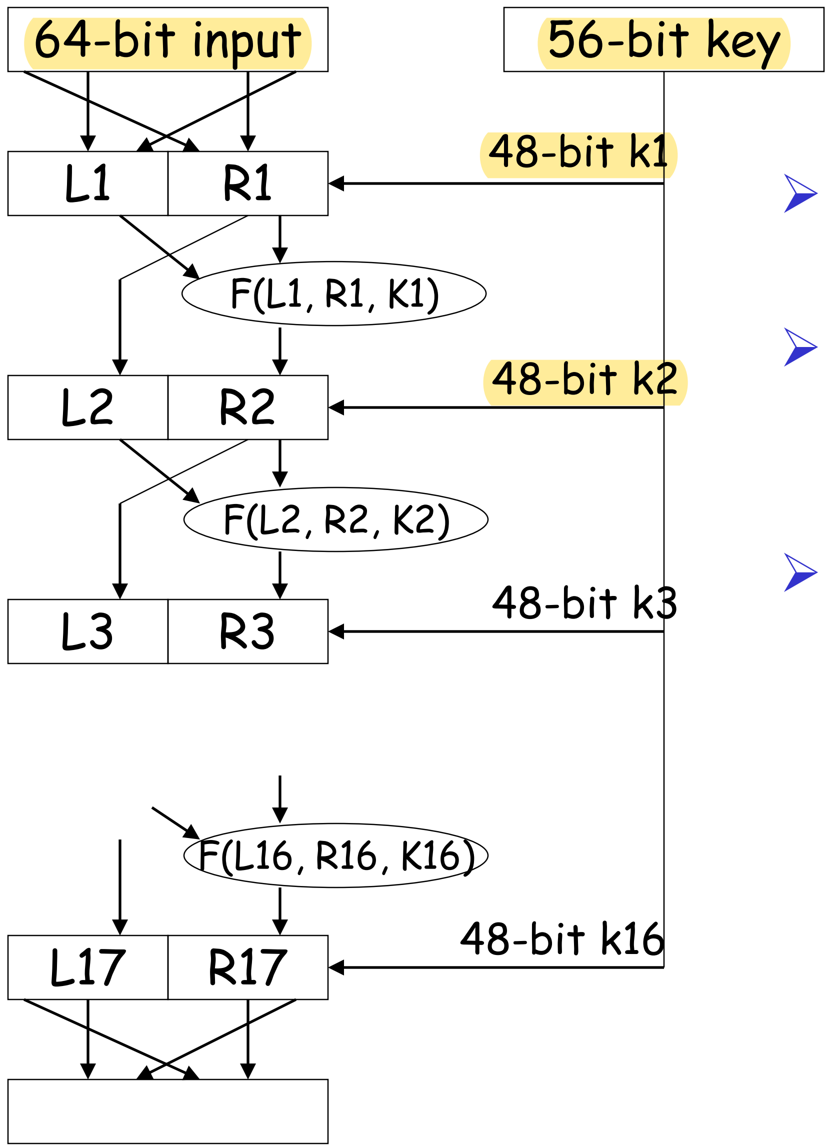
Security and Applications

DES and AES

Data Encryption Standard (DES) Basics



- DES run in reverse to decrypt
- Cracking DES
  - 1997: 140 days
  - 1999: 14 hours
- TripleDES uses DES 3 times in tandem
  - Output from 1 DES is input to next DES

DES Example

- **M** = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
  L = 0000 0001 0010 0011 0100 0101 0110 0111
  R = 1000 1001 1010 1011 1100 1101 1110 1111

- **K** = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001

DES Example: Key Generation

PC-1

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

DES Example: Key Generation

- From the original 64-bit key
- **K** = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001
- we get the 56-bit permutation
- **K+** = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111
- Next, split this K+ into left and right halves,
- $\boldsymbol{C_0}$ = 1111000 0110011 0010101 0101111
  $\boldsymbol{D_0}$ = 0101010 1011001 1001111 0001111

DES Example: Key Generation

Iteration	Shift
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

DES Example: Key Generation

- From original pair pair C_0 and D_0 we obtain:
- C_0 = 1111000011001100101010101111
  D_0 = 0101010101100110011110001111
- C_1 = 1110000110011001010101011111
  D_1 = 1010101011001100111100011110
- C_2 = 1100001100110010101010111111
  D_2 = 0101010110011001111000111101
- C_3 = 0000110011001010101011111111
  D_3 = 0101011001100111100011110101
- C_4 = 0011001100101010101111111100
  D_4 = 0101100110011110001111010101
- C_5 = 1100110010101010111111110000
  D_5 = 0110011001111000111101010101
- C_6 = 0011001010101011111111000011
  D_6 = 1001100111100011110101010101

C_7 = 1100101010101111111100001100
D_7 = 0110011110001111010101010110
C_8 = 0010101010111111110000110011
D_8 = 1001111000111101010101011001
C_9 = 0101010101111111100001100110
D_9 = 0011110001111010101010110011
C_{10} = 0101010111111110000110011001
D_{10} = 1111000111101010101011001100
C_{11} = 0101011111111000011001100101
D_{11} = 1100011110101010101100110011
C_{12} = 0101111111100001100110010101
D_{12} = 0001111010101010110011001111
C_{13} = 0111111110000110011001010101
D_{13} = 0111101010101011001100111100
C_{14} = 1111111000011001100101010101
D_{14} = 1110101010101100110011110001
C_{15} = 1111100001100110010101010111
D_{15} = 1010101010110011001111000111
C_{16} = 1111000011001100101010101111
D_{16} = 0101010101100110011110001111

DES Example: 48 bit Key Generation K_n

PC-2

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

DES Example: 48 bit Key Generation K_n

- K_1 = 000110 110000 001011 101111 111111 000111 000001 110010
- For the other keys we have
- K_2 = 011110 011010 111011 011001 110110 111100 100111 100101
  K_3 = 010101 011111 110010 001010 010000 101100 111110 011001
  K_4 = 011100 101010 110111 010110 110110 110011 010100 011101
  K_5 = 011111 001110 110000 000111 111010 110101 001110 101000
  K_6 = 011000 111010 010100 111110 010100 000111 101100 101111
  K_7 = 111011 001000 010010 110111 111101 100001 100010 111100
  K_8 = 111101 111000 101000 111010 110000 010011 101111 111011
  K_9 = 111000 001101 101111 101011 111011 011110 011110 000001
  K_{10} = 101100 011111 001101 000111 101110 100100 011001 001111
  K_{11} = 001000 010101 111111 010011 110111 101101 001110 000110
  K_{12} = 011101 010111 000111 110101 100101 000110 011111 101001
  K_{13} = 100101 111100 010111 010001 111110 101011 101001 000001
  K_{14} = 010111 110100 001110 110111 111100 101110 011100 111010
  K_{15} = 101111 111001 000110 001101 001111 010011 111100 001010
  K_{16} = 110010 110011 110110 001011 000011 100001 011111 110101

DES Example: Encode M

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

DES Example: Encode M

- Applying IP to M, we get
- **M** = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
  M’= 1100 1100 0000 0000 1100 1100 1111 1111 1111 0000 1010 1010 1111 0000 1010 1010

- From **M’**, we get $\boldsymbol{L_0}$ and $\boldsymbol{R_0}$
- $\boldsymbol{L_0}$ = 1100 1100 0000 0000 1100 1100 1111 1111
  $\boldsymbol{R_0}$ = 1111 0000 1010 1010 1111 0000 1010 1010

DES Example: Application of K_n

$\boldsymbol{L_n} = \boldsymbol{R_{n-1}}$

$\boldsymbol{R_n} = \boldsymbol{L_{n-1}} + \boldsymbol{f}(\boldsymbol{R_{n-1}}, \boldsymbol{K_n})$

Example

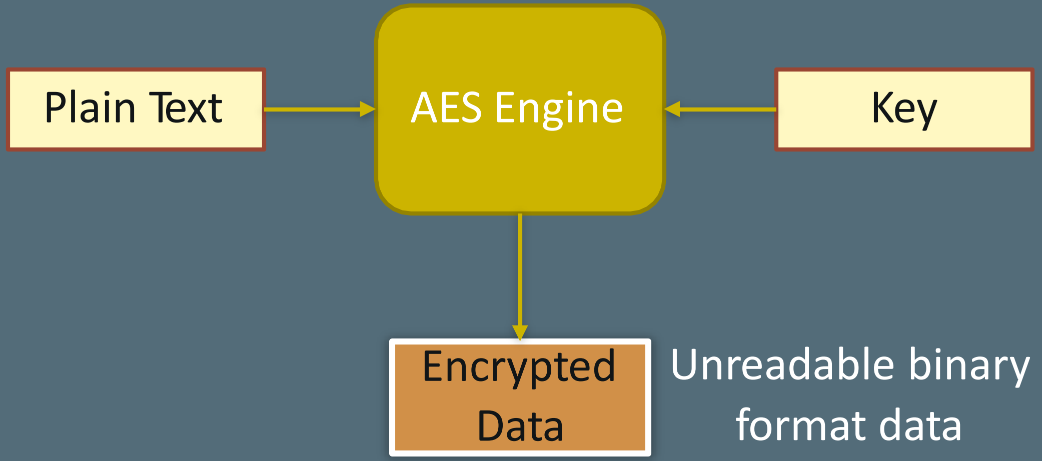
$\boldsymbol{K_1}$ = 000110 110000 001011 101111 111111 000111 000001 110010

$\boldsymbol{L_1} = \boldsymbol{R_0}$ = 1111 0000 1010 1010 1111 0000 1010 1010

$\boldsymbol{R_1} = \boldsymbol{L_0} + \boldsymbol{f}(\boldsymbol{R_0}, \boldsymbol{K_1})$

HOW AES Works?

- Requirements:
  - Software that implements the AES Algorithm
  - Inputs: Data (credit card number, plain text) and Key (encryption key)



Reversal for decryption

AES-Cipher

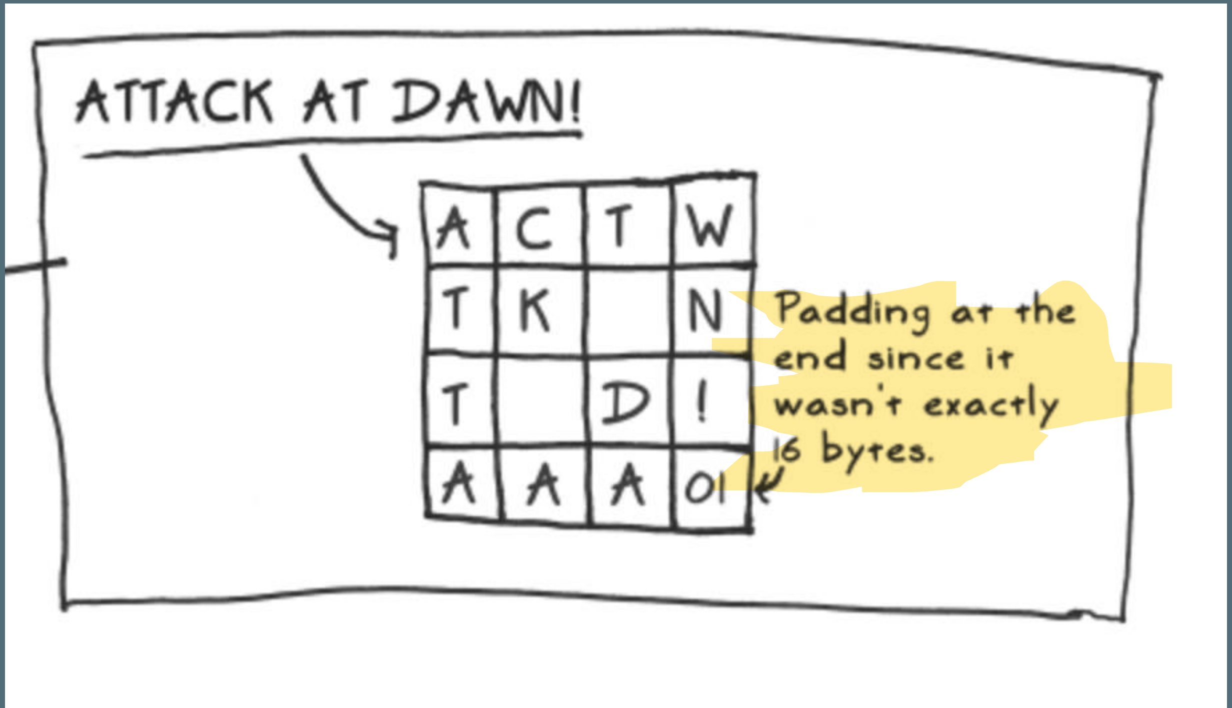
- AES is a block cipher

- Size of the block is 16 bytes

- Encryption Key Sizes:
  - 128 bit (16 Bytes)
  - 192 bit (24 Bytes)
  - 256 bit (32 Bytes)

AES Functionality

The plain text Message



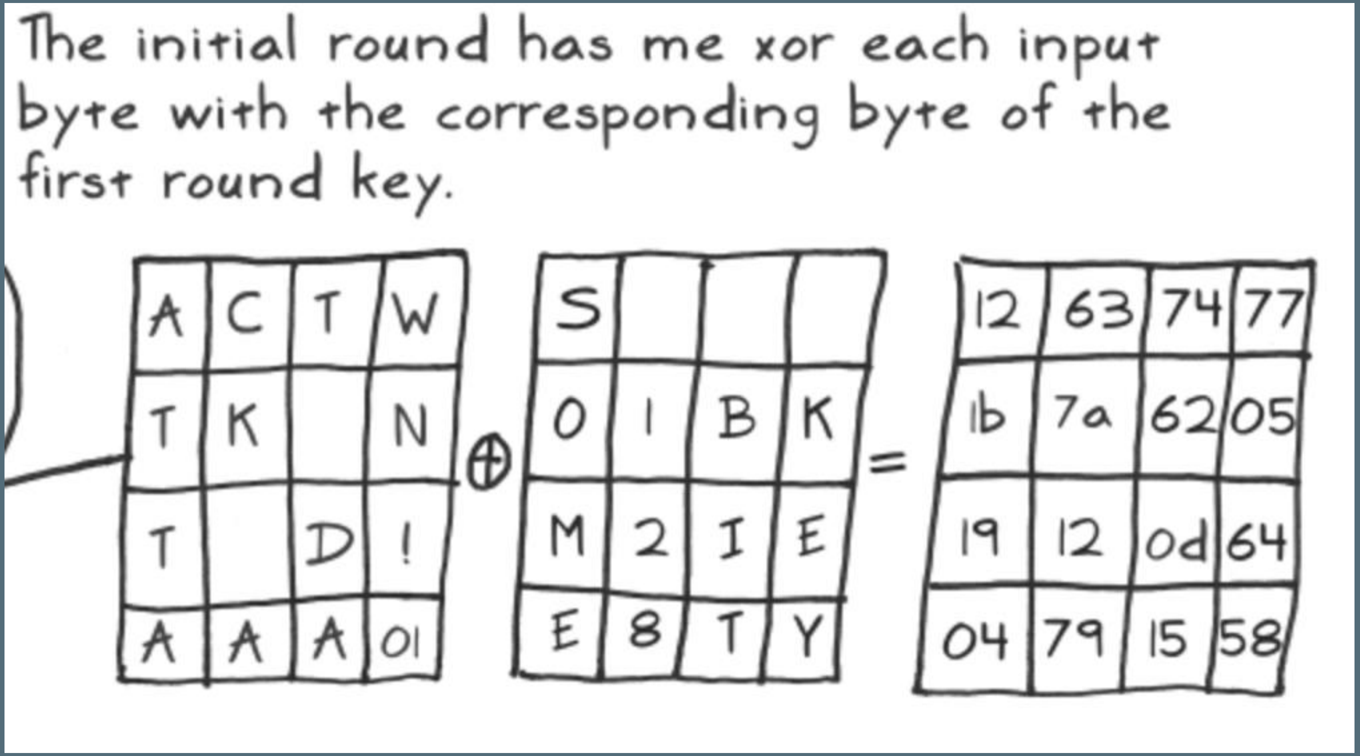
ATTACK AT DAWN!

A	C	T	W
T	K		N
T		D	!
A	A	A	01

Padding at the end since it wasn't exactly 16 bytes.

Encryption

Initial encryption with Key (128bit)



The initial round has me xor each input byte with the corresponding byte of the first round key.

A	C	T	W
T	K		N
T		D	!
A	A	A	01

⊕

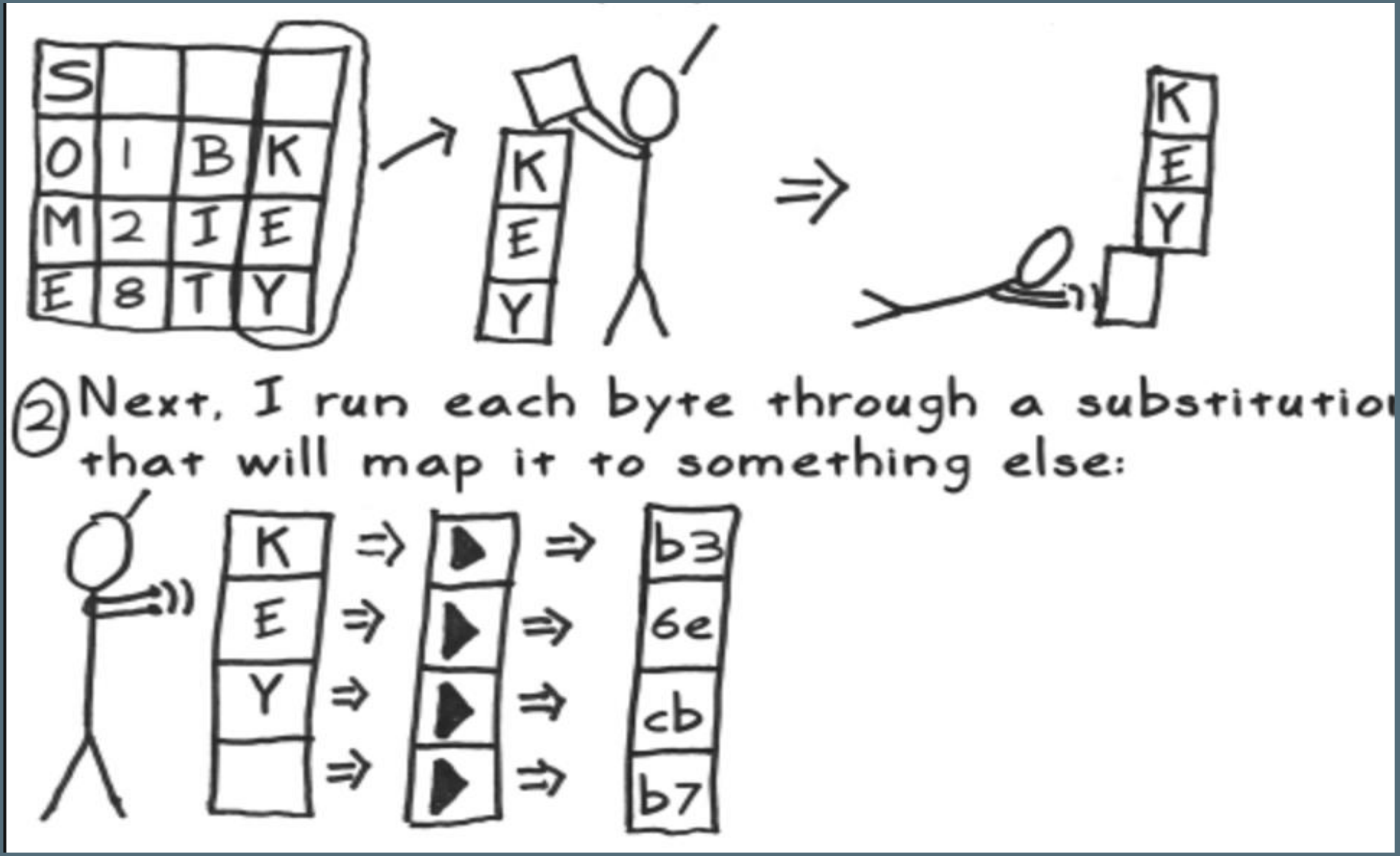
S			
O	1	B	K
M	2	I	E
E	8	T	Y

=

12	63	74	77
1b	7a	62	05
19	12	0d	64
04	79	15	58

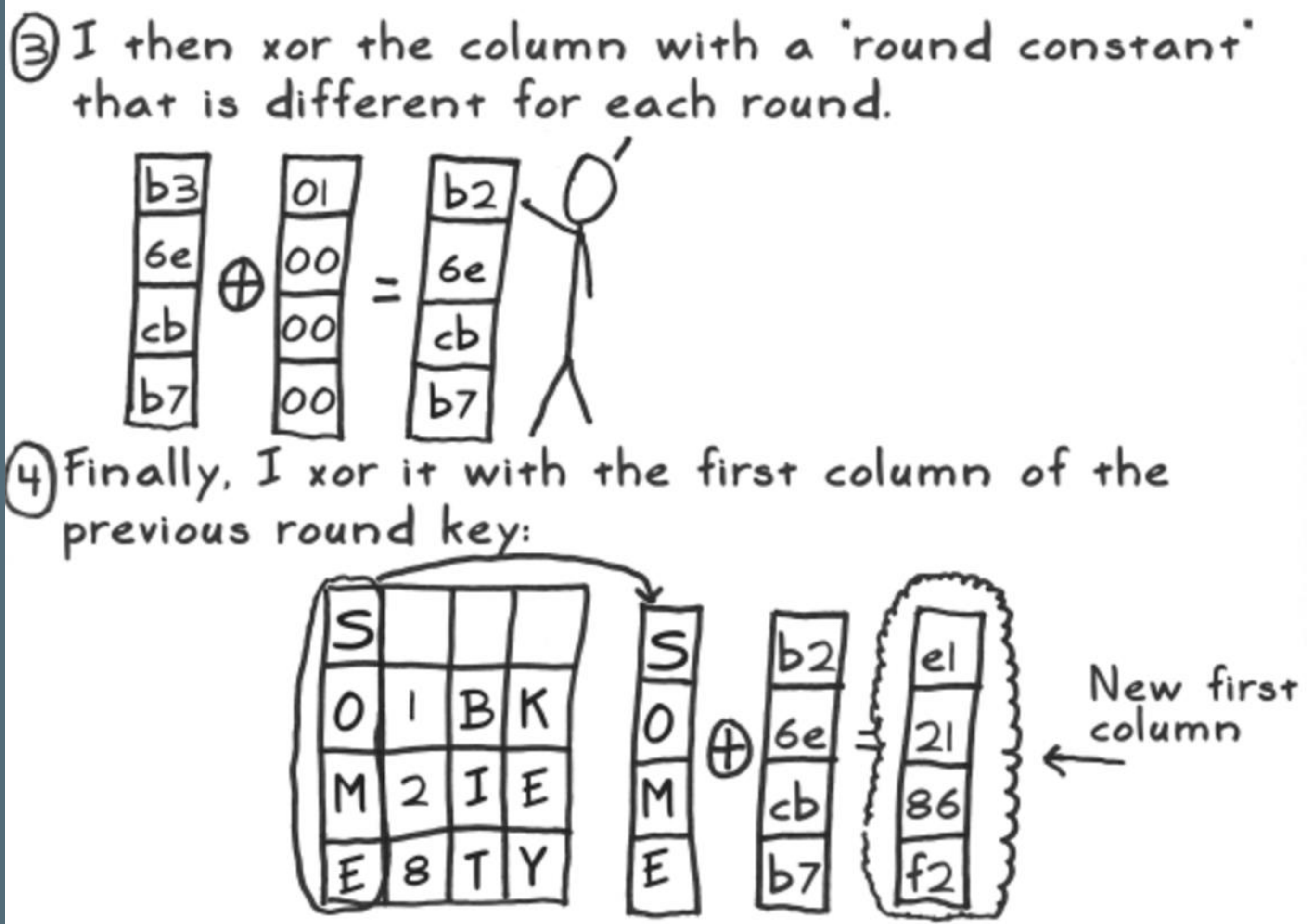
Encryption

Substitution technique for other keys


S
O 1 B K
M 2 I E
E 8 T Y
K
E
Y
K
E
Y
②Next, I run each byte through a substitutio
that will map it to something else:
K
E
Y
b3
6e
cb
b7

Keep Iterating

Substitution technique for other keys



③ I then xor the column with a 'round constant' that is different for each round.

④ Finally, I xor it with the first column of the previous round key:

S			
O	1	B	K
M	2	I	E
E	8	T	Y

Encryption Algorithm

Summary

Algorithm	Type	Key Size	Features
DES	Block Cipher	56 bits	Most Common, Not strong enough
TripleDES	Block Cipher	168 bits (112 effective)	Modification of DES, Adequate Security
Blowfish	Block Cipher	Variable (Up to 448 bits)	Excellent Security
AES	Block Cipher	Variable (128, 192, or 256 bits)	Replacement for DES, Excellent Security
RC4	Stream Cipher	Variable (40 or 128 bits)	Fast Stream Cipher, Used in most SSL implementations