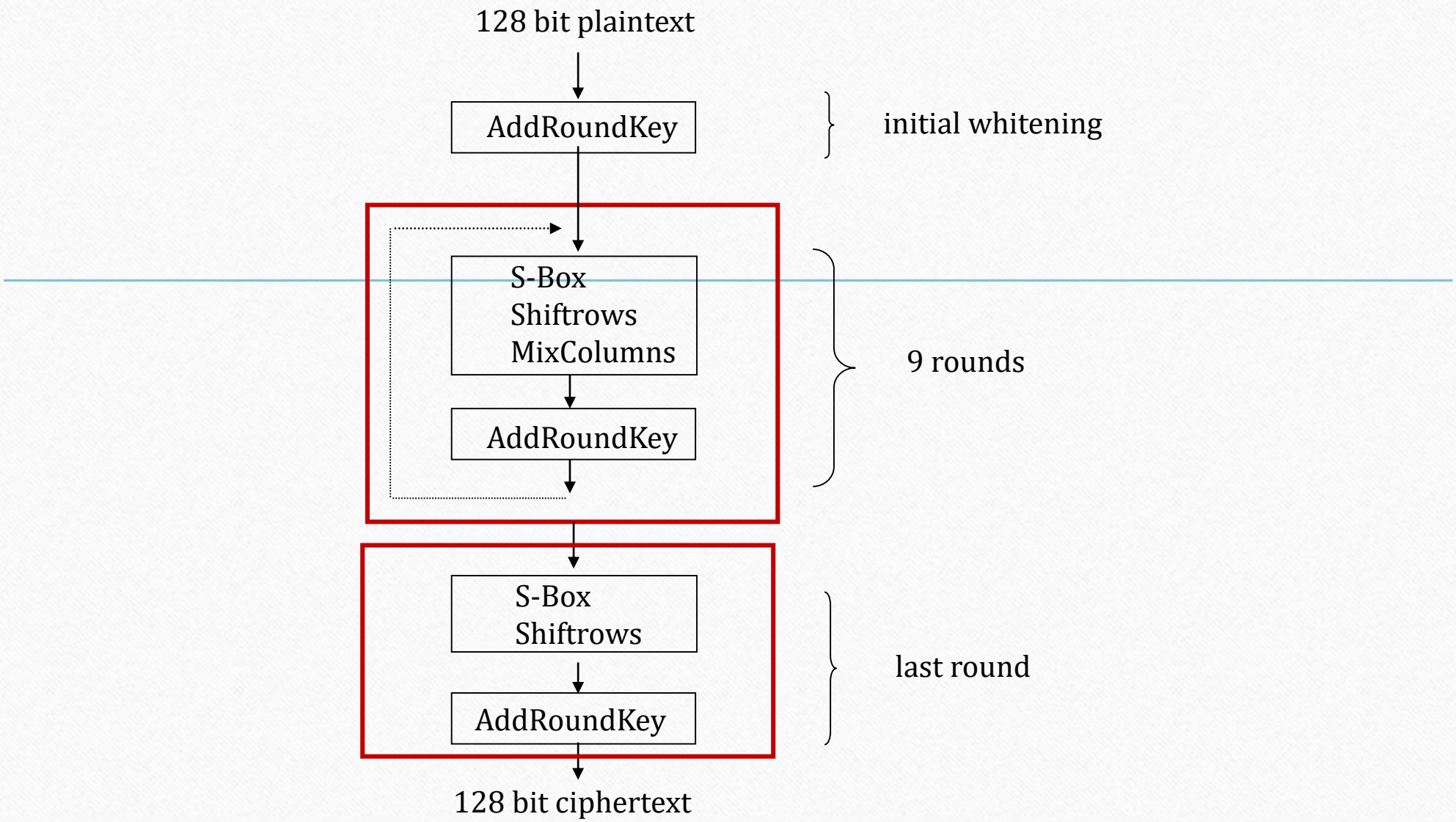
1. Confusion (Substitution): S-Box
2. Diffusion (Permutation): P-Box

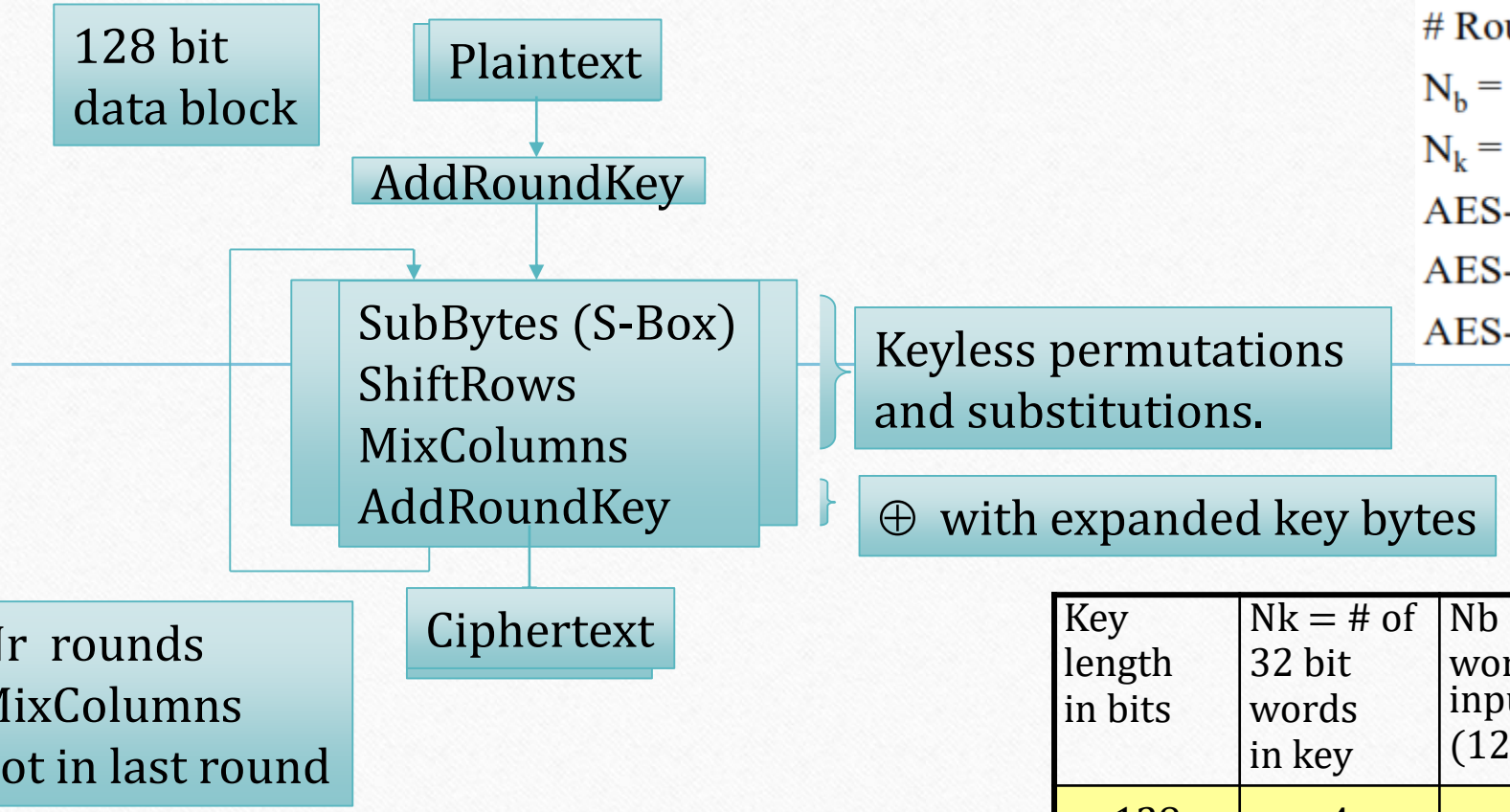
Advanced Encryption Scheme (AES)

AES structure:

- One noteworthy feature of this structure is that it is not a Feistel structure.
  Recall that in the classic Feistel structure, half of the data block is used to modify the other half of the data block, and then the halves are swapped.
- Two of the AES finalists, including Rijndael, do not use a Feistel structure but process the entire data block in parallel during each round using substitutions and permutation.
- 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: 32*4) serve as a round key for each round.


Encryption
Decryption
128-bit Data Block
128-bit Encrypted Data Block
Key Expansion
Key Expansion
Initial Round
Add Round Key
Add Round Key Shift Rows SubBytes
SubBytes Shift Rows Mix Columns Add Round Key
Main Body (9 rounds)
Add Round Key Mix Columns Shift Rows SubBytes
SubBytes Shift Rows Add Round Key
Final Round
Add Round Key
128-bit Encrypted Data Block
128-bit Data Block


128 bit plaintext
AddRoundKey
initial whitening
S-Box
Shiftrows
MixColumns
AddRoundKey
9 rounds
S-Box
Shiftrows
AddRoundKey
last round
128 bit ciphertext



Rounds $N_r = 6 + \max\{N_b, N_k\}$

N_b = 32-bit words in the block

N_k = 32-bit words in key

AES-128: 10

AES-192: 12

AES-256: 14

Key length in bits	Nk = # of 32 bit words in key	Nb = # of words in input/output (128 bits)	Nr = # of rounds
128	4	4	10
192	6	4	12
256	8	4	14

Variable key length and # of rounds.

Decryption not same as encryption.

SubBytes S-Box (table lookup at byte level, see FIPS197 for table values)

ShiftRows

A:

s00	s01	s02	s03
s10	s11	s12	s13
s20	s21	s22	s23
s30	s31	s32	s33

Shift row i
i positions
(i = 0 to 3)

s00	s01	s02	s03
s11	s12	s13	s10
s22	s23	s20	s21
s33	s30	s31	s32

s_{ij} is a byte

MixColumns

$$A \leftarrow \begin{matrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{matrix} * A$$

(in hex)

Usually implemented as a table lookup
Coefficients of a polynomial

AddRoundKey

$$A \leftarrow \text{round\_key} \oplus A$$

Advanced Encryption Scheme (AES)

AES structure:

- Four different stages are used, one of permutation and three of substitution
  1. **Substitute bytes:** Uses an S-box to perform a byte-by-byte substitution of the block
  2. **ShiftRows:** A simple **permutation**
  3. **MixColumns:** A substitution that makes use of arithmetic over $GF(2^8)$
  4. **AddRoundKey:** A simple bitwise XOR of the current block with a portion of the expanded key
- 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.

Advanced Encryption Scheme (AES)

AES structure

- 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.
- The AddRoundKey stage is, in effect, a form of Vernam. The other three stages together provide confusion, diffusion, and nonlinearity, but by themselves would provide no security because they do not use the key.
  - The cipher can be viewed as alternating operations of XOR encryption (AddRoundKey) of a block, followed by scrambling of the block (the other three stages), followed by XOR encryption, and so on.
  - This scheme is both efficient and highly secure.

Advanced Encryption Scheme (AES)

AES structure:

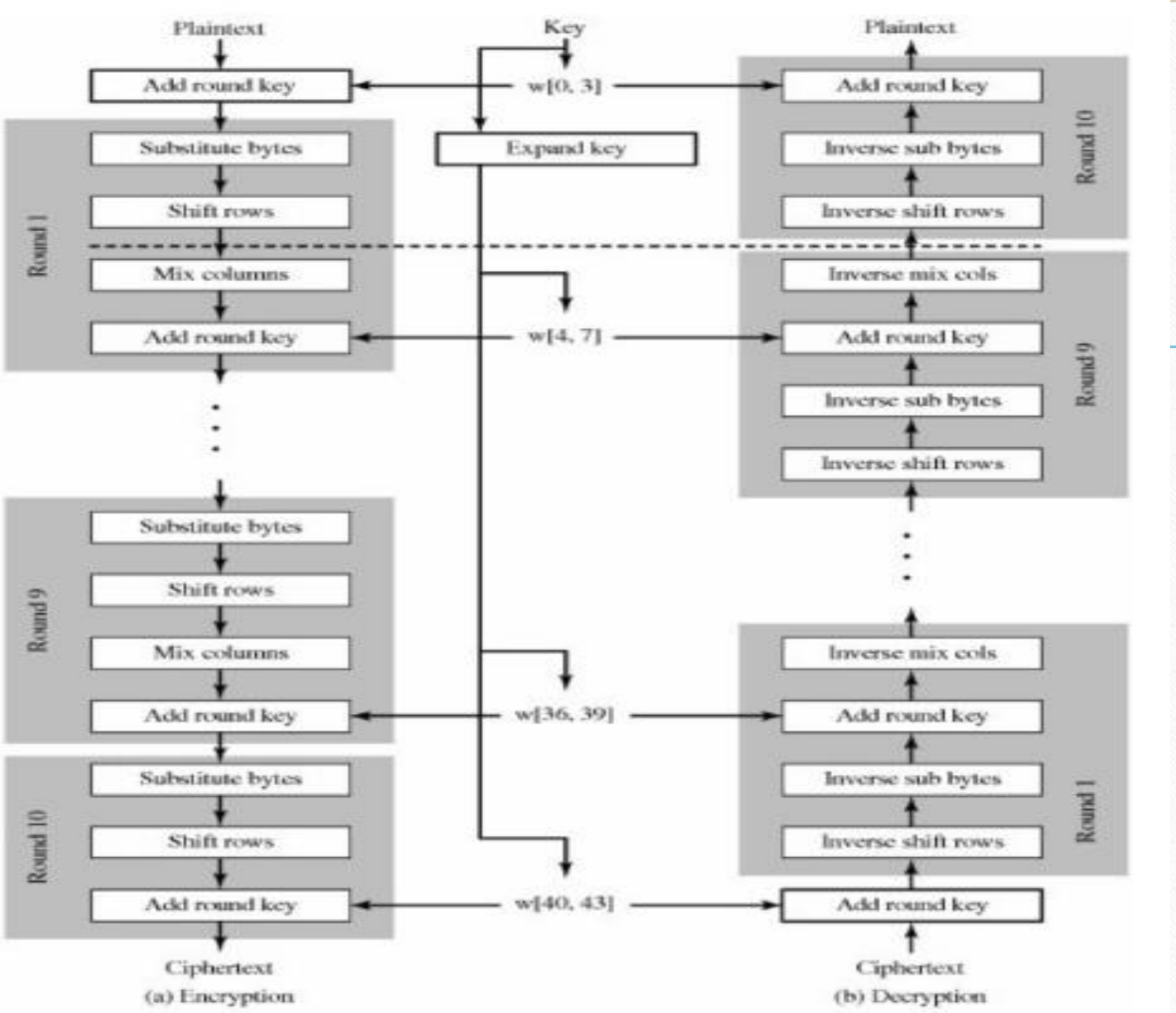
- Each stage is easily reversible. For the Substitute Byte (SubByte), ShiftRows, and MixColumns stages, an inverse function is used in the decryption algorithm. For the AddRoundKey stage, the inverse is achieved by XORing the same round key to the block, using the result that $A \oplus A \oplus B = B$.
- 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 structure of AES.
- Once it is established that all four stages are reversible, it is easy to verify that decryption does recover the plaintext.

Advanced Encryption Scheme (AES)

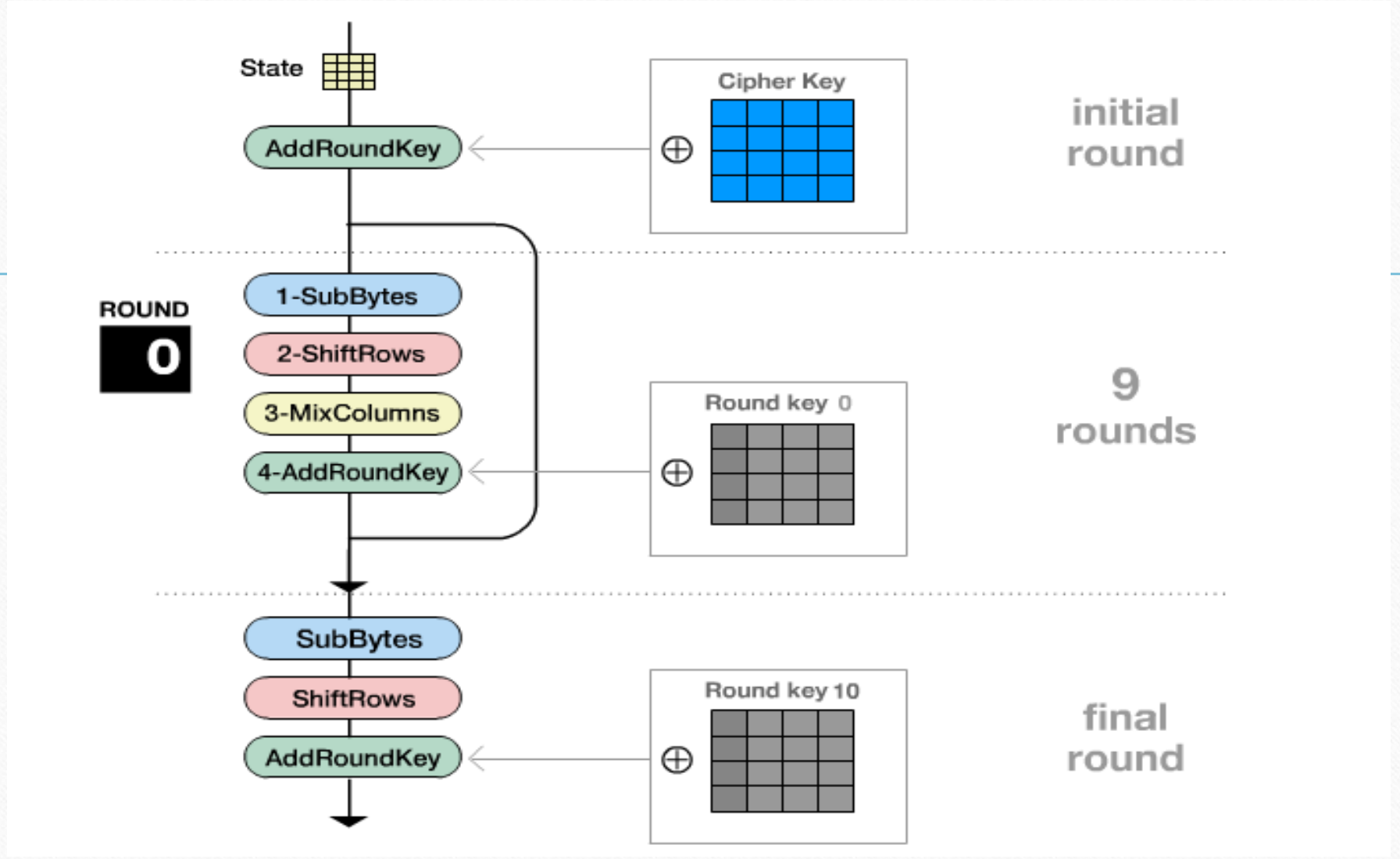
AES structure:

- Figure below lays out encryption and decryption going in opposite vertical directions. At each horizontal point (e.g., the dashed line in the figure), **State is the same for both encryption and decryption.**

- 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.


Plaintext
Add round key
Substitute bytes
Shift rows
Mix columns
Add round key
Substitute bytes
Shift rows
Mix columns
Add round key
Substitute bytes
Shift rows
Add round key
Ciphertext
(a) Encryption
Round 1
Round 9
Round 10
Key
w[0, 3]
Expand key
w[4, 7]
w[36, 39]
w[40, 43]
Plaintext
Add round key
Inverse sub bytes
Inverse shift rows
Inverse mix cols
Add round key
Inverse sub bytes
Inverse shift rows
Inverse mix cols
Add round key
Inverse sub bytes
Inverse shift rows
Add round key
Ciphertext
(b) Decryption
Round 10
Round 9
Round 1

AES: Encryption Process



AES: Encryption Process

This is a block from the plain text message to be encrypted

State

32	88	31	e0
43	5a	31	37
f6	30	98	07
a8	8d	a2	34

Cipher Key

2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

To encryption process

To key Schedule

00110010
00101011
00011001 = 19

10001000
00101000
10100000 = A0

hexadecimal notation:

Ex: 32 = 00110010 (1 byte)

3hex 2hex

AES: Rounds

- The Rijndael cipher consists of

1. An initial Round Key addition
2. N_{r-1} Rounds
3. A final round

$$Round(State, RoundKey)$$
$$\{$$
$$ByteSub(State);$$
$$ShiftRow(State);$$
$$MixColumn(State);$$
$$AddRoundKey(State, RoundKey);$$
$$\}$$

AES: Rounds

- The final round of the cipher is slightly different and is defined by;

$$FinalRound(State, RoundKey)$$
$$\{ByteSub(State);$$
$$ShiftRow(State);$$
$$AddRoundKey(State, RoundKey);$$
$$\}$$

- The functions (Round, ByteSub, ShiftRow, …)operate on arrays to which pointers (State, RoundKey) are provided.
- **Note:** The Expanded Key shall **always** be derived from the Cipher Key and never be specified directly.
- There are however no restrictions on the selection of the Cipher Key itself.

AES: Transformations

- There are 4 transformations namely,

1. SubBytes

2. ShiftRows

3. MixColumns

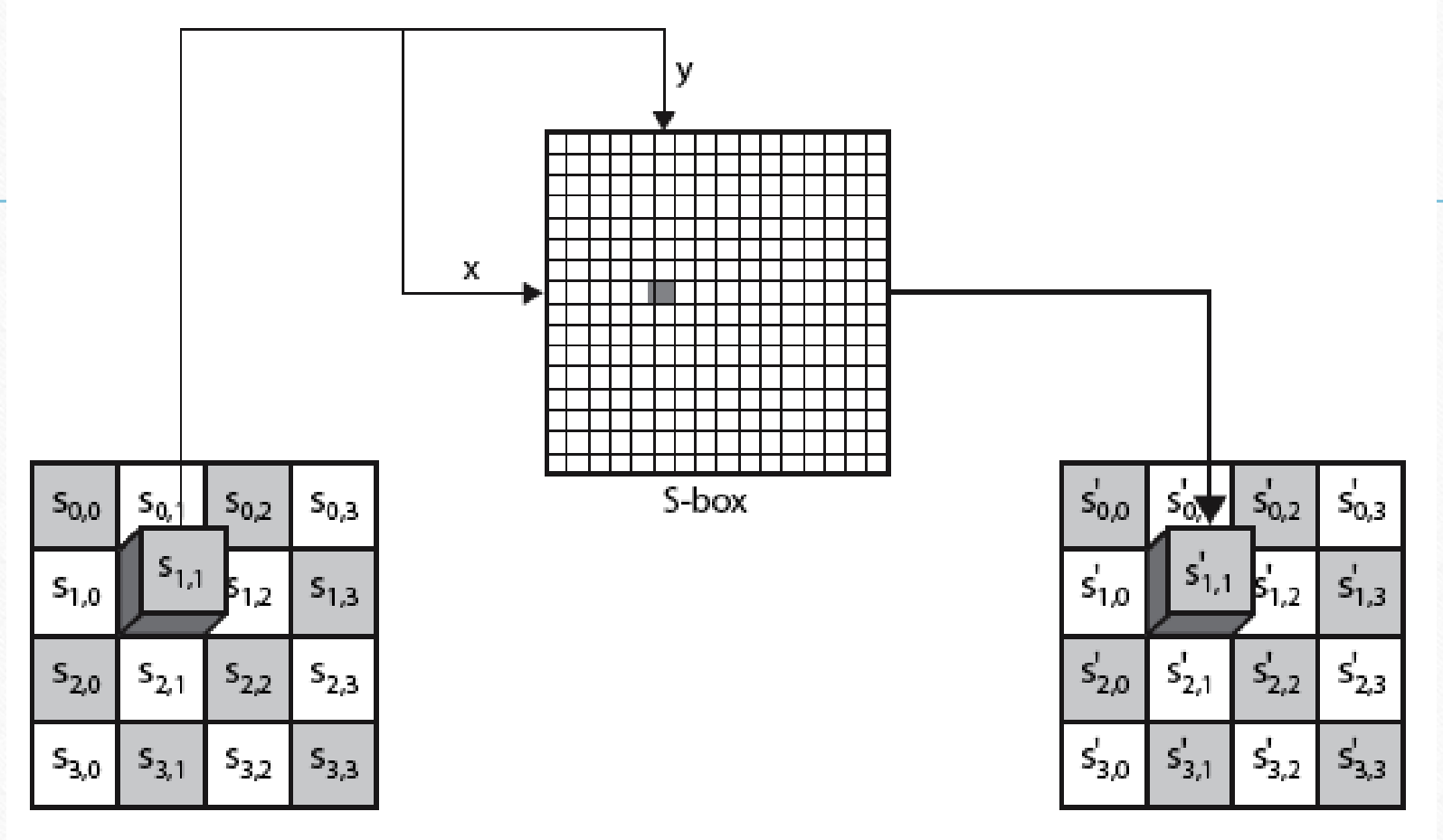
4. AddRoundKey

AES: Rounds

Byte Substitution (SubByte)

- A simple substitution of each byte
- Uses one S-box of 16x16 bytes containing a permutation of all 256 8-bit values
- Each byte of state is replaced by byte indexed by row (left 4-bits) & column (right 4-bits)
  - eg. byte {95} is replaced by byte in row 9 column 5
  - which has value {2A}
- S-box constructed using defined transformation of values in GF(256)

AES: SubBytes



AES: SBox

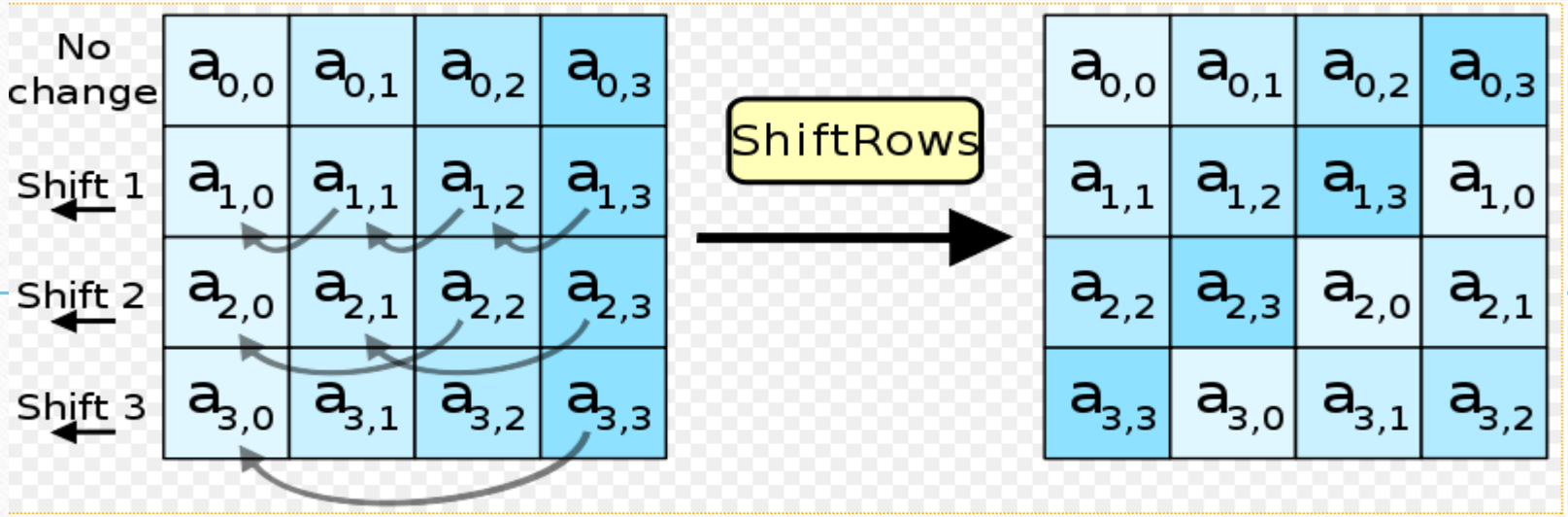
		y														
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	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: Rounds

ShiftRows

- A circular byte shift in each row
  - 1st row is unchanged
  - 2nd row does 1 byte circular shift to left
  - 3rd row does 2 byte circular shift to left
  - 4th row does 3 byte circular shift to left
- Decrypt inverts using shifts to right
- Since state is processed by columns, this step permutes bytes between the columns

AES: Rijndael ShiftRows



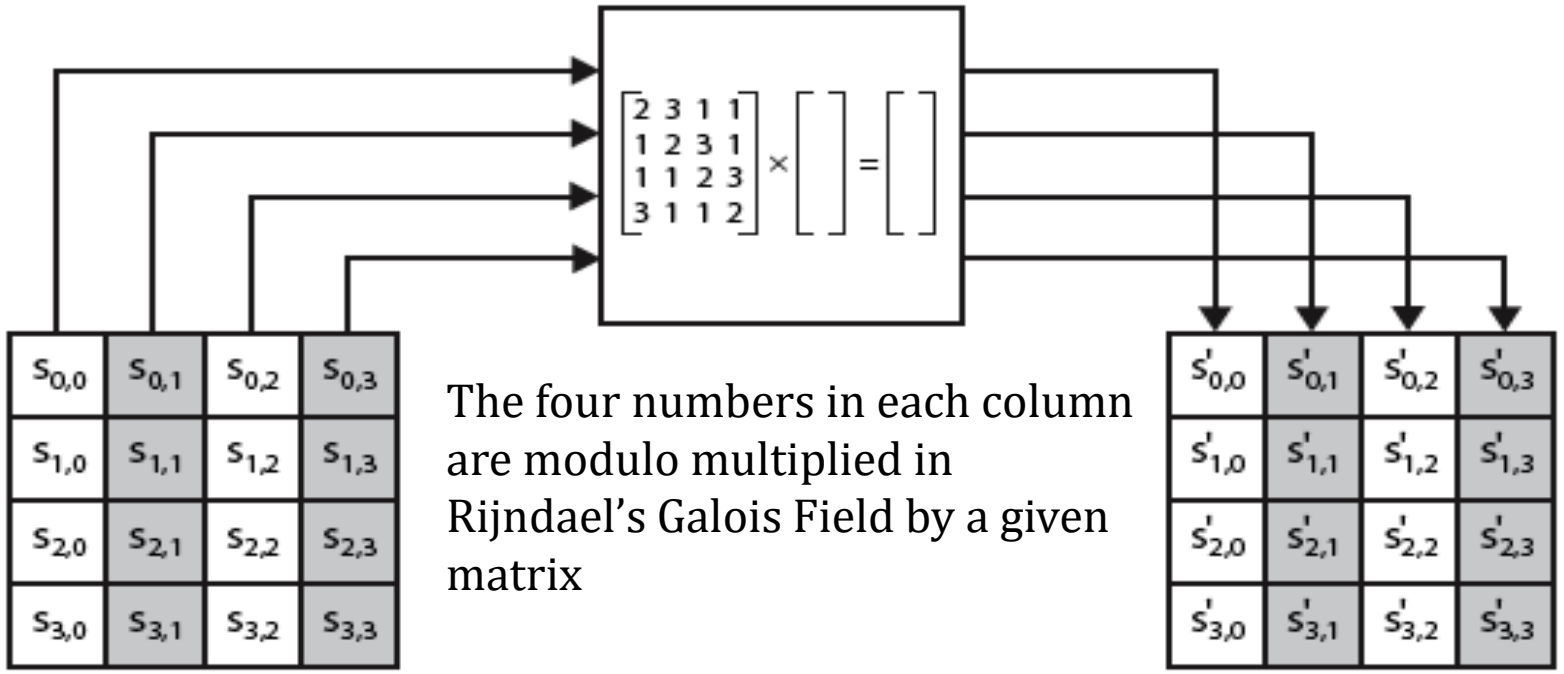
d4	e0	b8	1e
27	bf	b4	41
11	98	5d	52
ae	f1	e5	30

d4	e0	b8	1e
bf	b4	41	27
5d	52	11	98
30	ae	f1	e5

AES: Rounds

MixColumns

- Each column is processed separately
- Each byte is replaced by a value dependent on all 4 bytes in the column



The four numbers in each column are modulo multiplied in Rijndael's Galois Field by a given matrix

AES: Rounds

MixColumns

- Example

03.Bf

$$03.bf = \{10\ XOR\ 01 = 11\}.\{10111111\}$$
$$= \{10111111.10\}XOR\{10111111.01\}$$
$$= \{10111111.10\}XOR\{10111111\}$$
$$= \{01111110\}XOR\{00011011\}XOR\{10111111.01\}$$
$$= \{11011010\}$$

AES: Rounds

MixColumns

- Example

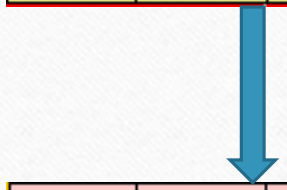
{d4}.{02} = 1101 0100 << 1 (<< is left shift, 1 is the number of shift done, pad on with 0's)

= 1010 1000 XOR 0001 1011 (because the leftmost is a 1 before shift)

= 1011 0011

AES: Rijndael MixColumns

d4	e0	b8	1e
bf	b4	41	27
5d	52	11	98
30	ae	f1	e5



04	e0	48	28
66	cb	f8	06
81	19	d3	26
e5	9a	7a	4c

$$\begin{bmatrix} d4 \\ bf \\ 5d \\ 30 \end{bmatrix} \bullet \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} = \begin{bmatrix} 04 \\ 66 \\ 81 \\ e5 \end{bmatrix}$$

02	0000 0010
d4	1101 0100
02 × d4	1 1010 1000
m(x)	1 0001 1011
b3	1011 0011

$02 \times d4 = b3$

$03 \times bf = da$

$01 \times 5d = 5d$

$01 \times 30 = 30$

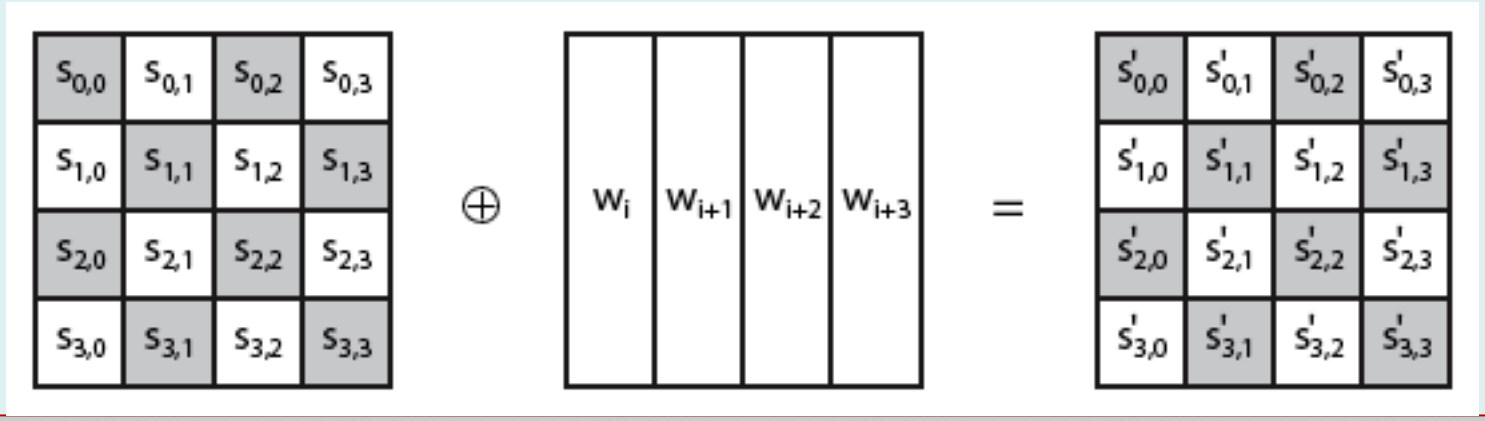
M(x)=column mixer

$b3 \oplus da \oplus 5d \oplus 30 = 04$

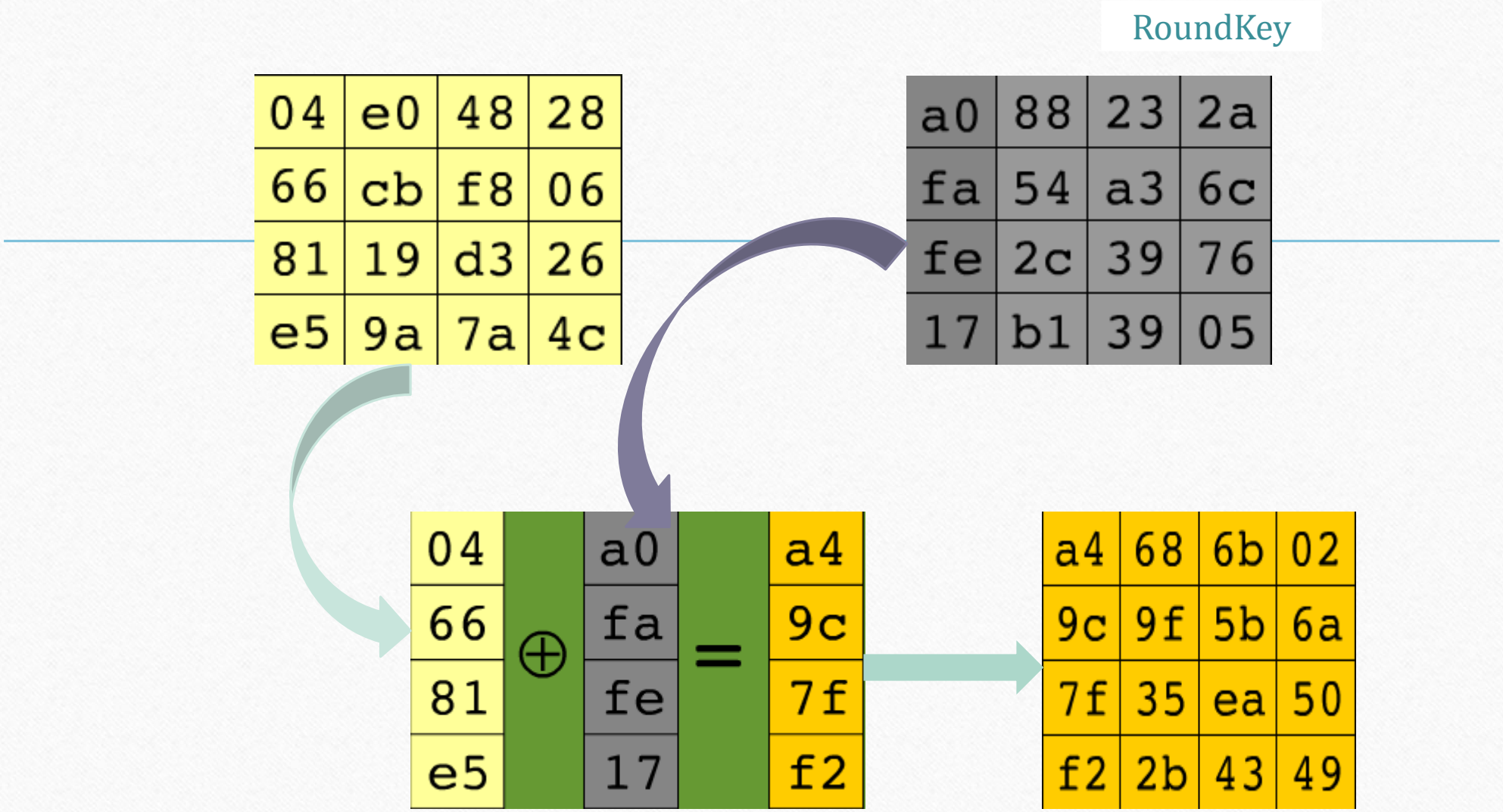
AES: Rounds

AddRoundKey

- XOR state with 128-bits of the round key
- Again processed by column (though effectively a series of byte operations)
- Inverse for decryption identical
  - Since XOR own inverse, with reversed keys
- Designed to be as simple as possible

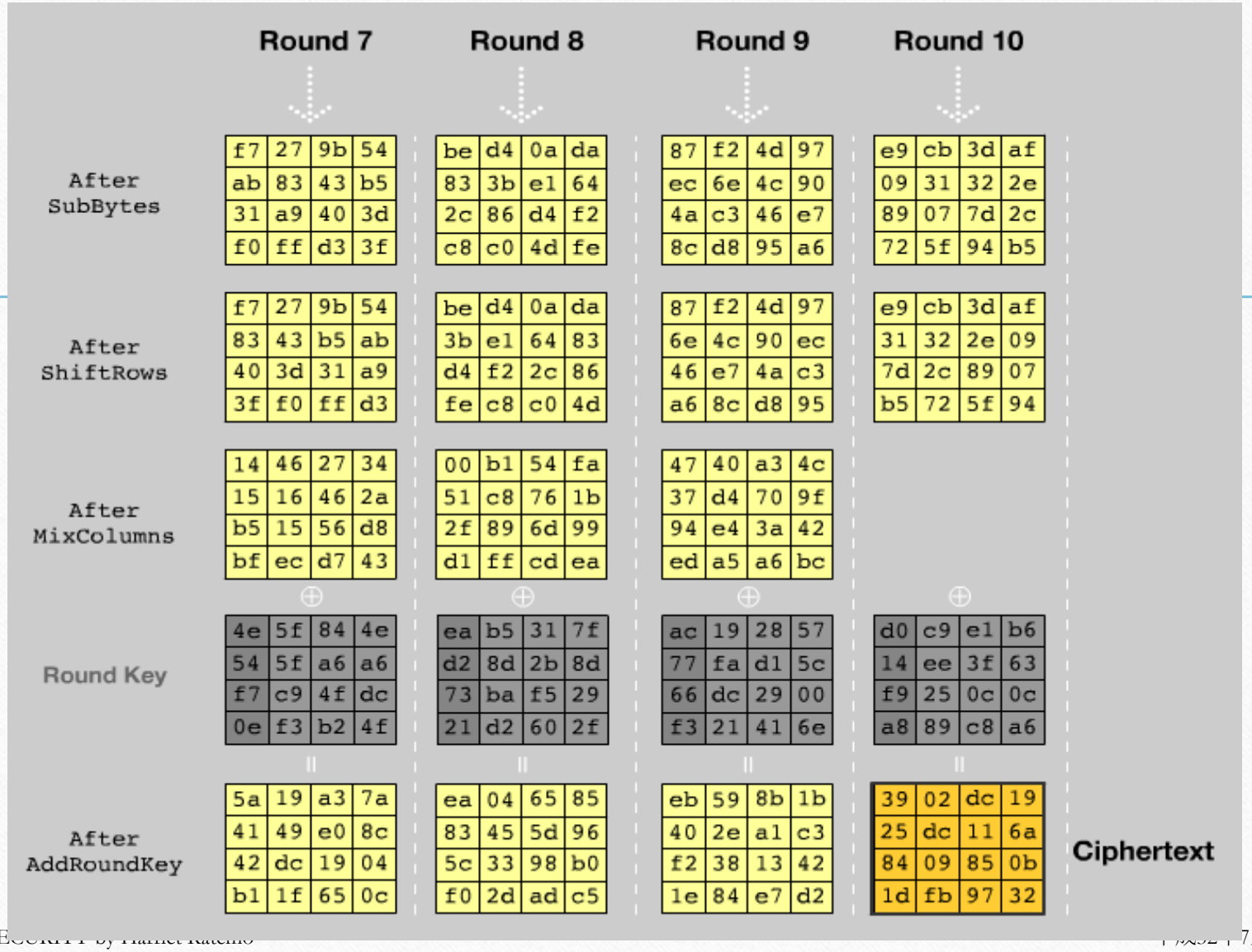


AES: Rijndael AddRoundKey



AES: Rijndael

AES: Rijndael

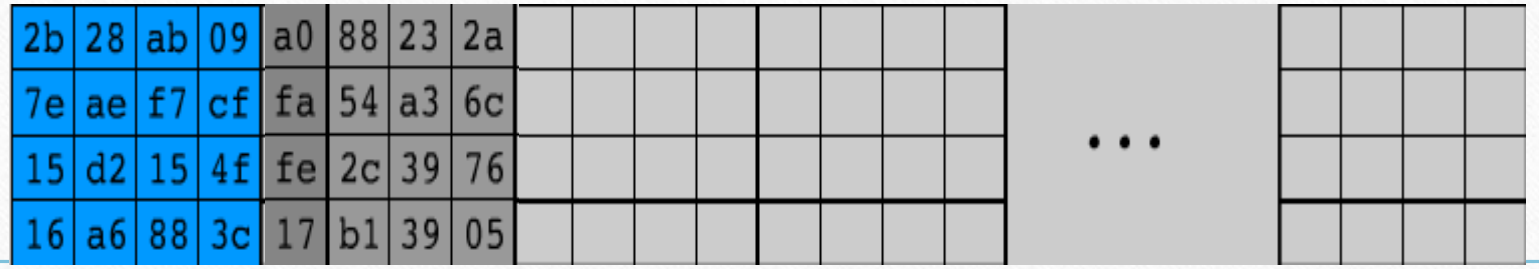


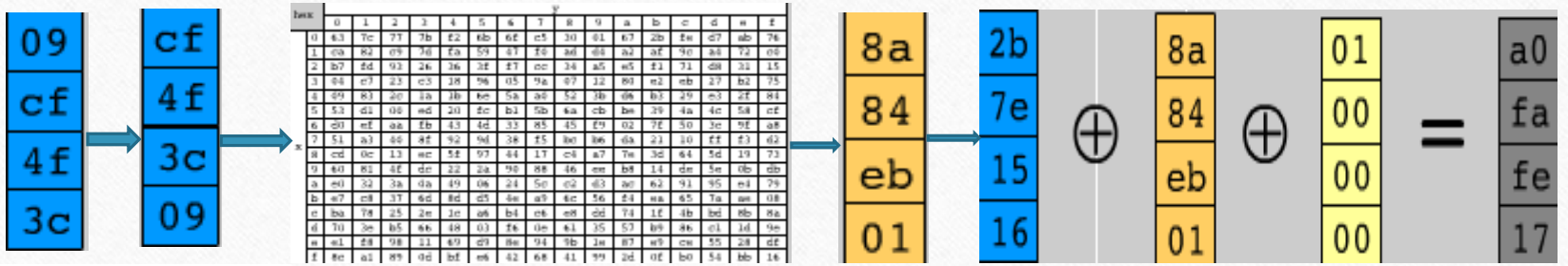
AES: Riindael Key Schedule

Cipher Key

Round Key 1

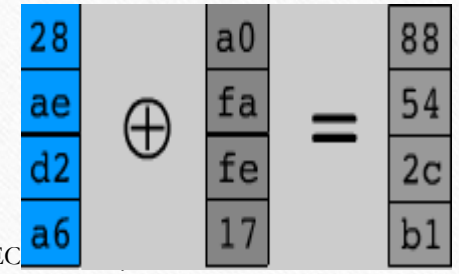
Round Key10

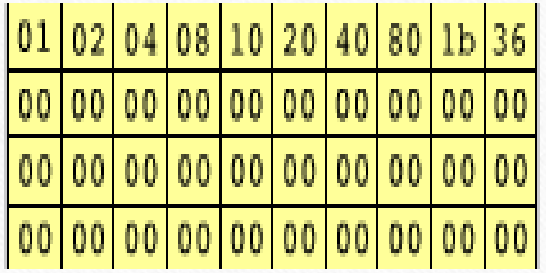




RotWord(word rotation)

S-Box





Rcon(round constant)

AES: Rijndael Key Schedule

		y														
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	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

Rijndael Summary

- SubBytes stage: Intended to achieve a non linear substitution cipher which is an important property for a block cipher to prevent differential cryptanalysis
- ShiftRows and MixColumns: Intended to achieve a mixture of the bytes positioned at different places on a plaintext message block
- AddRoundKey: Provides the necessary secret randomness to the message distribution

Advantages of Rijndael

i. Simplicity of design: The cipher does not base its security on obscure and not well understood interactions between arithmetic operations

ii. Variable block length: The block lengths of 192 and 256 bits allows the construction of a collision-resistant iterated hash function using Rijndael as the compression function

iii. The possibility of extensions: Although the number of rounds is fixed in the specifications, it can be increased as a parameter in case of security problems.

iv. Secured encryption/decryption system: The expanded key is always derived from the cipher key.

Rijndael Summary

Impact of AES on Applied Cryptography

- No need for multiple encryption i.e. triple DES due to enlarged and variable key size and data block sizes of 128, 192, 256-bits
- Its wide use is hoped to lead to an emergence of new hash functions of compatible security strengths.

Average Time Required for Exhaustive Key Search

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	$2^{128} \approx 3.4 \times 10^{38}$	$2^{127}\ ns = 5.3 \times 10^{21} years$	5.3×10^{17} years
168	3DES	$2^{168} \approx 3.7 \times 10^{50}$	$2^{167}\ ns = 5.8 \times 10^{33} years$	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	$2^{191}\ ns = 9.8 \times 10^{40} years$	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	$2^{255}\ ns = 1.8 \times 10^{60} years$	1.8×10^{56} years
26 characters permutation	Monoalphabetic	$26! = 4 \times 10^{26}$	$2 \times 2^{26}\ ns = 6.3 \times 10^{9} years$	6.3×10^{6} years


BREAK