SECRET KEY CRYPTO

Anwitaman DATTA
SCSE, NTU Singapore

 |

Acknowledgement: The following lecture slides are based on, and uses material from the text book **Cryptography and Network Security** (various eds) **by** **William Stallings**

⌘ Stream and Block ciphers

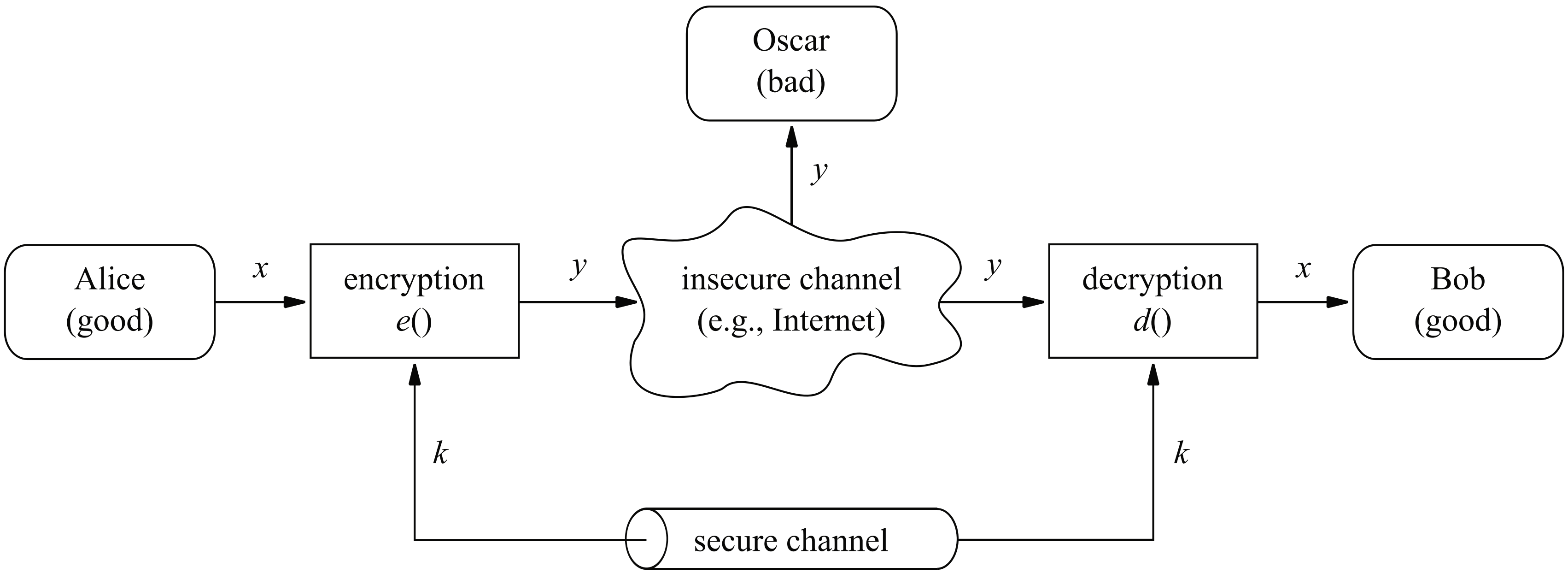
⌘ DES and AES algorithms

⌘ Modes of operations

SECRET KEY

CRYPTO

System model

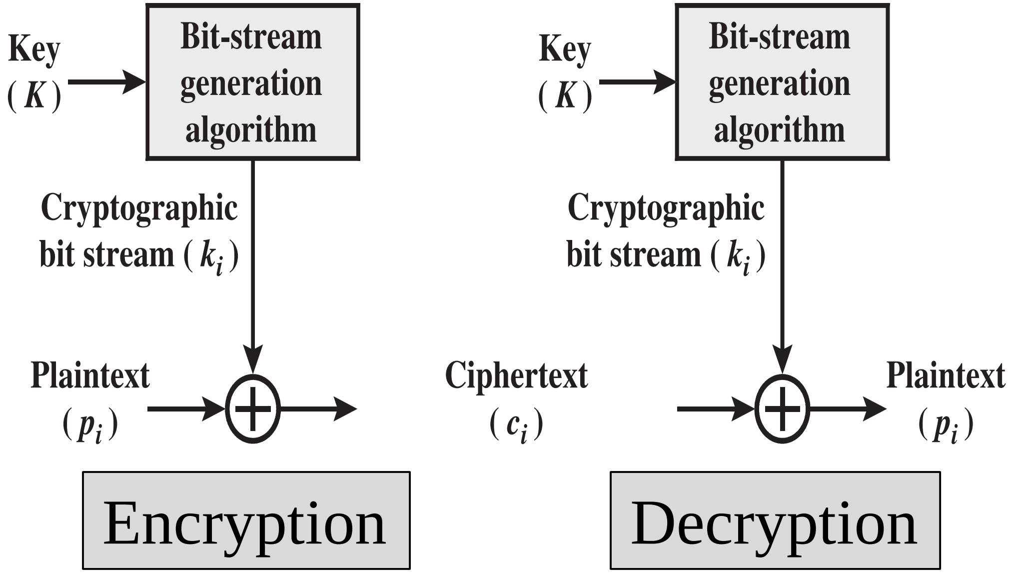


- Secret key (a.k.a symmetric key) cryptography

Stream ciphers

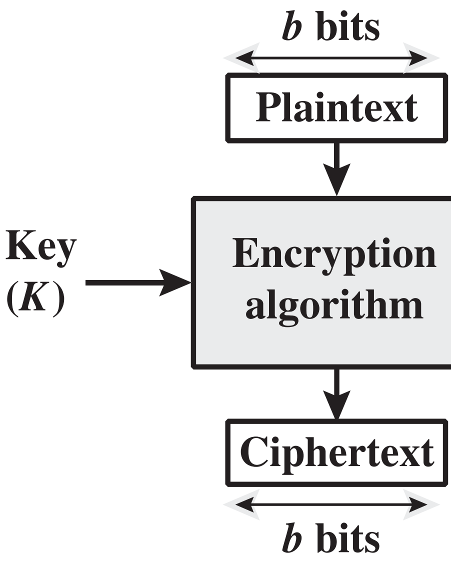
Process one symbol (e.g. bit/byte) at a time, e.g.:

- Vigenère and Vernam ciphers
- one time pad
- ChaCha20 used in TLS/SSL implementations



Block ciphers

A block of plaintext is processed together, to create a block of ciphertext (of same size).
e.g.: DES, AES, …
- can be used to create a stream cipher

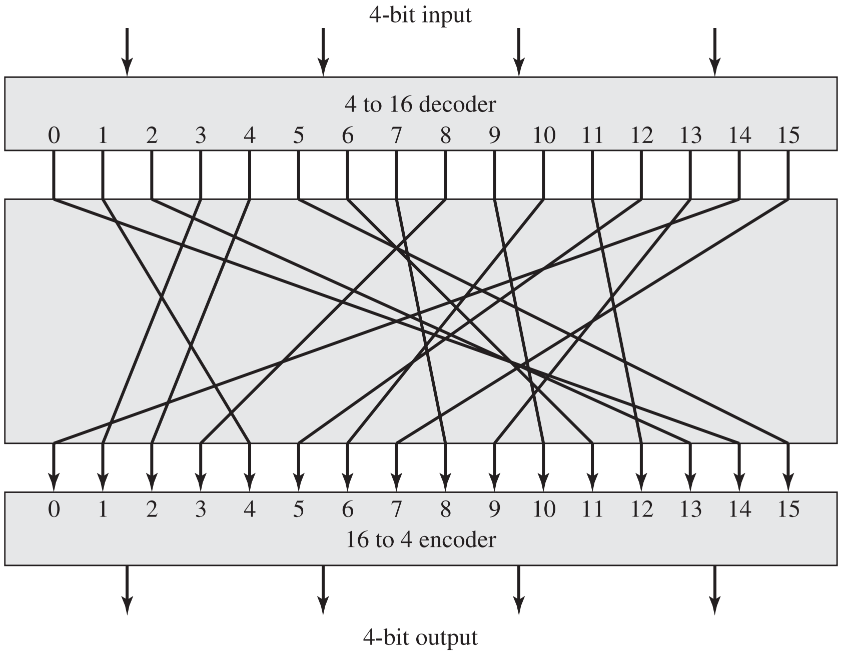


Essentially *a mapping* (for a *b*-bits block)
- Input space: 2^b
- Output space: 2^b

How many such mappings exist?
- In general?
- That are reversible? $(2^b)!$

Block ciphers

4-bits example



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

4×2^4 bits required to represent mapping

- Ideal block cipher
- Practical?

In absence of an ideal cipher …

Use tractable building blocks

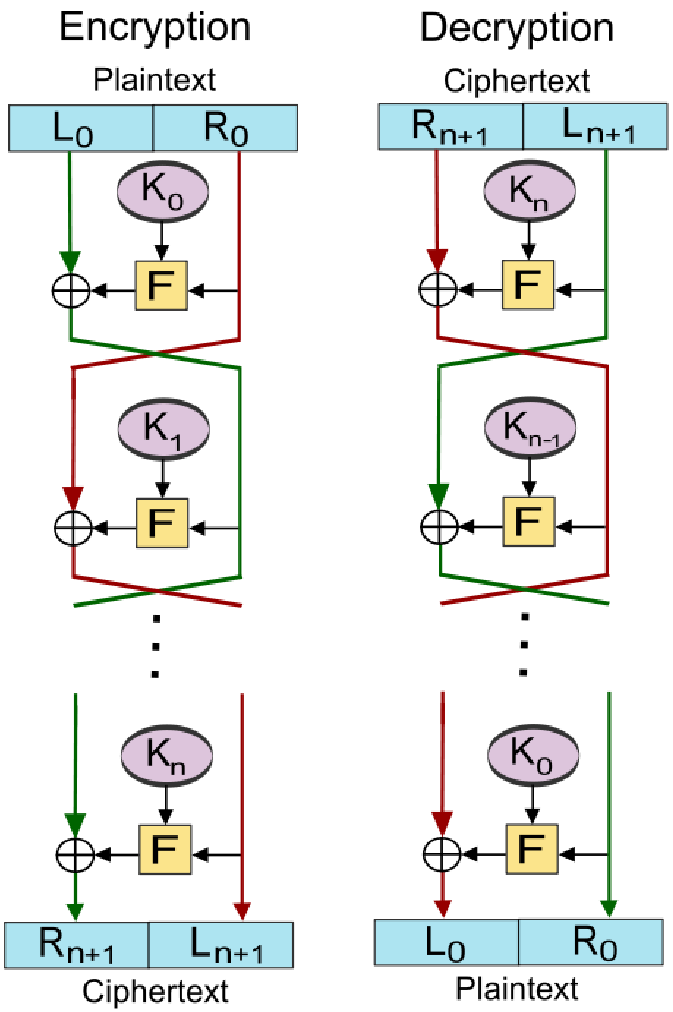
Diffusion	• Dissipate the statistical structure of the plaintext
Confusion	• Make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible
Cascade	• (Re-)apply multiple times the smaller units of encryption, to realize a stronger encryption

⌘ Often achieved with a Substitution-Permutation network

e.g., AES, somewhat open to interpretation: Feistel network (used in DES)

Check also: https://en.wikipedia.org/wiki/Confusion_and_diffusion

Feistel cipher

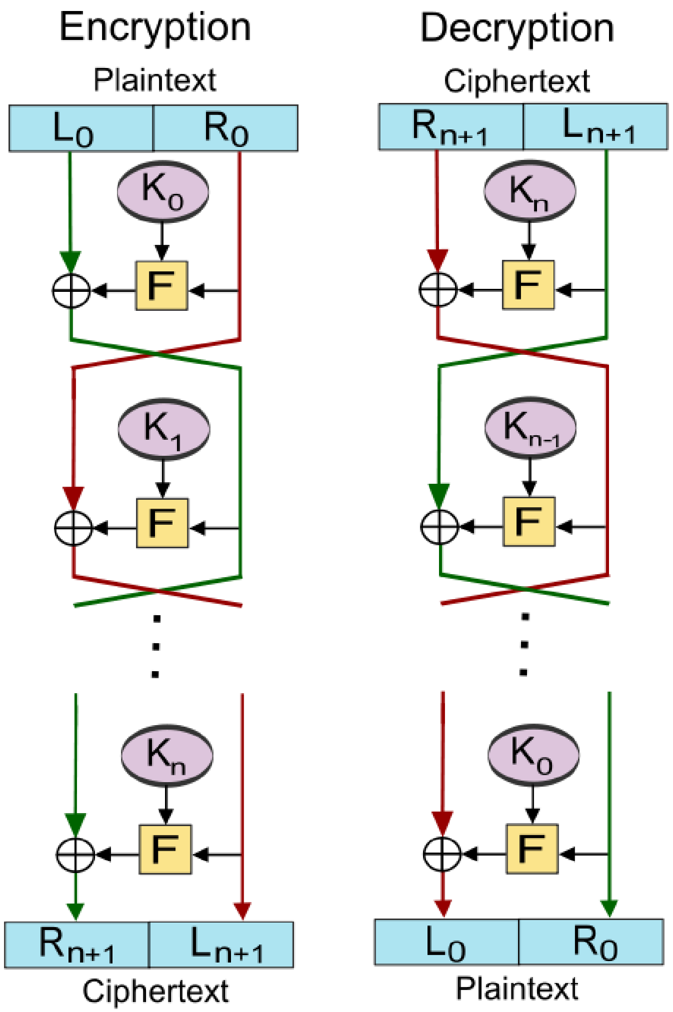


Horst Feistel
1915-1990

Split input in two halves

- Alternatively repeat:
- Substitution with round function $F(-,K_i)$ and XOR
- Permutation: swap to halves

Feistel cipher



Considerations

- Block size
- Key size
- Number of rounds

DES: Data Encryption Standard

Based on Feistel's work at IBM since late 1960s

1971
LUCIFIER: Lloyd's London cash dispensing machine

1973
National Bureau for Standards (**NBS**) RFP for national cipher standard

1977
Adopted as Data Encryption Standard (**DES**)

1994
National Institute of Standards and Technology (**NIST**) reaffirms DES for federal use

1998
Electronic Frontier Foundation (**EFF**) breaks DES w/ $250K machine

DES

Big picture

64 bit blocks

64 bit keys

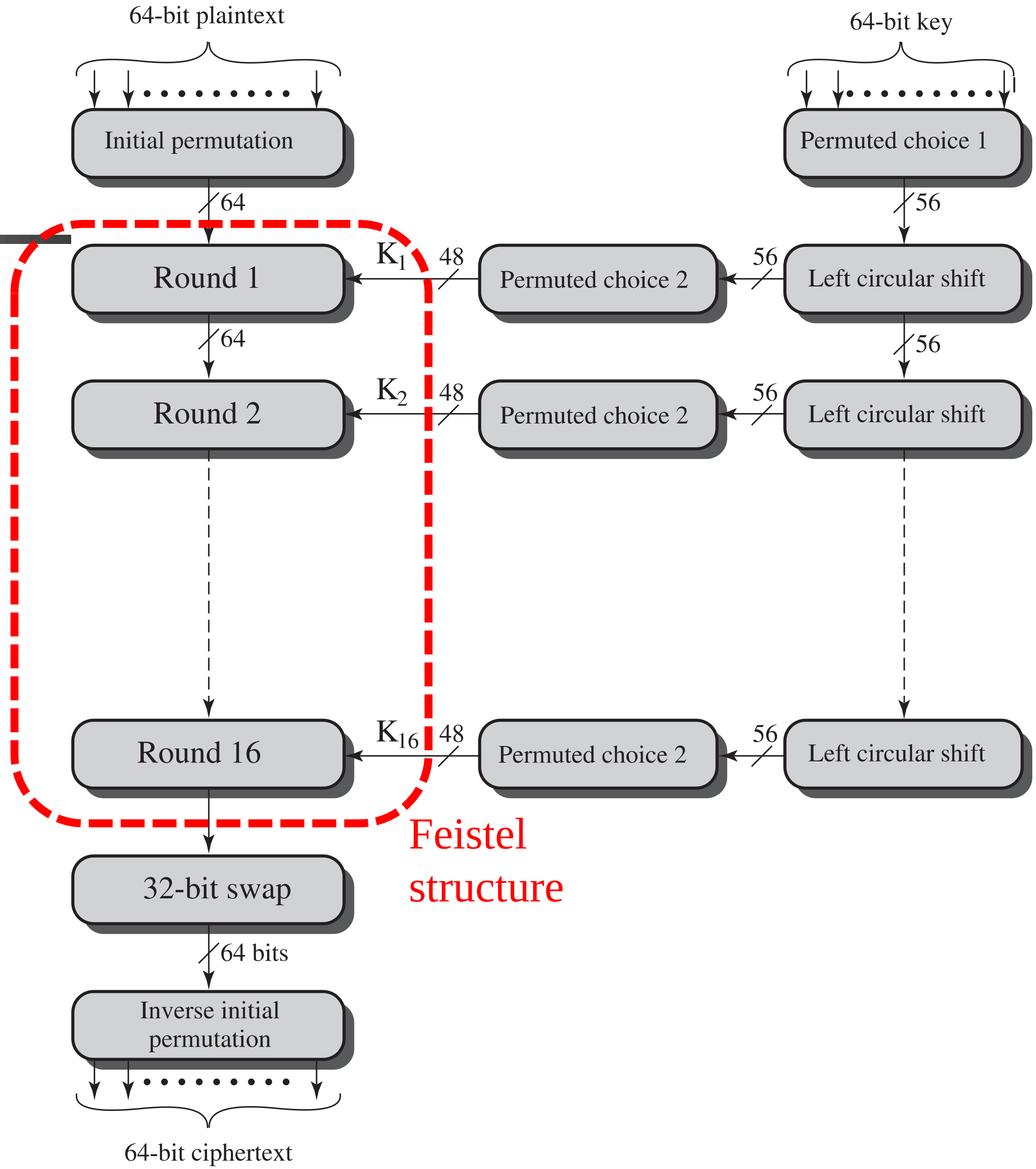
- Only 56 bits used

3 phases

- Initial permutation (IP)
- Repeated rounds

  Feistel structure

- IP^{-1}



DES

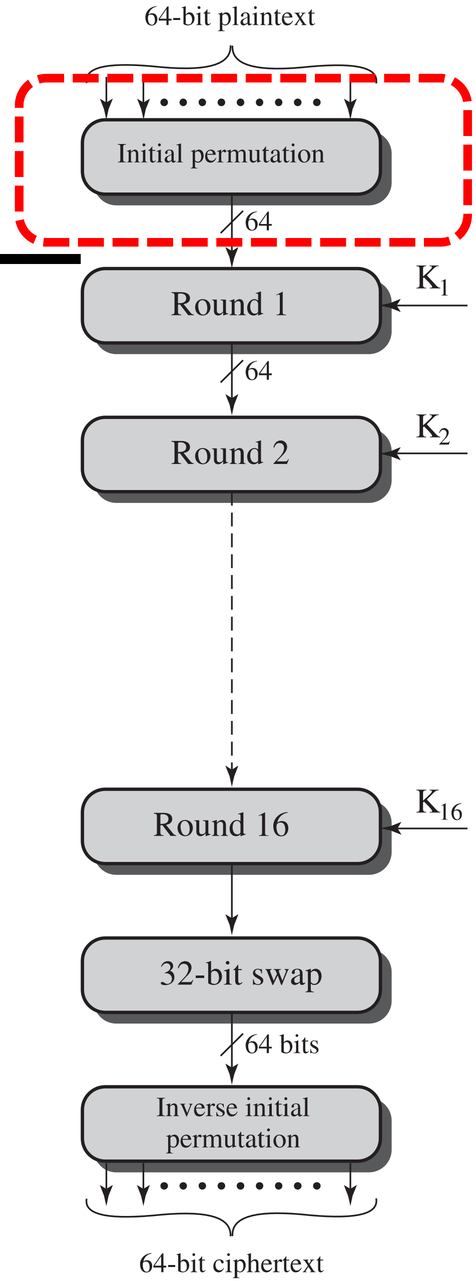
Big picture

3 phases

- Initial permutation (IP)
- Repeated rounds

  Feistel structure
- IP^{-1}

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



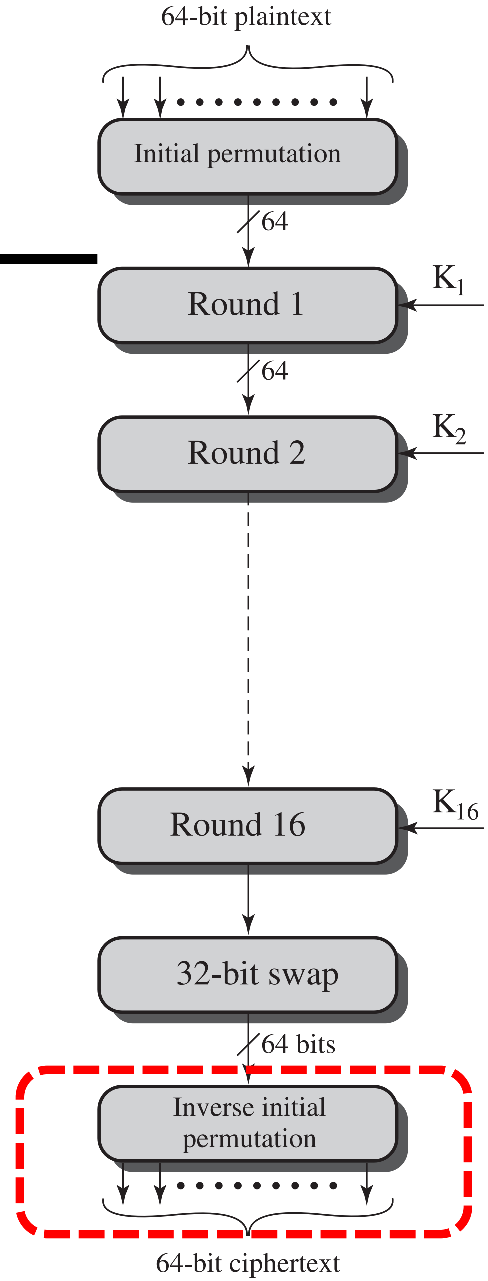
DES

Big picture

3 phases

- Initial permutation (IP)
- Repeated rounds
  Feistel structure
- 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

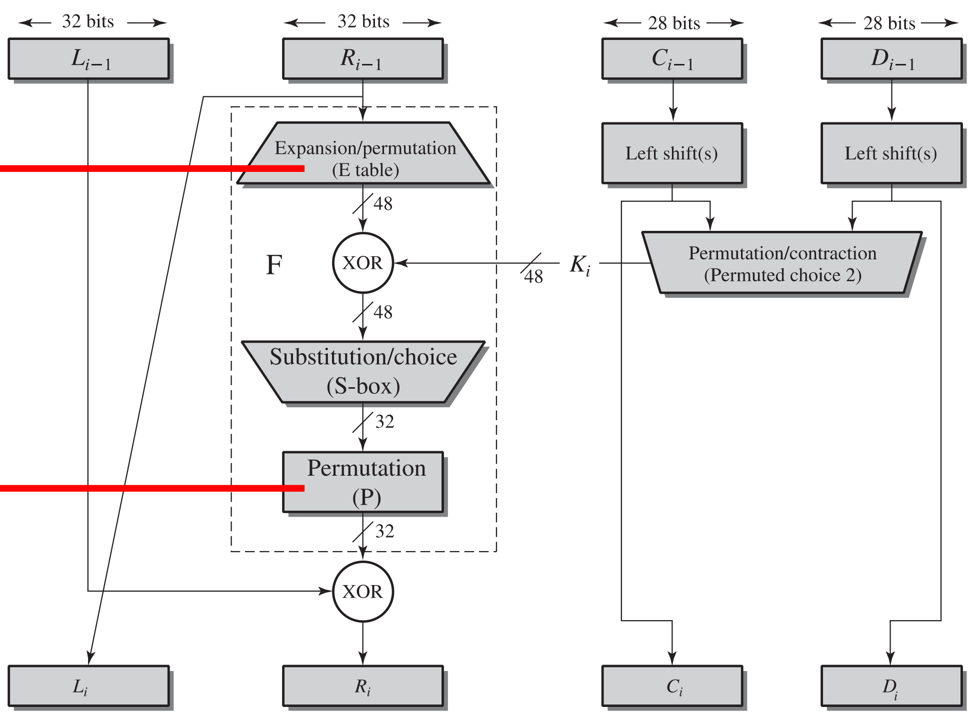


DES

Inside a round of DES

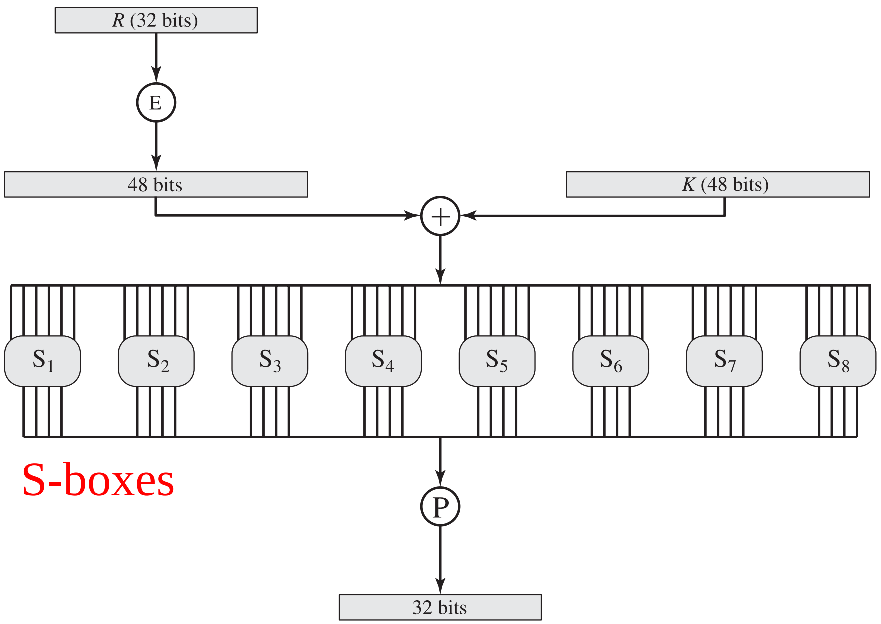
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

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



DES

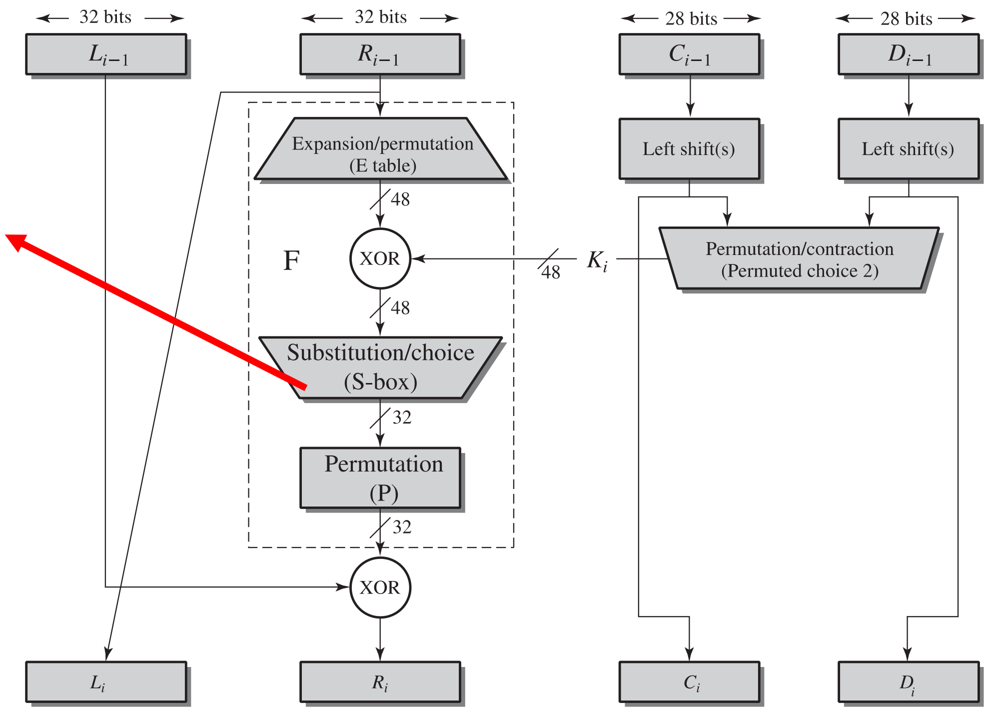
Inside a round of DES



First & last bits decide the row of S_i

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

Remaining bits determine the column



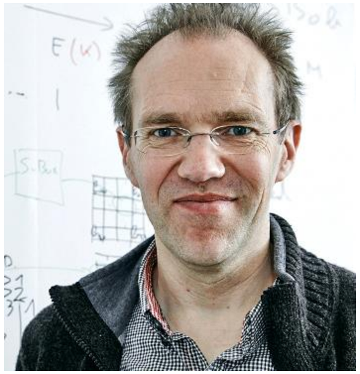
DES

Self study: Sections 4.1 through 4.3 of the (7ed) Stallings textbook

AES: Advanced Encryption Standard

aka Rijndael

Vincent Rijmen
born in 1970



Joan Daemen
born in 1965

➡ AES-128 web demo

AES: Advanced Encryption Standard

aka Rijndael

All AES operations are on 8-bit byte strings

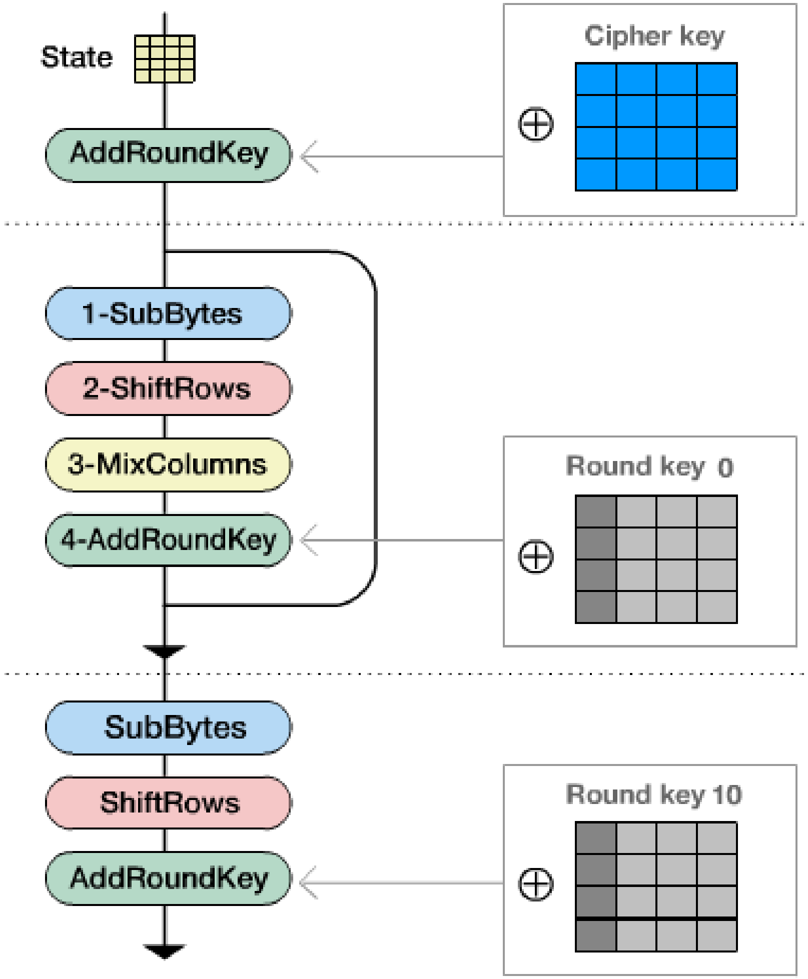
- Addition, multiplication and division in GF(2^8)
- Recall: need *polynomial arithmetic*
- All AES GF(2^8) computations are based on the ***irreducible polynomial***

$$x^8 + x^4 + x^3 + x + 1$$

AES
Big picture

Substitution-Permutation network:

- 128 bits plaintext input
- 16/24/32 byte keywords
  AES-128, AES-192, AES-256
- 10/12/14 rounds
- Four types of transforms per round



AES

Big picture

Inputs:

- Plaintext 4×4 column major order matrix of bytes (termed as the state)
- Cipher key
  - which is *expanded into round keys*
  - serves as an input to AddRoundKey transformation in each round

in_0	in_4	in_8	in_{12}
in_1	in_5	in_9	in_{13}
in_2	in_6	in_{10}	in_{14}
in_3	in_7	in_{11}	in_{15}

AES

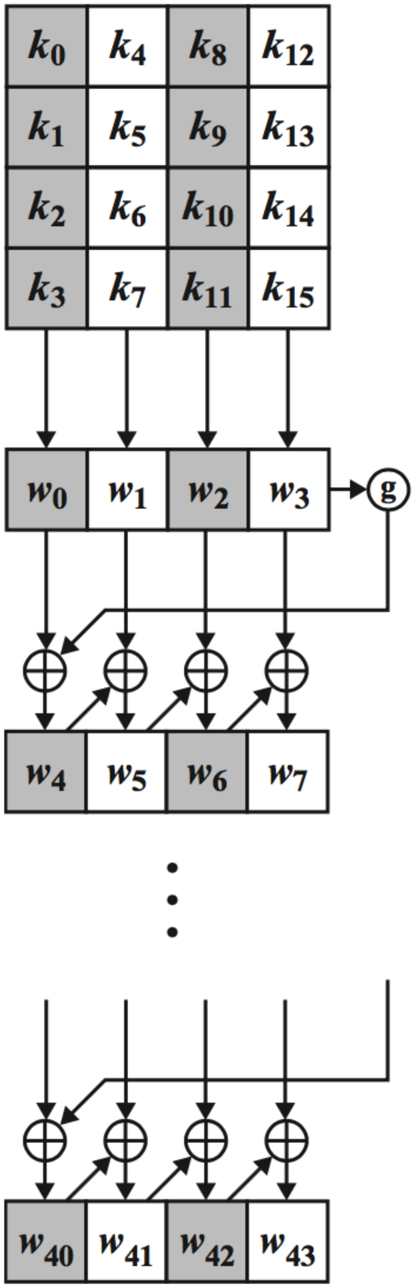
Key expansion: round keys generation

Expand into N+1 round keys

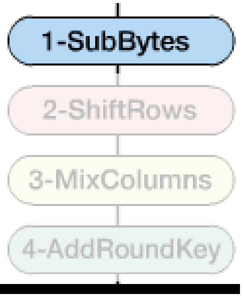
- Four word (16 byte) key mapped into a linear array of 44 words (176 bytes)
- Function g() involves byte rotation, substitution and XOR with some round constant

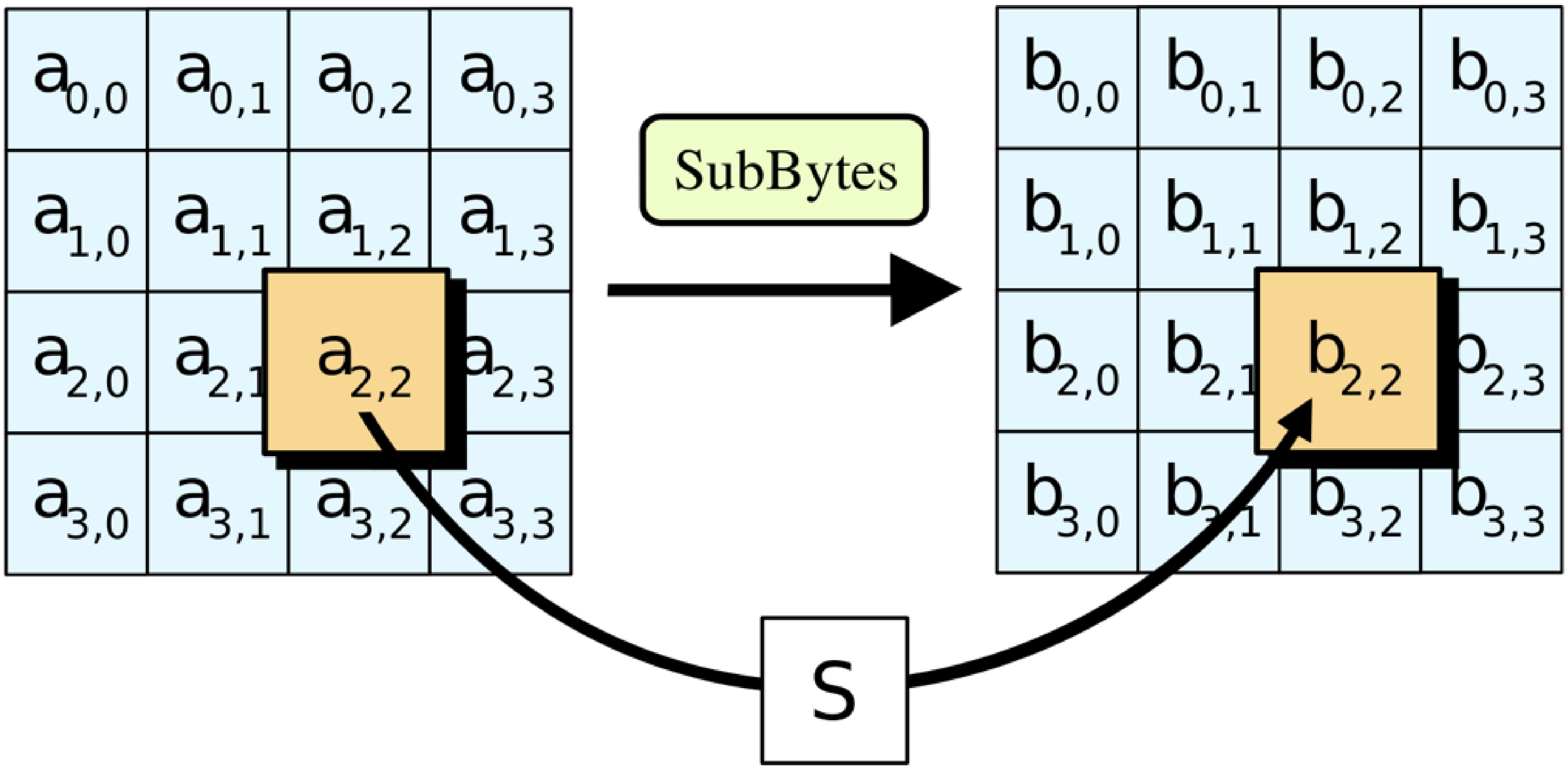
Purpose

- Diffusion of cipher key differences
- Non-linearity and elimination of symmetries



AES
Substitute Bytes Transform





1-SubBytes
2-ShiftRows
3-MixColumns
4-AddRoundKey

AES S-box: look-up table

➡ AES S-box: calculation web demo

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	[illegible]	[illegible]	[illegible]	[illegible]	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	[illegible]	[illegible]	[illegible]	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	[illegible]	44	[illegible]	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	[illegible]	88	46	EE	B8	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
A	E0	32	3A	0A	49	06	24	5C	C2	D3	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	37	6D	8D	D5	4E	A9	6C	56	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	[illegible]	8B	8A
[illegible]	[illegible]	[illegible]	[illegible]	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
[illegible]	[illegible]	[illegible]	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

e.g., 95 → 2A

Contains permutation for all 256 possible 8-bit values

left 4-bits to determine row
right 4-bits for column

AES S-box: construction

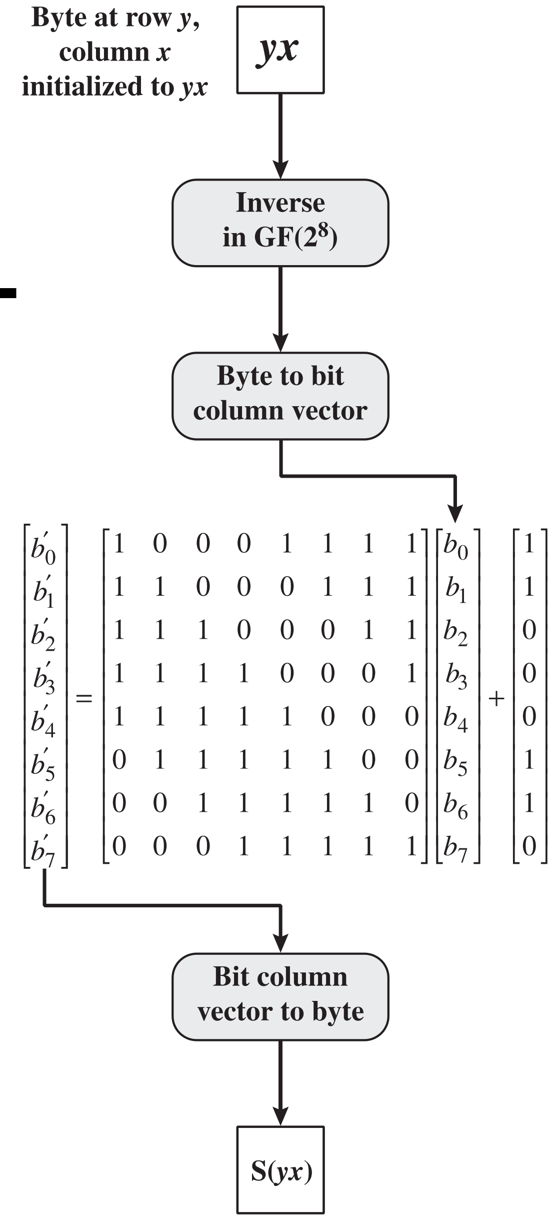
➡ AES S-box: calculation web demo

Example:

- $95_{HEX} \in GF(2^8)$
- 95^{-1}=8A [refer to question pool 2: Q8]
- $8A_{HEX} = 10001010_2$

$$
\begin{bmatrix}
1&0&0&0&1&1&1&1\\
1&1&0&0&0&1&1&1\\
1&1&1&0&0&0&1&1\\
1&1&1&1&0&0&0&1\\
1&1&1&1&1&0&0&0\\
0&1&1&1&1&1&0&0\\
0&0&1&1&1&1&1&0\\
0&0&0&1&1&1&1&1
\end{bmatrix}
\begin{bmatrix}0\\1\\0\\1\\0\\0\\0\\1\end{bmatrix}
\oplus
\begin{bmatrix}1\\1\\0\\0\\0\\1\\1\\0\end{bmatrix}
=
\begin{bmatrix}1\\0\\0\\1\\0\\0\\1\\0\end{bmatrix}
\oplus
\begin{bmatrix}1\\1\\0\\0\\0\\1\\1\\0\end{bmatrix}
=
\begin{bmatrix}0\\1\\0\\1\\0\\1\\0\\0\end{bmatrix}
$$

- S(95)=2A



$$
\begin{bmatrix}b'_0\\b'_1\\b'_2\\b'_3\\b'_4\\b'_5\\b'_6\\b'_7\end{bmatrix}
=
\begin{bmatrix}
1&0&0&0&1&1&1&1\\
1&1&0&0&0&1&1&1\\
1&1&1&0&0&0&1&1\\
1&1&1&1&0&0&0&1\\
1&1&1&1&1&0&0&0\\
0&1&1&1&1&1&0&0\\
0&0&1&1&1&1&1&0\\
0&0&0&1&1&1&1&1
\end{bmatrix}
\begin{bmatrix}b_0\\b_1\\b_2\\b_3\\b_4\\b_5\\b_6\\b_7\end{bmatrix}
+
\begin{bmatrix}1\\1\\0\\0\\0\\1\\1\\0\end{bmatrix}
$$

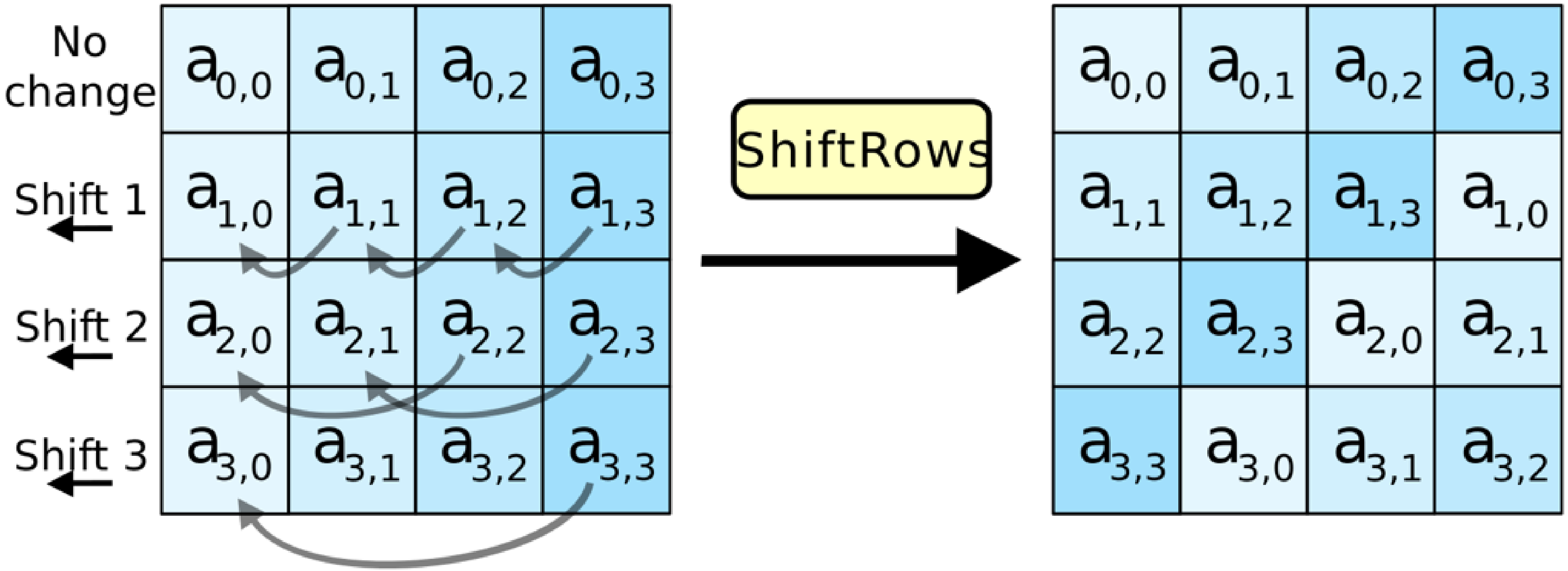
(a) Calculation of byte at row y, column x of S-box

AES S-box: design rationale

- Low correlation between input/output bits
- Output a non-linear function of input
  using multiplicative inverse provides non-linearity
- Constant chosen so that:
  there are no fixed points: S(a)=a
  there are no opposite fixed points: S(a)= $\bar{a}$
  $\bar{a}$ is the bit-wise complement of a
- S-box is invertible, but there are no self-inverses
  S(a)≠IS(a)

AES

ShiftRows Transform

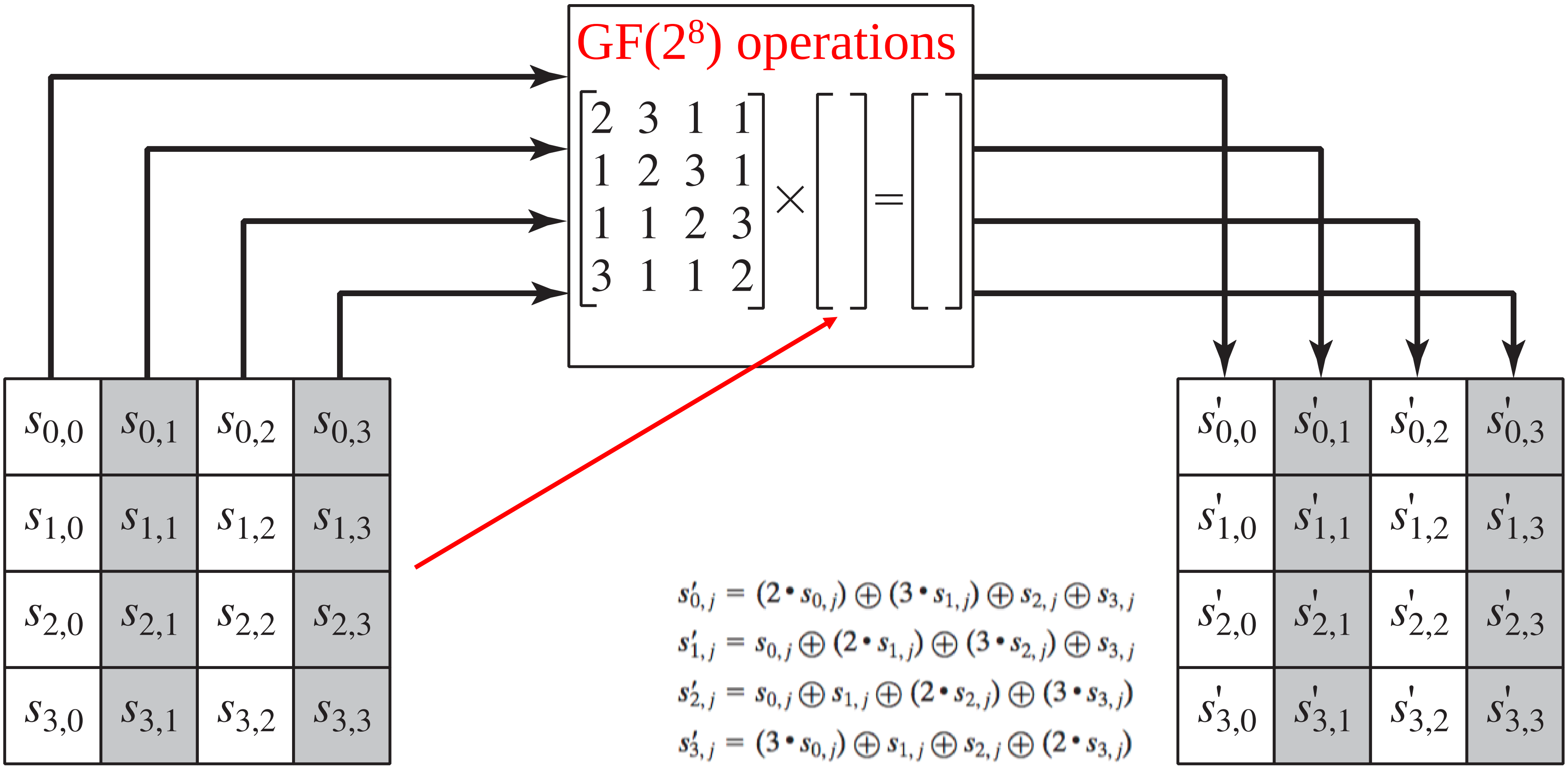


- i^{th} row gets i-1left circular shift

  4 bytes of a column are spread to four different columns

AES

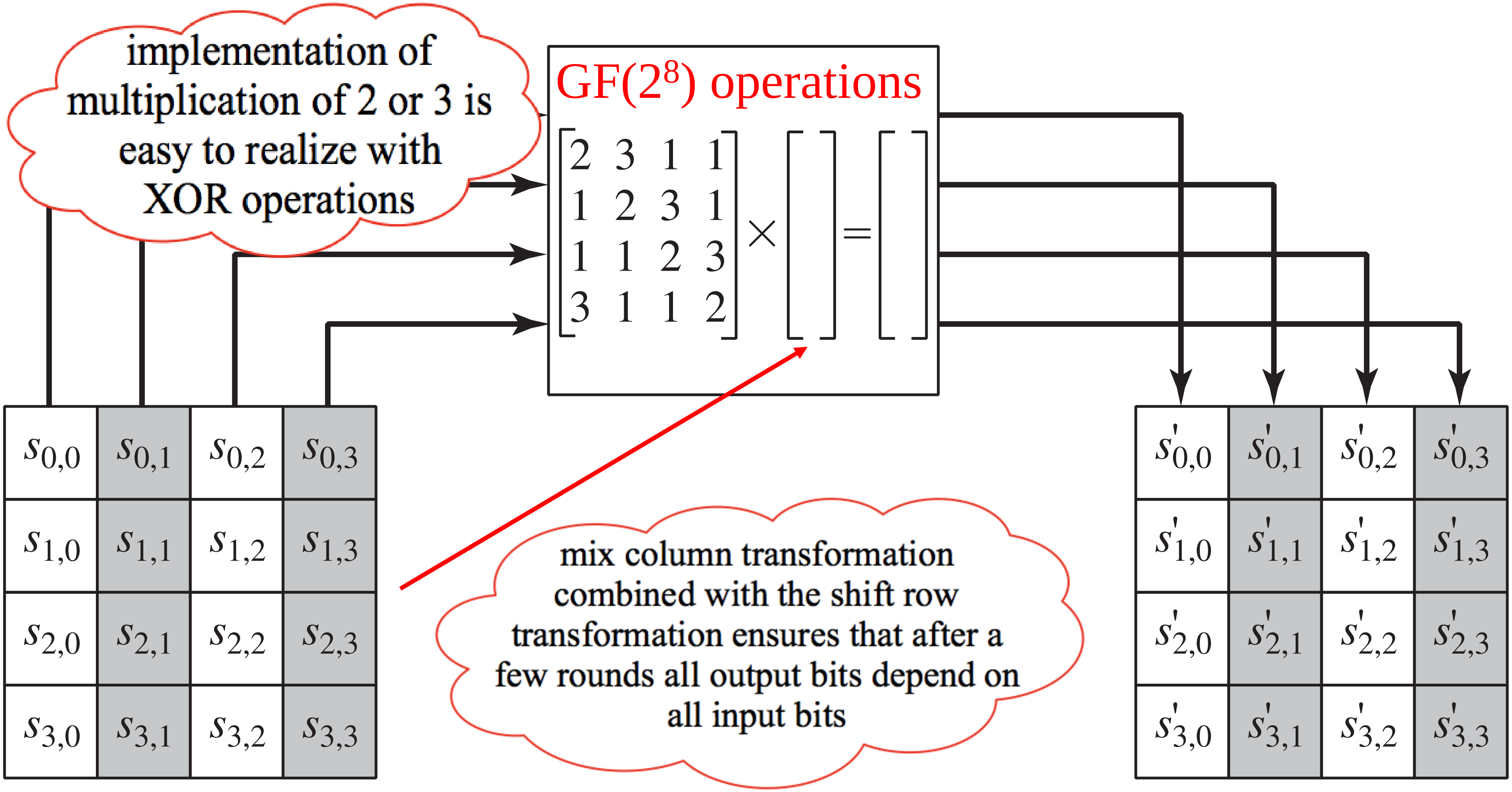
MixColumns Transform



$$s'_{0,j} = (2 \bullet s_{0,j}) \oplus (3 \bullet s_{1,j}) \oplus s_{2,j} \oplus s_{3,j}$$
$$s'_{1,j} = s_{0,j} \oplus (2 \bullet s_{1,j}) \oplus (3 \bullet s_{2,j}) \oplus s_{3,j}$$
$$s'_{2,j} = s_{0,j} \oplus s_{1,j} \oplus (2 \bullet s_{2,j}) \oplus (3 \bullet s_{3,j})$$
$$s'_{3,j} = (3 \bullet s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \bullet s_{3,j})$$

AES

MixColumns Transform



AES

MixColumns Transform: Example

$$
\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}
\begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix}
=
\begin{bmatrix} 47 & 40 & A3 & 4C \\ 37 & D4 & 70 & 9F \\ 94 & E4 & 3A & 42 \\ ED & A5 & A6 & BC \end{bmatrix}
$$

Note (see Section 6.3, Stallings text book, 7ed):

- Multiplication by 2 (is essentially multiplication by x in polynomial representation) can be realized using a 1-bit left shift
- Followed by a conditional XOR with 00011011 if leftmost bit of original value prior to shift is 1.

  Why? Hint: Something to do with the irreducible polynomial …

AES

MixColumns Transform: Example

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

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

=

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

$$(\{02\} \bullet \{87\}) \oplus (\{03\} \bullet \{6E\}) \oplus \{46\} \oplus \{A6\} = \{47\}$$

$$\{02\} \bullet \{87\} = (0000\ 1110) \oplus (0001\ 1011)$$

Note: {87}=1000 0111

$$\{03\} \bullet \{6E\} = \{6E\} \oplus (\{02\} \bullet \{6E\}) = (0110\ 1110) \oplus (1101\ 1100)$$

Note: {6E}=0110 1110

$$\begin{aligned} \{02\} \bullet \{87\} &= 0001\ 0101 \\ \{03\} \bullet \{6E\} &= 1011\ 0010 \\ \{46\} &= 0100\ 0110 \\ \{A6\} &= \underline{1010\ 0110} \\ & 0100\ 0111 = \{47\} \end{aligned}$$

AES

AddRoundKey Transform: Example

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

=

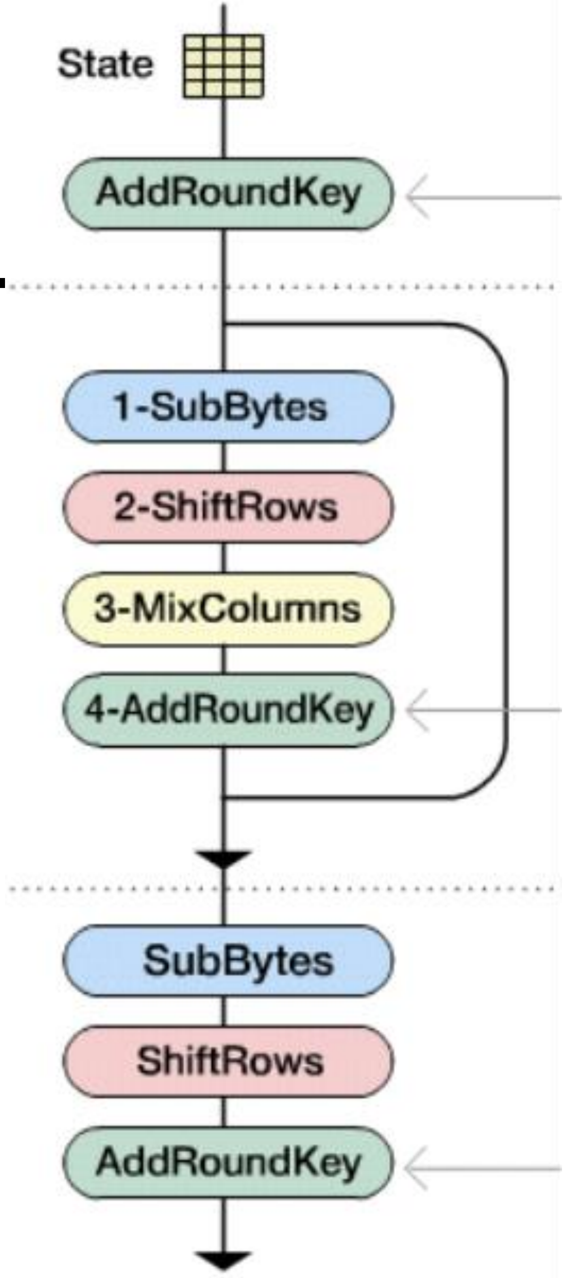
EB	59	8B	1B
40	2E	A1	C3
F2	38	13	42
1E	84	E7	D6

The 128 bits of the state are bitwise XORed with the 128 bits of the round key

AES: Wrap-up

- The cipher begins and ends with an AddRoundKey stage. Why?
  It's in effect a Vernam (one-time pad) cipher

- The other three stages together provide confusion, diffusion and non-linearity, but by themselves, they provide no security.
  Why? No secrets involved in the other steps



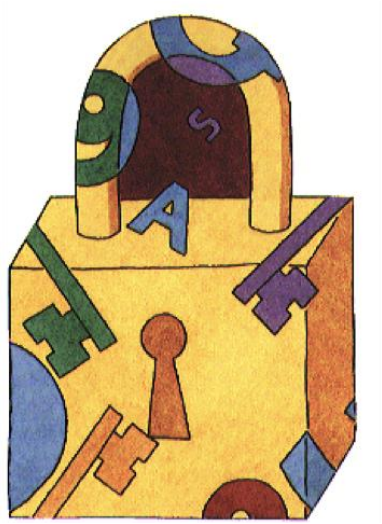
➡ AES-128 web demo

AES

Self study: Sections 6.1 through 6.5 of the (7ed) Stallings textbook

Beyond a block cipher

- Making do with a broken/obsolete cipher
- Encrypting data larger than the block size
- Realizing stream cipher using a block cipher



Someone broke DES, now what?

1998

Electronic Frontier Foundation (**EFF**) breaks DES w/ $250K machine

Someone broke DES, now what?

Use DES multiple times in a cascade!

How many times?

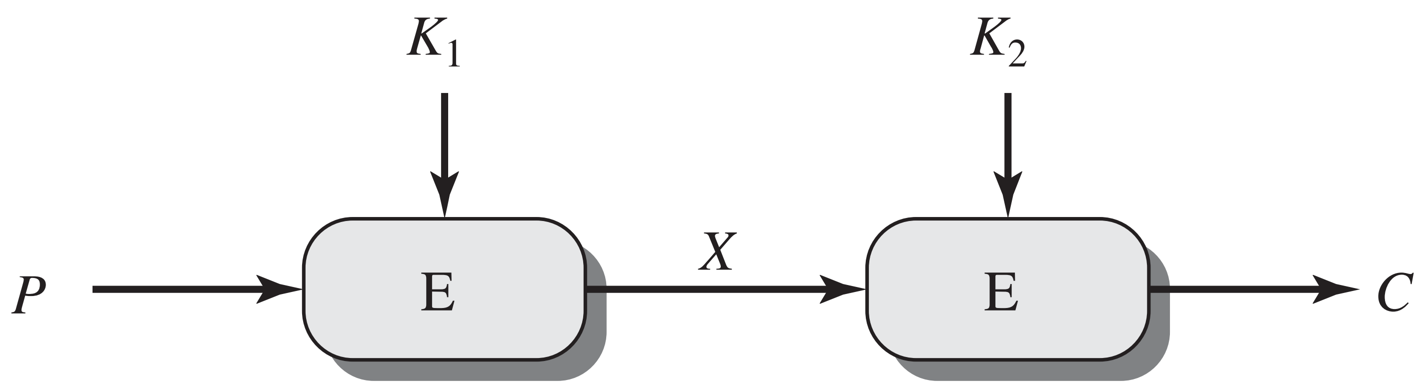
Double DES

Use DES multiple times in a cascade!

How many times?

Use **DES twice**, with two keys

- It "may" help us achieve an effective key size of $2 \times 56 = 112$ bits?



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

Double DES: Two potential issues

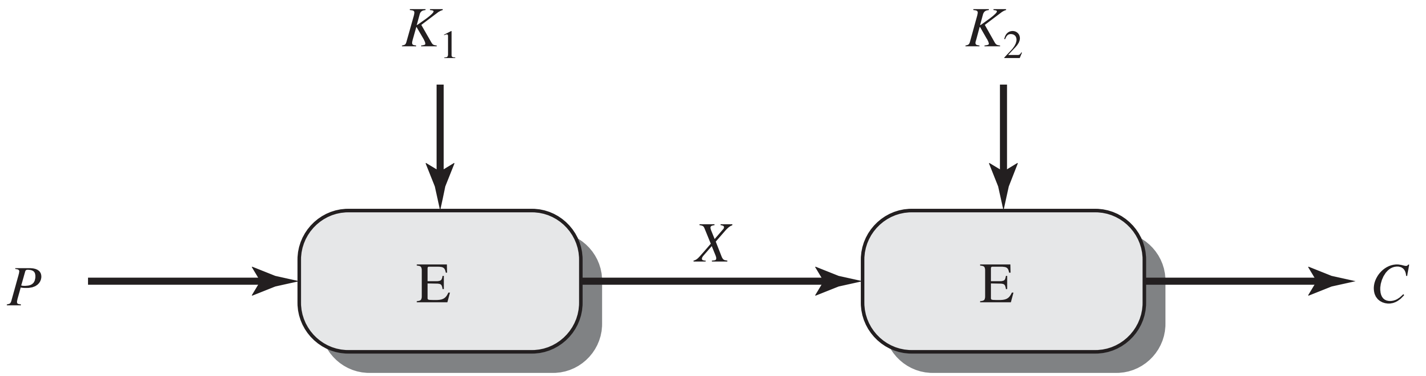
Use **DES twice**, with two keys

- It "may" help us achieve an effective key size of $2 \times 56 = 112$ bits?

Potential issue #1:

- What if: $E(K_2, E(K_1, P)) \equiv E(K_3, P)$

  Turns out not to be of concern!



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

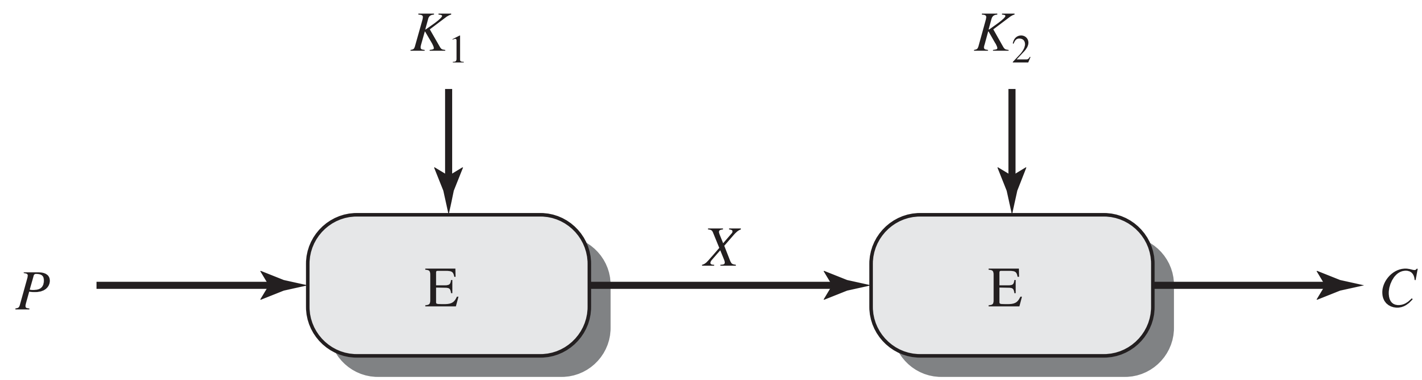
Double DES: Two potential issues

Use **DES twice**, with two keys

- It "may" help us achieve an effective key size of $2 \times 56 = 112$ bits?

Potential issue #2:

- Exploit: $C = E(K_2, E(K_1, P)) \implies \exists X = E(K_1, P) = D(K_2, C)$
  Meet-in-the-middle attack using known plain/cipher-text pairs

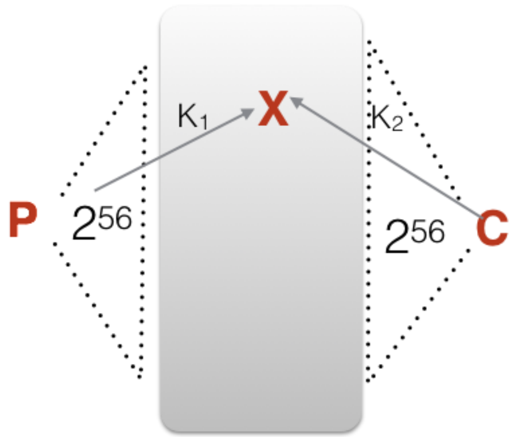


Encryption $C = E(K_2, E(K_1, P))$

Meet-in-the-middle (MITM) attack

A generic attack, but why double DES is not used …

- A known plain-text/cipher-text pair attack
- Encrypt the plain-text with all possible 2^{56} keys
- Likewise, "decrypt" the cipher-text with all possible 2^{56} keys
- Look for matching "X"es
- One such match: 2^{48} false alarms
- Two such matches: Chance of false alarm 2^{-16}



With two pairs of known plain-text/cipher-text, double DES key can be guessed with very high confidence, for roughly same computational complexity as breaking DES itself!

3DES (Triple DES)

Use DES multiple times in a cascade!

How many times?

Use **DES thrice**: third time lucky!

If three stages of DES are used, with three keys:

- Uses a rather long 3*56=168 bits key
- MITM attack cost will be 2^{112}

3DES with two keys:

- Equivalent security (as with 3 keys) against standard MITM

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

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

Not fitting in a block, now what?

1 byte of data comes intermittently, block cipher has128 bits input/128 bits output

.

.

.

1KB data, block cipher has 128 bits input

.

.

.

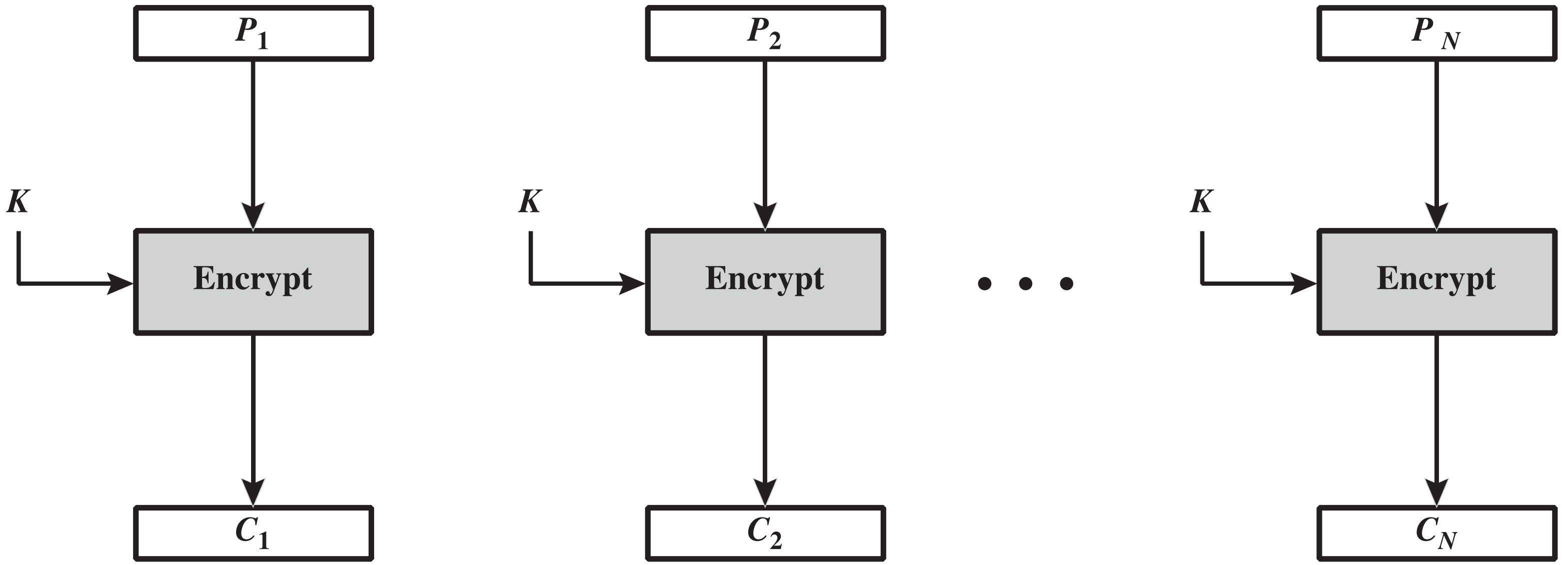
1TB data, block cipher has 128 bits input

Not fitting in a block, now what?

Scenario: Plaintext is larger than block size

- Simplest solution: Just chunk the plaintext and encrypt separately
- This is known as **Electronic Code Book** (ECB)

$$C_j = E(K, P_j)$$

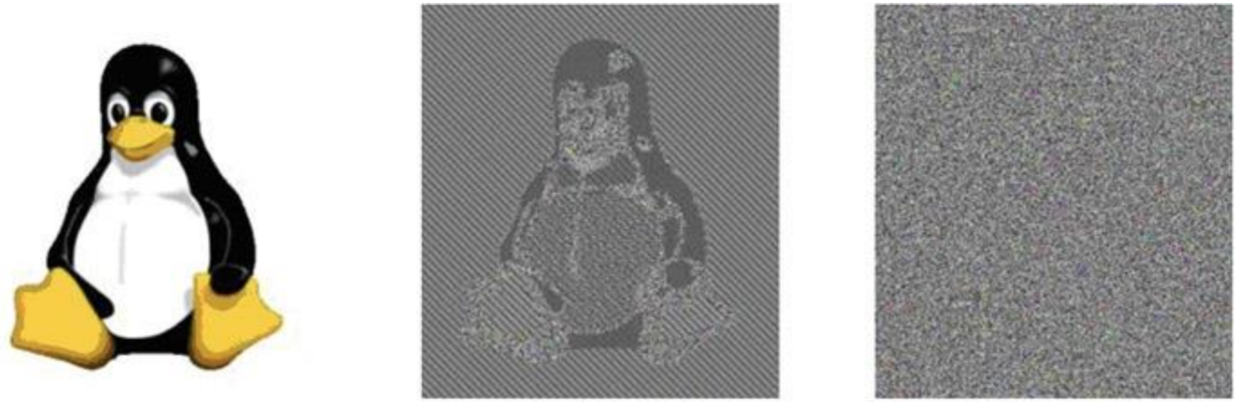


Not fitting in a block, now what?

Scenario: Plaintext is larger than block size

- Simplest solution: Just chunk the plaintext and encrypt separately
- This is known as **Electronic Code Book** (ECB)

  $C_j = E(K, P_j)$
- ECB is good for short messages, but not for large ones (particularly if plain-text is likely to repeat, since then, so will the cipher-text!)

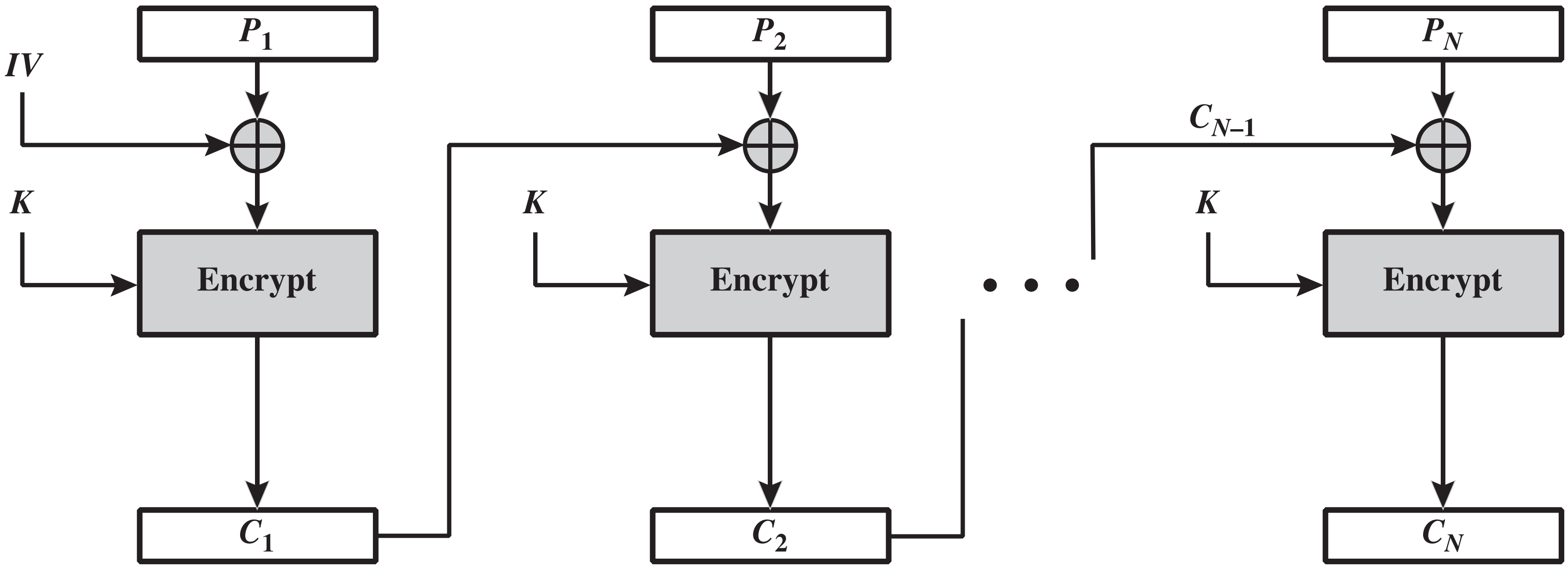


Original Image | Encrypted using ECB mode | Encrypted using other mode

Not fitting in a block, now what?

Cipher block chaining (CBC) $C_j = E(K, C_{j-1} \bigoplus P_j)$

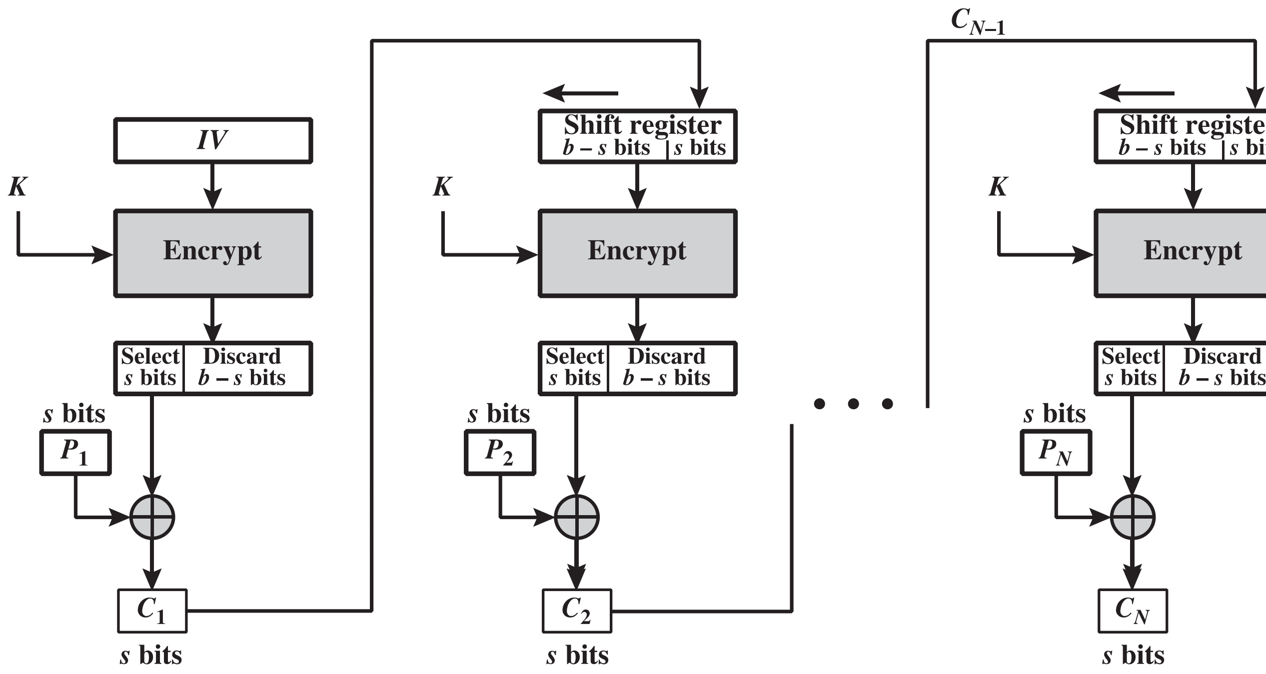
- Design concern: choosing a good initial vector (IV)
- Limitation: No “random access” since decryption is possible only in stages



Stream cipher with a block cipher?

Cipher Feedback Mode (CFB)

- The plain-text is not itself being input to the cipher, but the bit-string being XORed with the plaintext depends on the prior plain-text



- Note: there are other ways to realize a stream cipher using a block cipher

Block cipher operations

Self study: Sections 7.1 through 7.7 of the (7ed) Stallings textbook

- Note: there are several modes of operations not discussed in lectures. (self study)