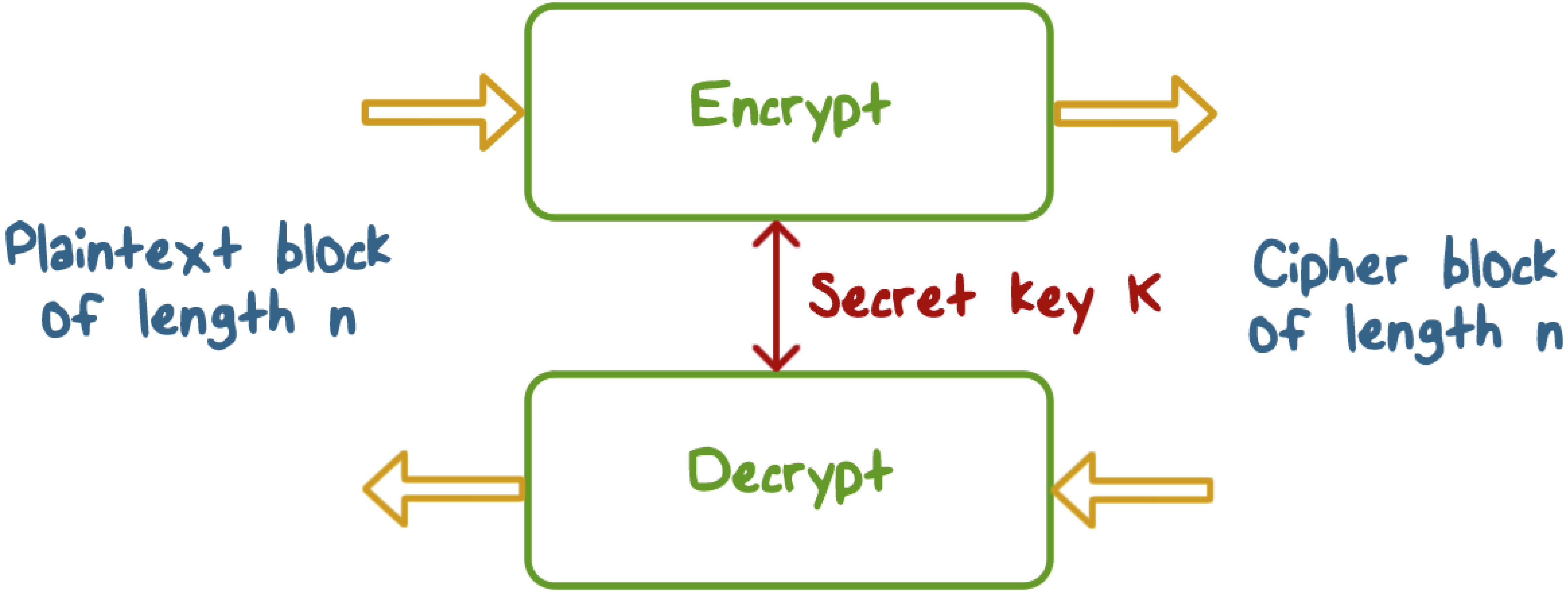
Symmetric Encryption

Lesson Introduction

- Block cipher primitives
- DES
- AES
- Encrypting large message
- Message integrity

Block Cipher Scheme



Block Cipher Primitives

Confusion:

●An encryption operation where the relationship between the key and ciphertext is obscured

•Achieved with **substitution**

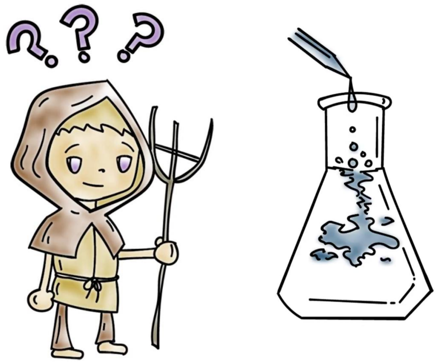
Block Cipher Primitives

Diffusion:

- An encryption operation where the influence of one plaintext bit is spread over many ciphertext bits with the goal of hiding statistical properties of the plaintext

  - Achieved with **permutation**

Block Cipher Primitives



- Both confusion and diffusion by themselves **cannot provide (strong enough) security**

- **Round:** combination of substitution and permutation, and do so often enough so that a bit change can affect every output bit

Block Cipher Quiz

Select all correct answers to complete that statement.

A block cipher should...

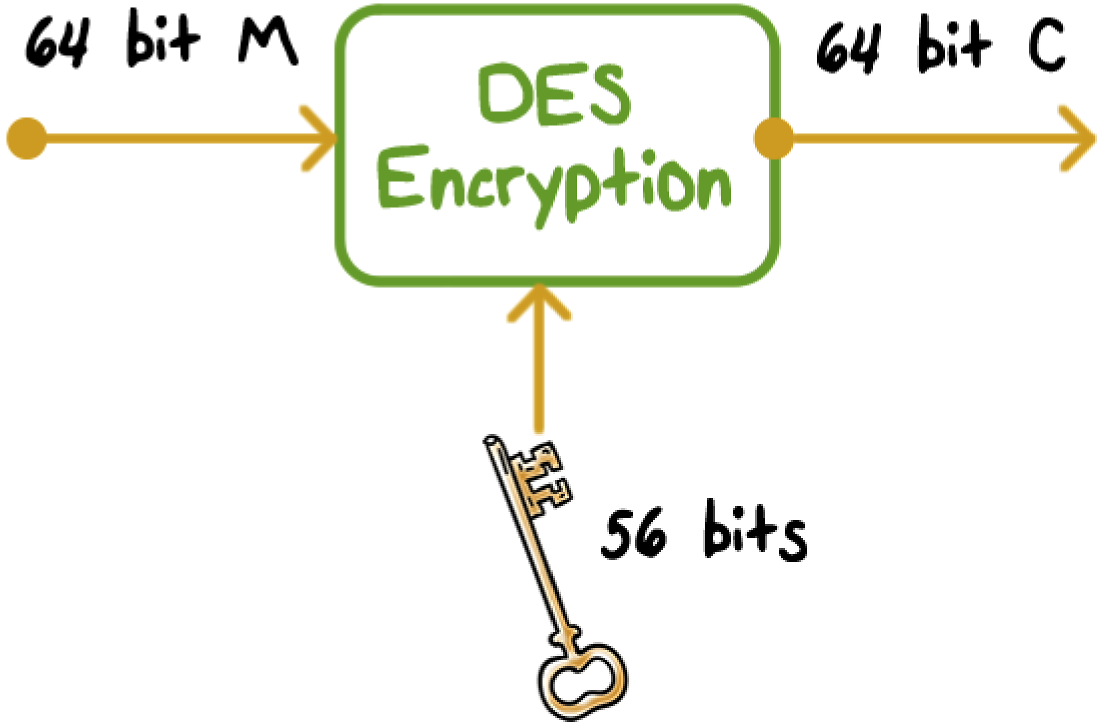
□ Use substitution to achieve confusion

□ Use permutation to achieve diffusion

□ Use a few rounds, each with a combination of substitution and permutation

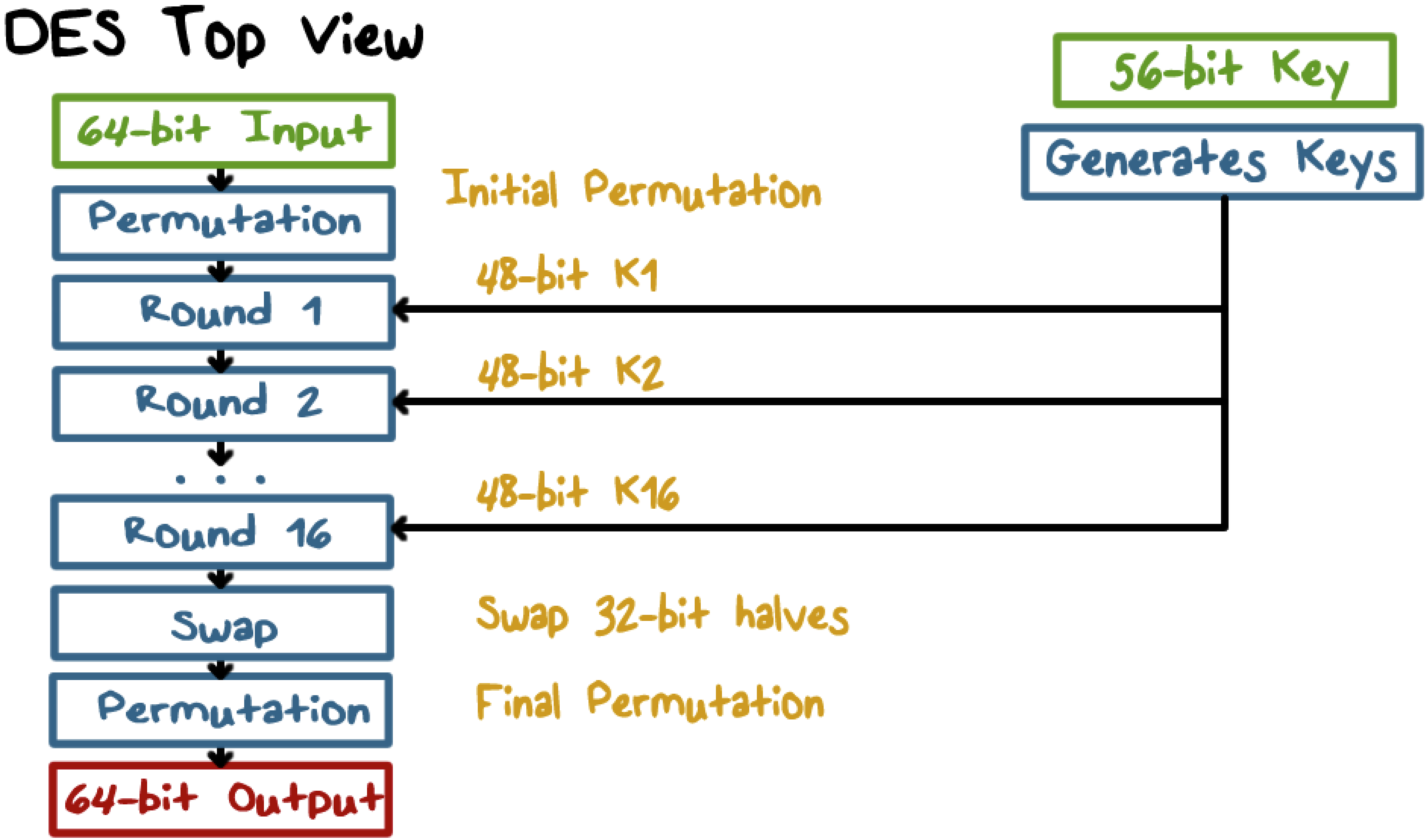
□ Keep the algorithm secret

Data Encryption Standard



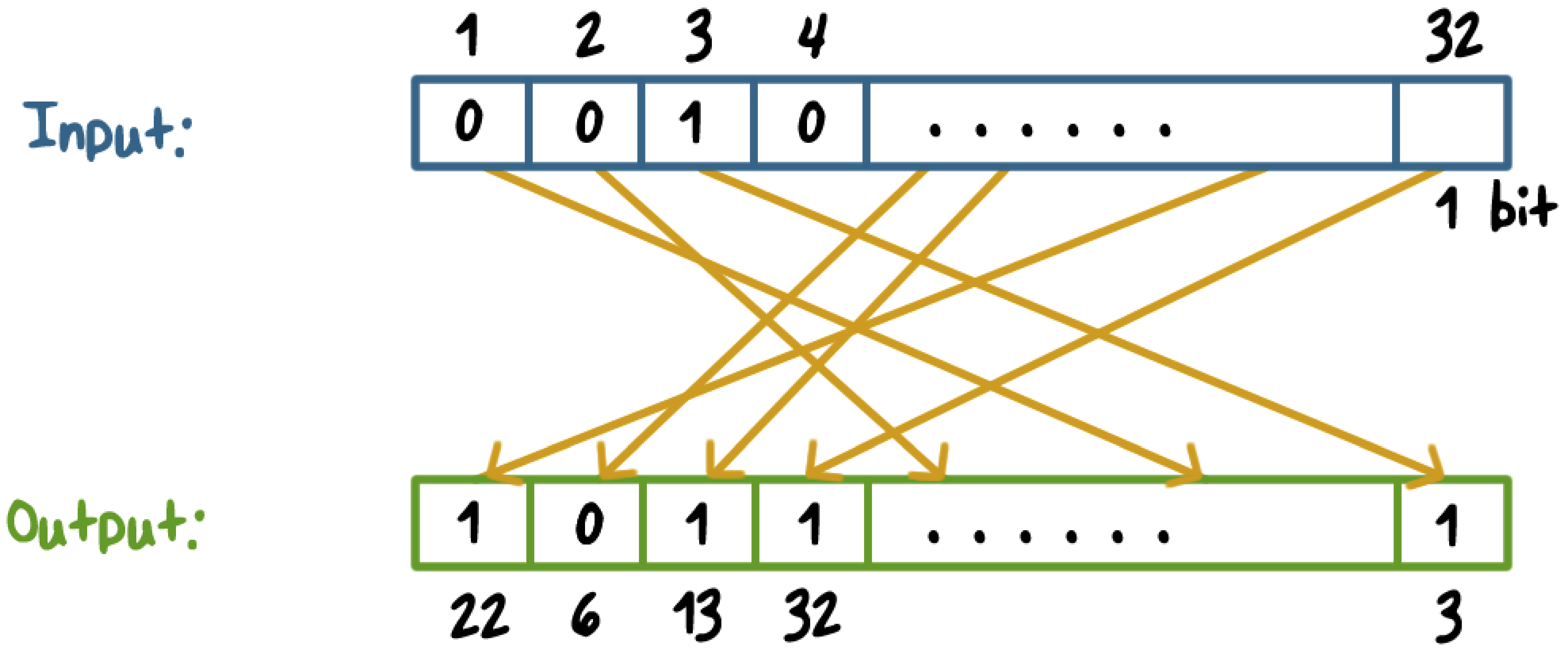
- Published in 1977, standardized in 1979
- **Key:** 64 bit quantity=8-bit parity+56-bit key
  - Every 8th bit is a parity bit
- 64 bit input, 64 bit output

Data Encryption Standard



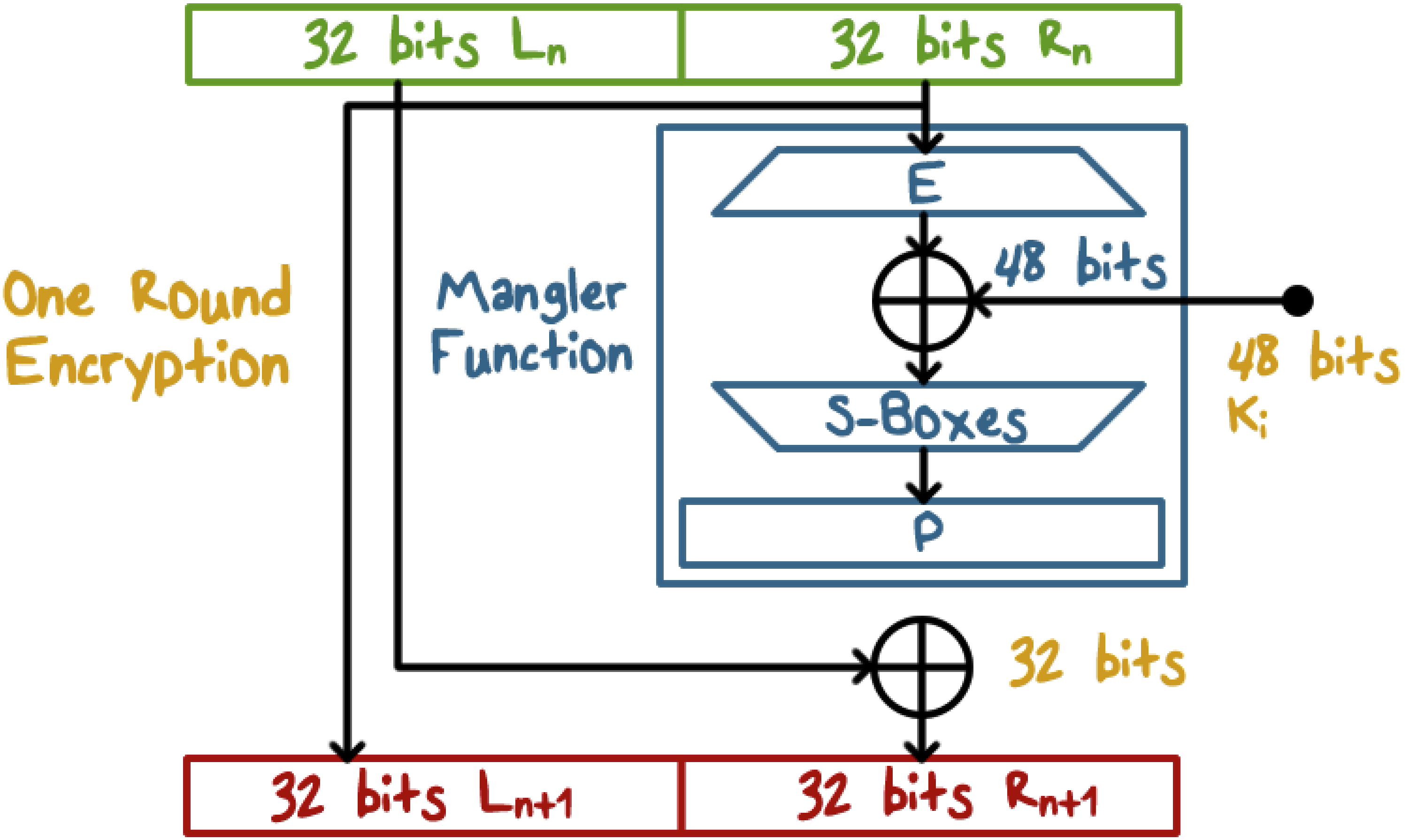
Data Encryption Standard

Bit Permutation (1-to-1)



Data Encryption Standard

A DES Round

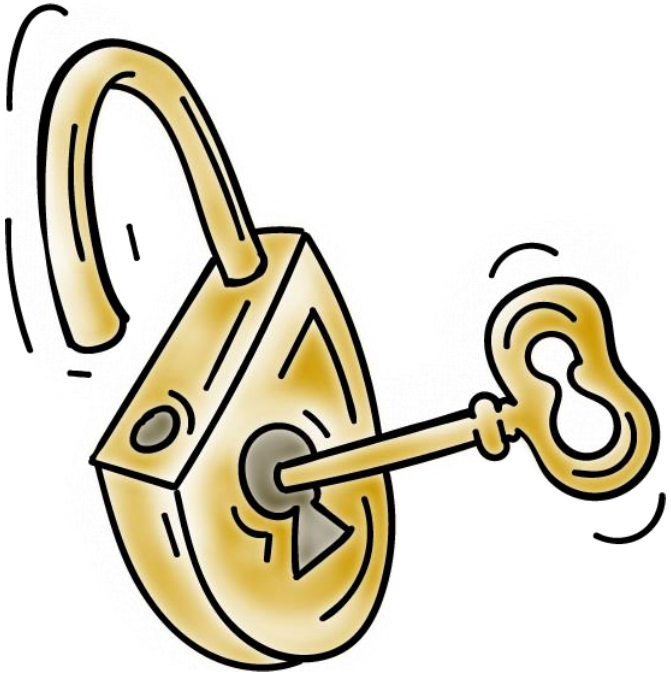


- Can be expressed as:

  $L_{n+1}=R_n$

  $R_{n+1}=L_n \text{ XOR } M(R_n,K_n)$

Decryption



- **Apply the same operations key sequence in reverse:**
  - Round 1 of decryption uses key of the last round in encryption
- Each round:
  - **Input:** $R_{n+1}|L_{n+1}$
    - Due to the swap operation at the end of encryption
  - **Output:** $R_n|L_n$
- The swap operation at the end will produce the correct result: L|R

XOR Quiz

Use the XOR function and the given key to encrypt the word “Hi”.

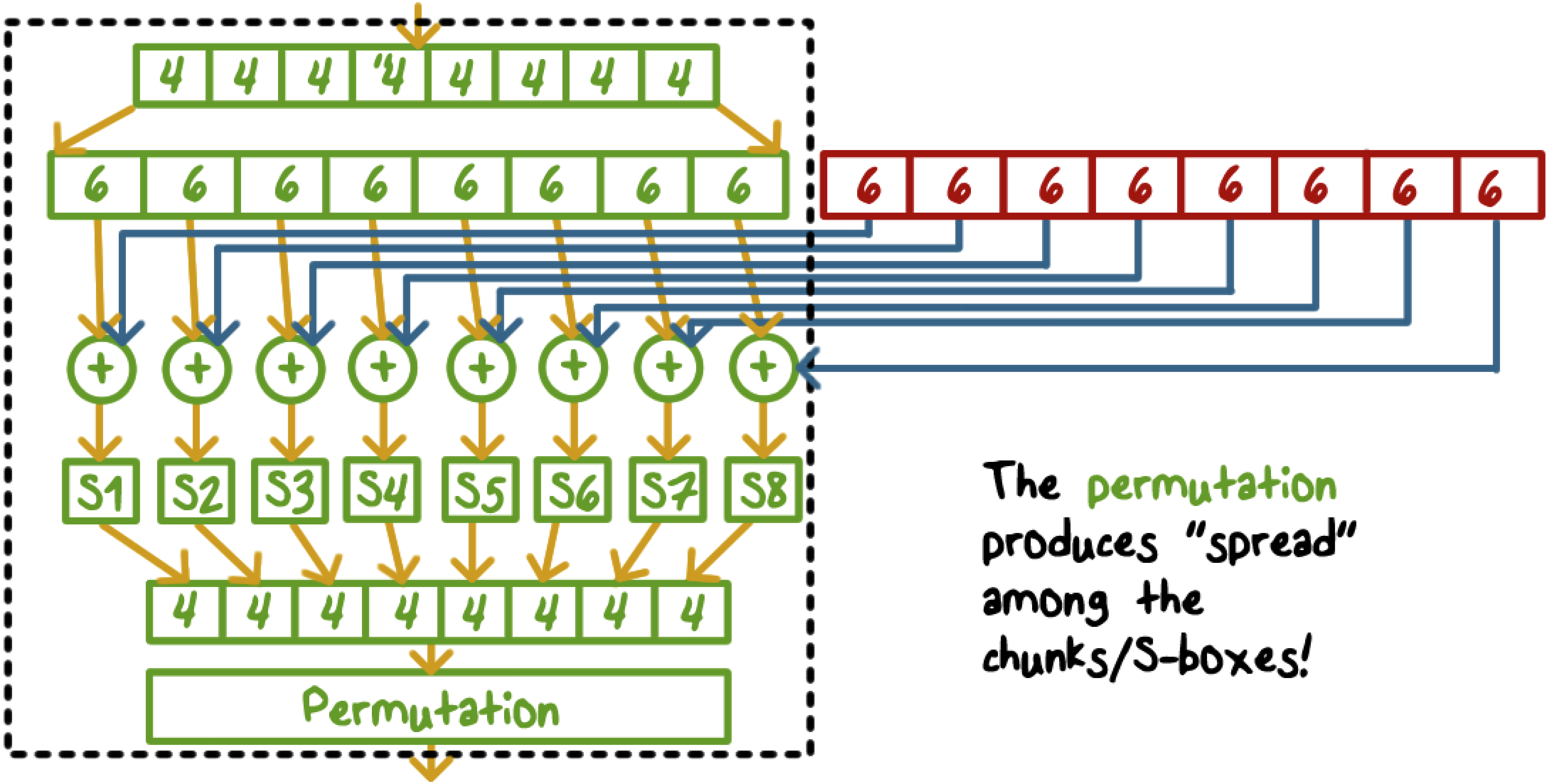
key = FA F2

Hi = □□□□ □□□□ □□□□ □□□□

FA F2 = □□□□ □□□□ □□□□ □□□□

Hi encrypted = □□□□ □□□□ □□□□ □□□□

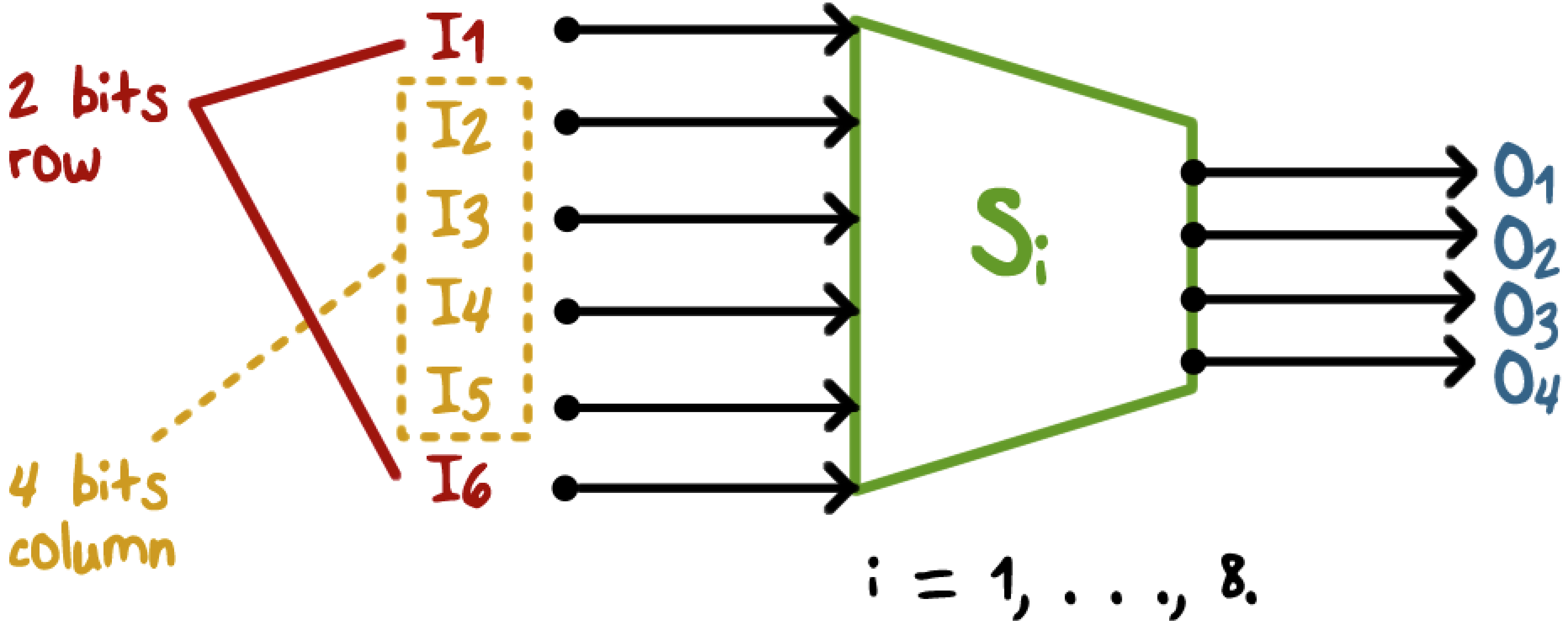
Mangler Function



The permutation produces "spread" among the chunks/S-boxes!

S-Box (Substitute and Shrink)

- 48 bits => 32 bits. (8*6 => 8*4)
- 2 bits used to select amongst 4 substitutions for the rest of the 4-bit quantity



S-Box Quiz

For the given input, determine the output.

S_5		Middle 4 bits of input															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Outer bits	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

Input:

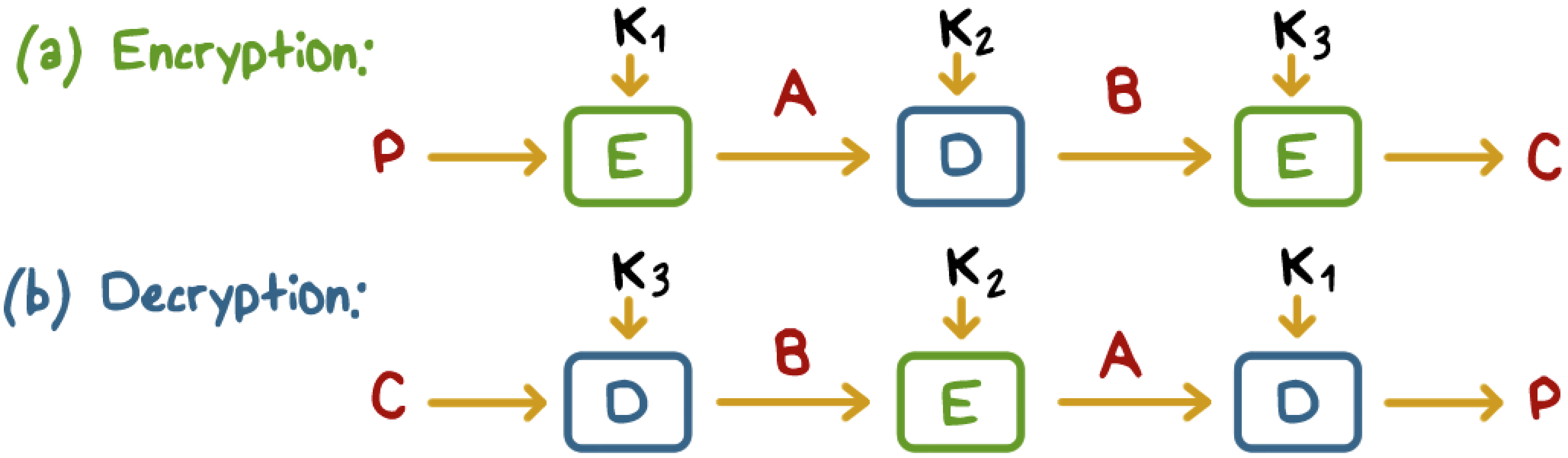
011011

Output:

Security of DES

- **Key space is too small** (2^{56} keys)
  - Exhaustive key search relative easy with today’s computers
- **S-box design criteria have been kept secret**
- **Highly resistant** to cryptanalysis techniques published years after DES

Triple DES



- K_1=K_3 results in an equivalent 112-bit DES which provides a sufficient key space
- Distinct K_1, K_2, K_3 results in an even stronger 168-bit DES
- Can run as a single DES with K_1 = K_2

DES Quiz

Check all the statements that are true:

□ To decrypt using DES, same algorithm is used, but with per-round keys used in the reversed order

□ With Triple DES the effective key length can be 56, 112, and 168

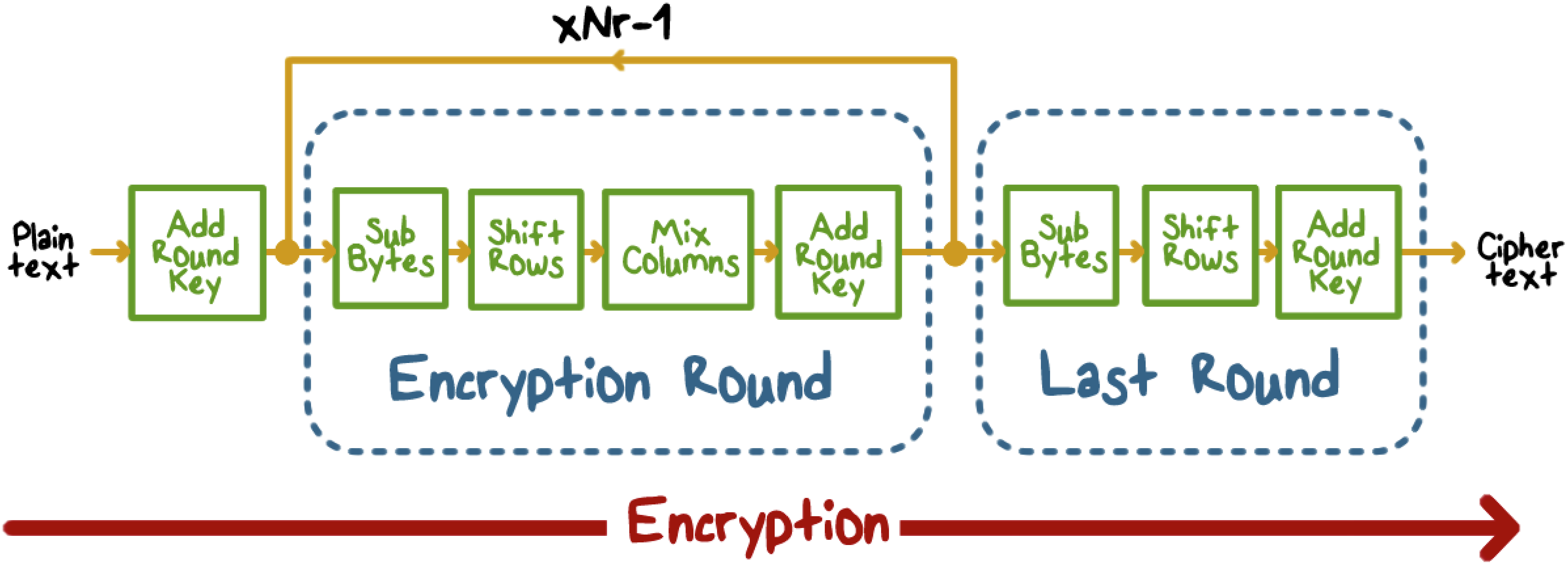
□ Each round of DES contains both substitution and permutation operations

□ The logics behind the S-boxes are well-known and verified

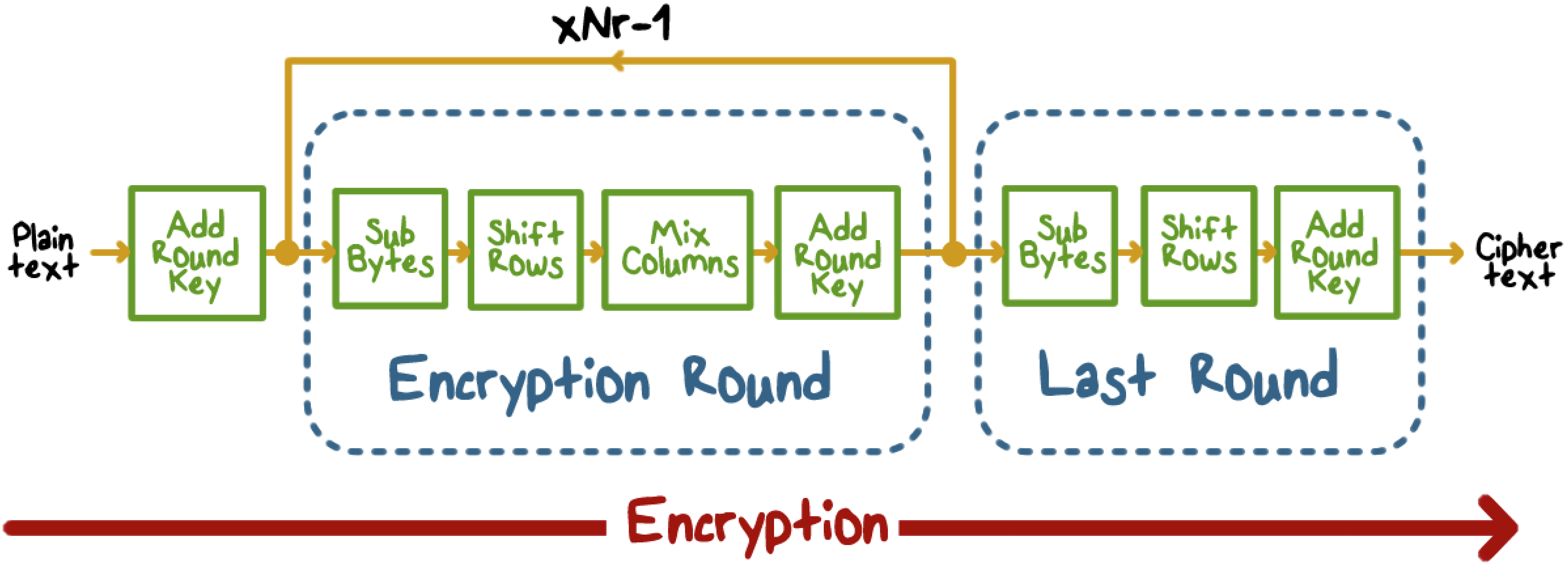
Advanced Encryption Standard

- In 1997, the **U.S. National Institute for Standards and Technology (NIST)** put out a public call for a replacement to DES
- It narrowed down the list of submissions to five finalists, and ultimately chose an algorithm (Rijndael) that is now known as the **Advanced Encryption Standard** (**AES**)
- New (Nov. 2001) symmetric-key NIST standard, replacing DES
- **Processes data in 128 bit blocks**
- **Key length can be 128, 192, or 256 bits**

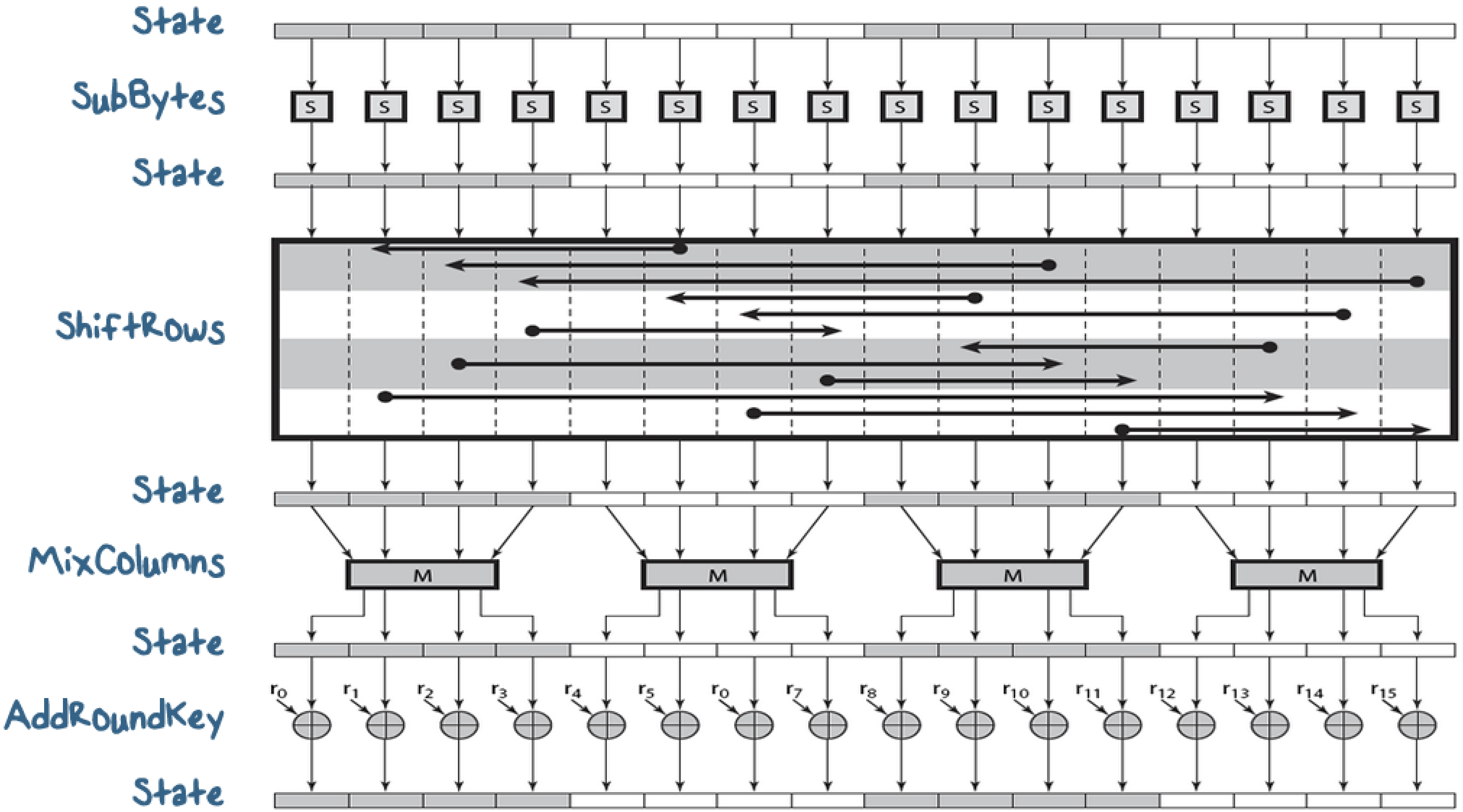
Advanced Encryption Standard



AES Round


xNr-1
Plain text
Add Round Key
Sub Bytes
Shift Rows
Mix Columns
Add Round Key
Encryption Round
Sub Bytes
Shift Rows
Add Round Key
Last Round
Cipher text
Encryption

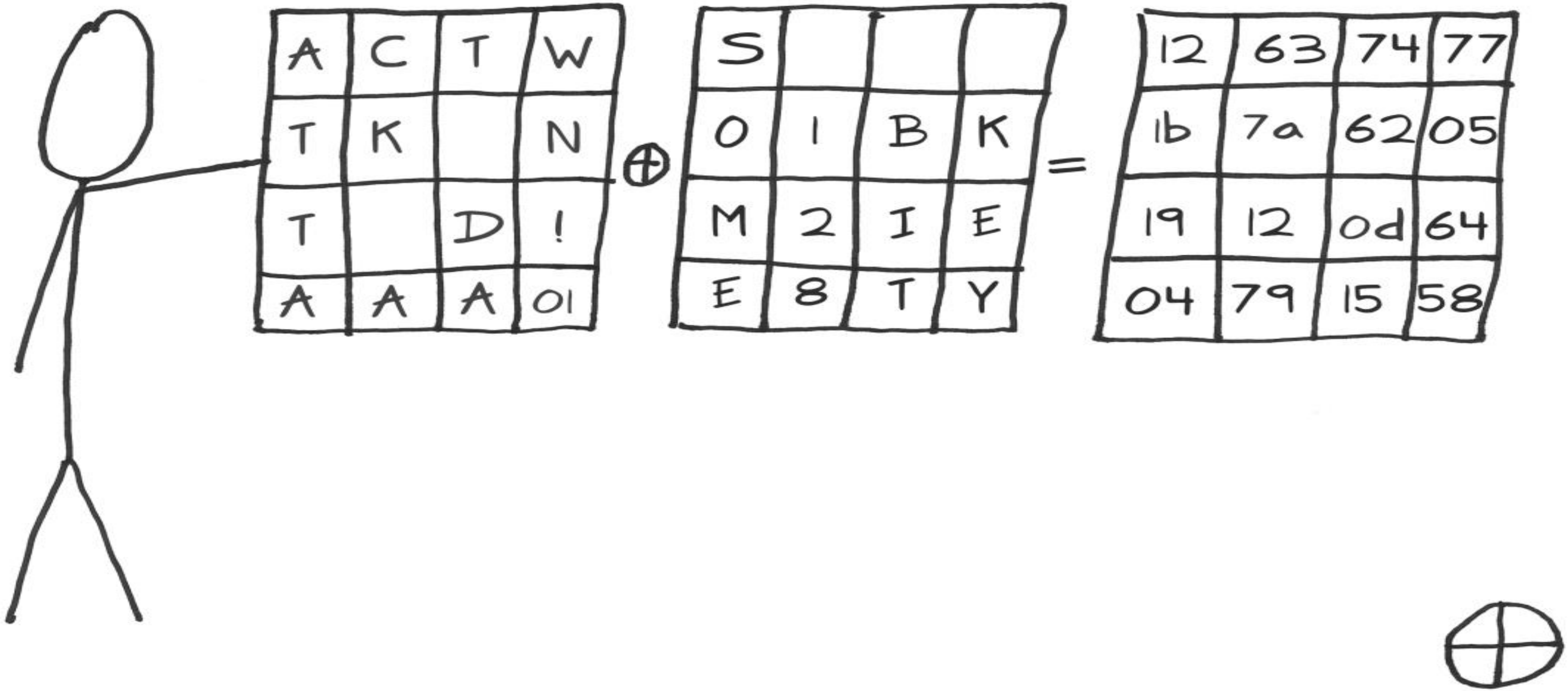
AES Round


State
SubBytes
S
State
ShiftRows
State
MixColumns
M
State
AddRoundKey
r_0 r_1 r_2 r_3 r_4 r_5 r_0 r_7 r_8 r_9 r_{10} r_{11} r_{12} r_{13} r_{14} r_{15}
State

●A Stick Figure Guide to AES

- http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html

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

⊕

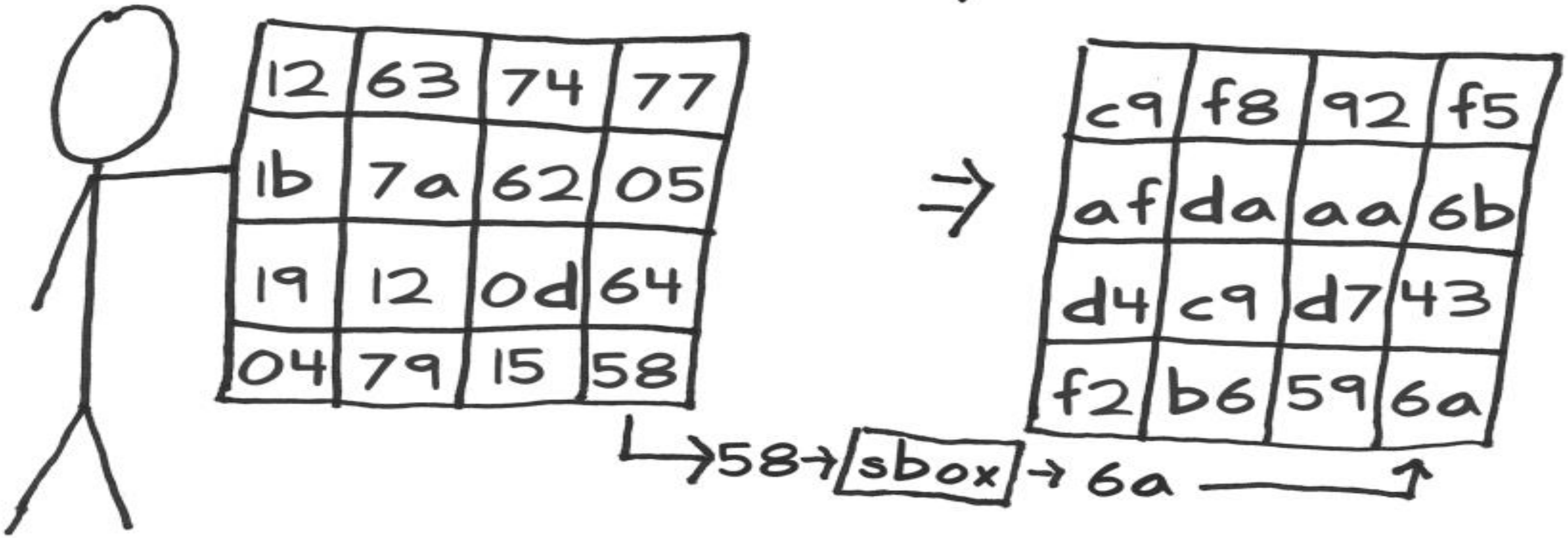
S			
0	1	B	K
M	2	I	E
E	8	T	Y

=

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

Applying Confusion: Substitute Bytes

I use confusion (Big Idea #1) to obscure the relationship of each byte. I put each byte into a substitution box (sbox), which will map it to a different byte:



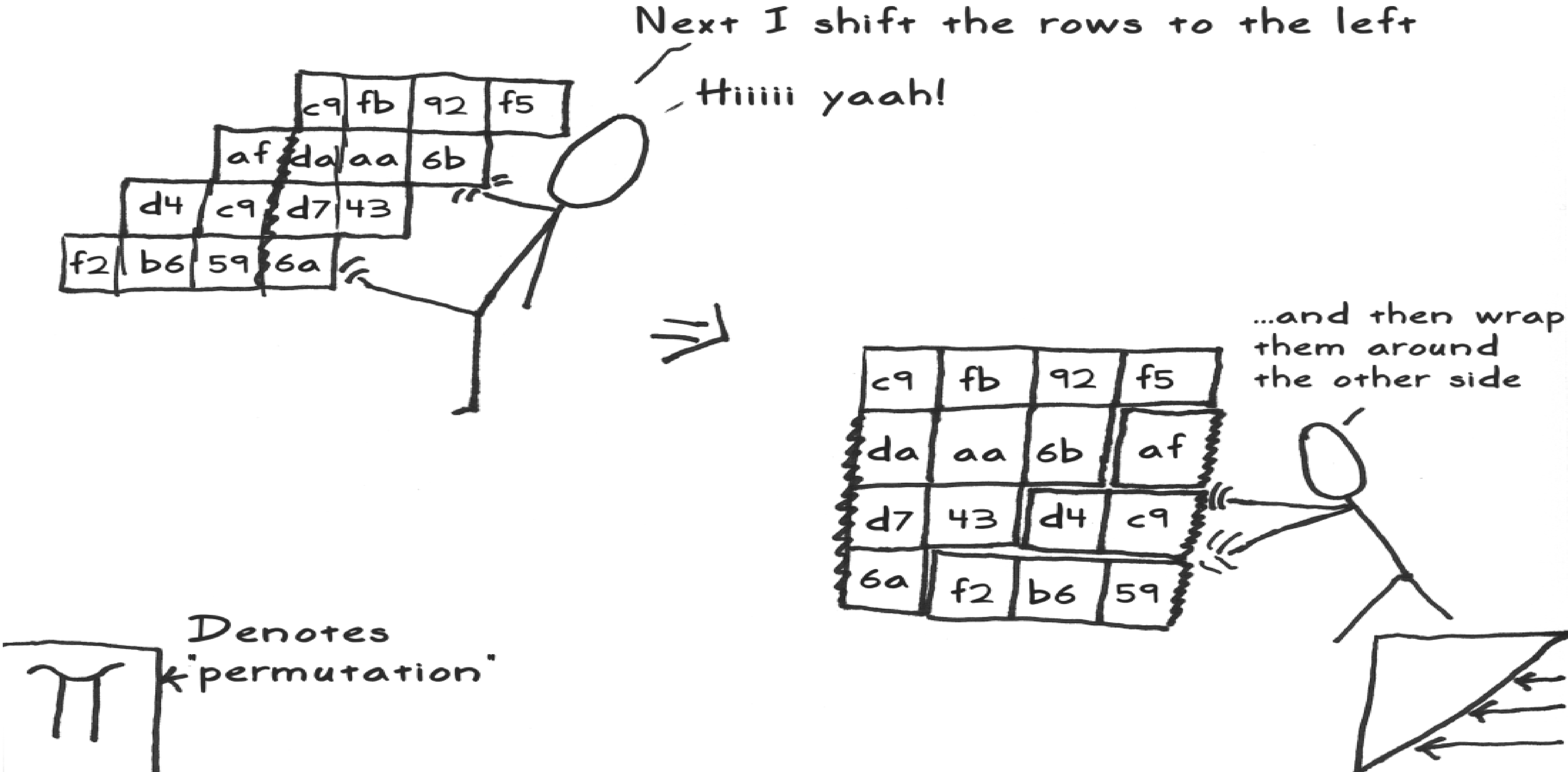
12	63	74	77
1b	7a	62	05
19	12	0d	64
04	79	15	58

⇒

c9	f8	92	f5
af	da	aa	6b
d4	c9	d7	43
f2	b6	59	6a

Denotes "confusion"

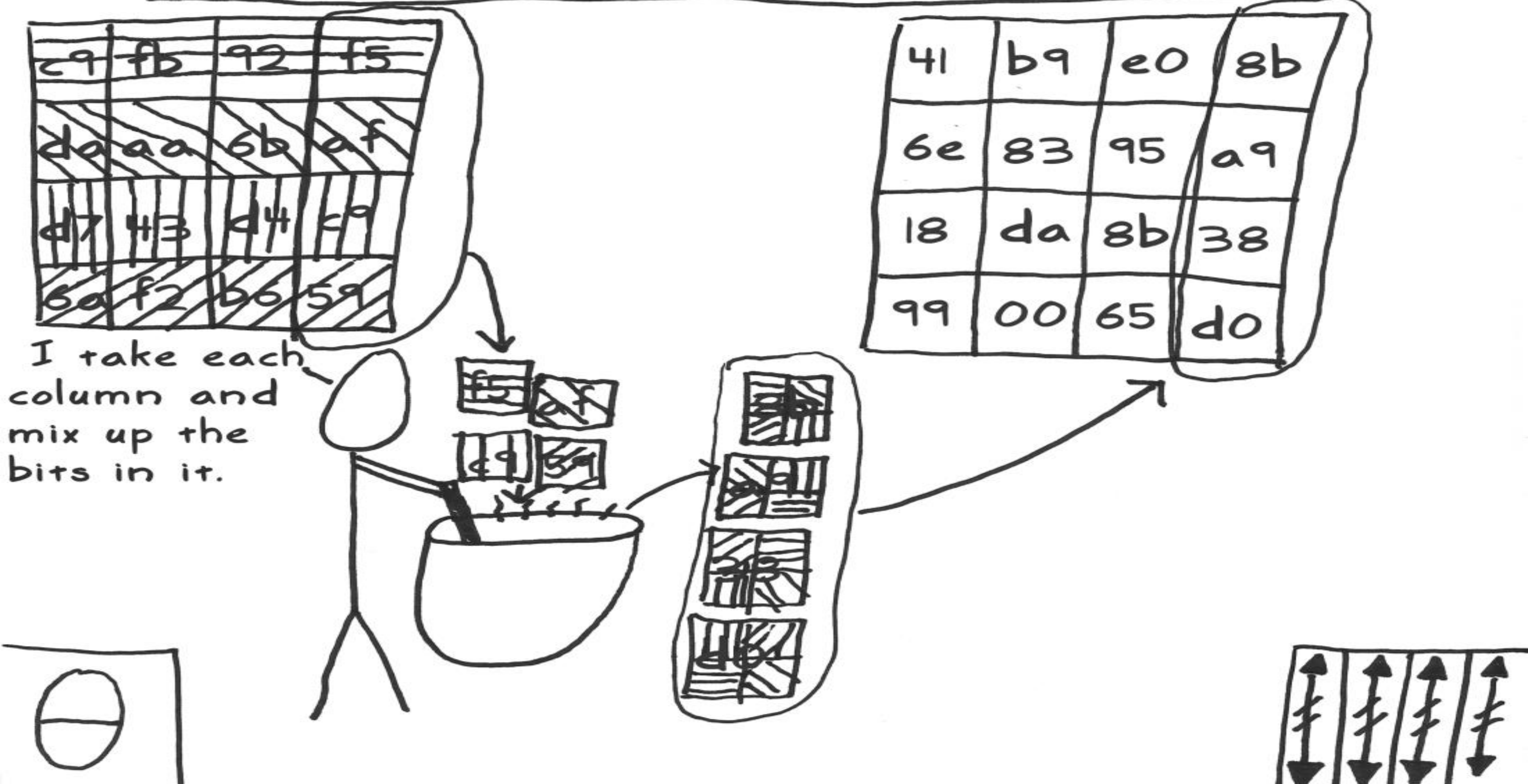
Applying Diffusion, Part 1: Shift Rows


Next I shift the rows to the left
Hiiiii yaah!
...and then wrap them around the other side
Π
Denotes "permutation"

			c9	fb	92	f5
		af	da	aa	6b	
	d4	c9	d7	43		
f2	b6	59	6a			

c9	fb	92	f5
da	aa	6b	af
d7	43	d4	c9
6a	f2	b6	59

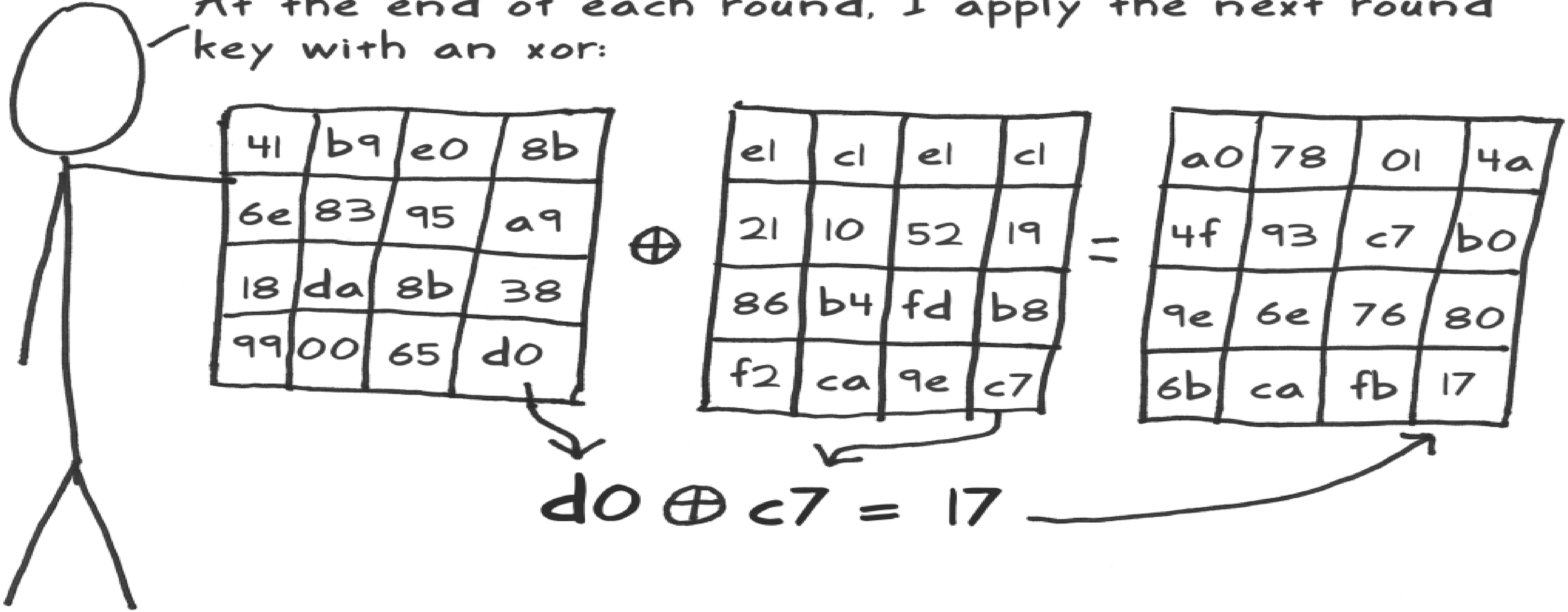
Applying Diffusion, Part 2: Mix Columns



c9	fb	92	f5
da	aa	6b	af
d7	f3	d4	c9
6a	f2	b6	59

41	b9	e0	8b
6e	83	95	a9
18	da	8b	38
99	00	65	d0

Applying Key Secrecy: Add Round Key

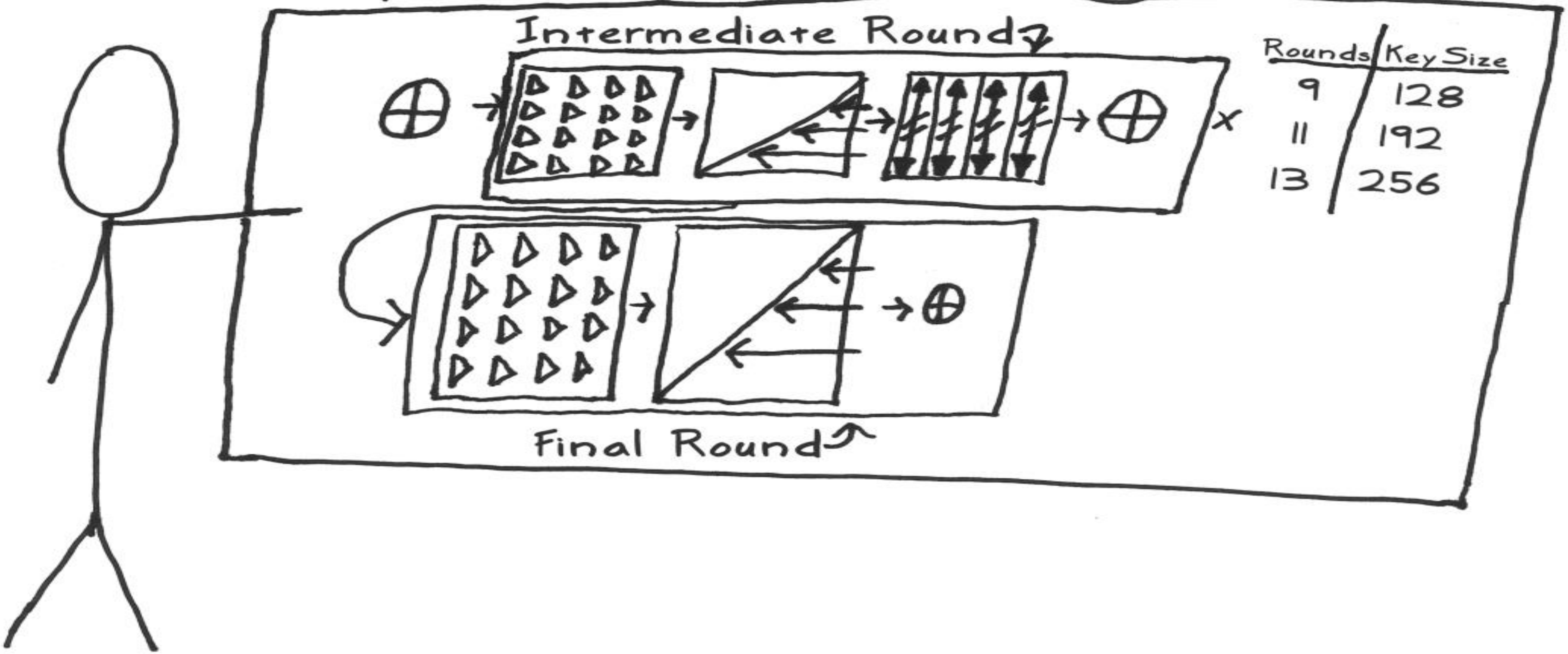


At the end of each round, I apply the next round key with an xor:

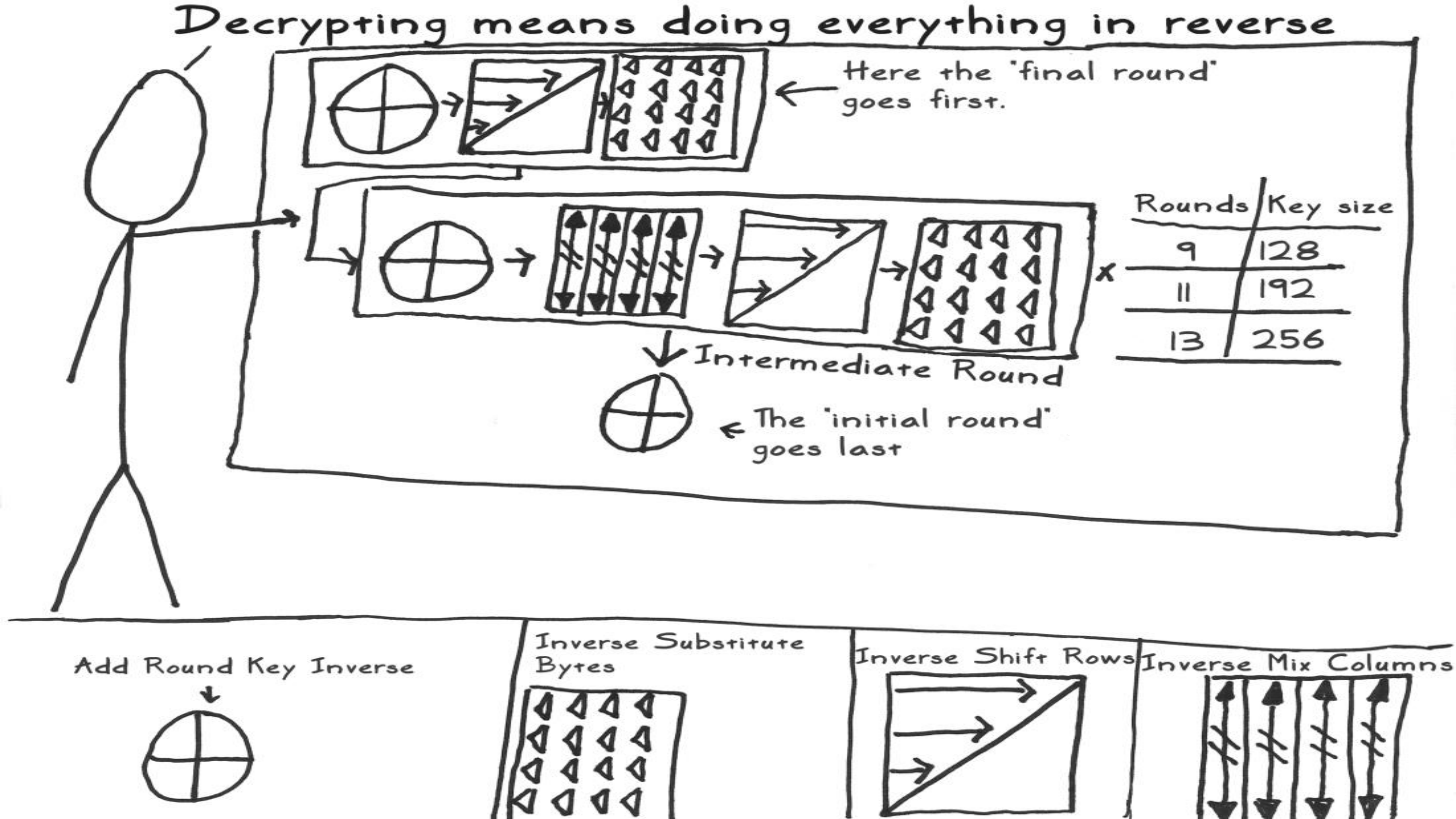
$$\begin{bmatrix} 41 & b9 & e0 & 8b \\ 6e & 83 & 95 & a9 \\ 18 & da & 8b & 38 \\ 99 & 00 & 65 & d0 \end{bmatrix} \oplus \begin{bmatrix} e1 & c1 & e1 & c1 \\ 21 & 10 & 52 & 19 \\ 86 & b4 & fd & b8 \\ f2 & ca & 9e & c7 \end{bmatrix} = \begin{bmatrix} a0 & 78 & 01 & 4a \\ 4f & 93 & c7 & b0 \\ 9e & 6e & 76 & 80 \\ 6b & ca & fb & 17 \end{bmatrix}$$

$$d0 \oplus c7 = 17$$

So in pictures, we have this:



Rounds	Key Size
9	128
11	192
13	256


Decrypting means doing everything in reverse
Here the "final round" goes first.
x
Intermediate Round
The "initial round" goes last
Add Round Key Inverse
Inverse Substitute Bytes
Inverse Shift Rows
Inverse Mix Columns

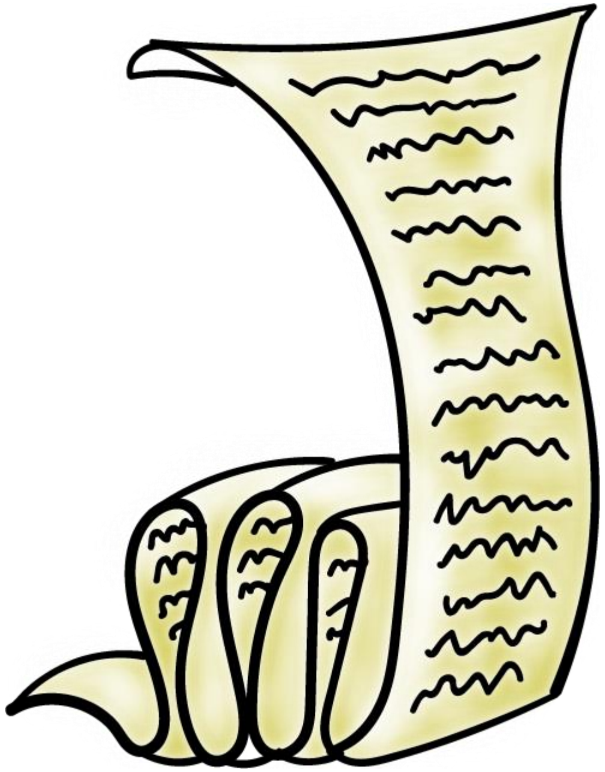
Rounds	Key size
9	128
11	192
13	256

AES Encryption Quiz

Check all the statements that are true:

- □ To decrypt using AES, just run the same algorithm in the same order of operations
- □ Each operation or stage in AES is reversible
- □ AES can support key length of 128, 192, 256
- □ AES is much more efficient than Triple DES

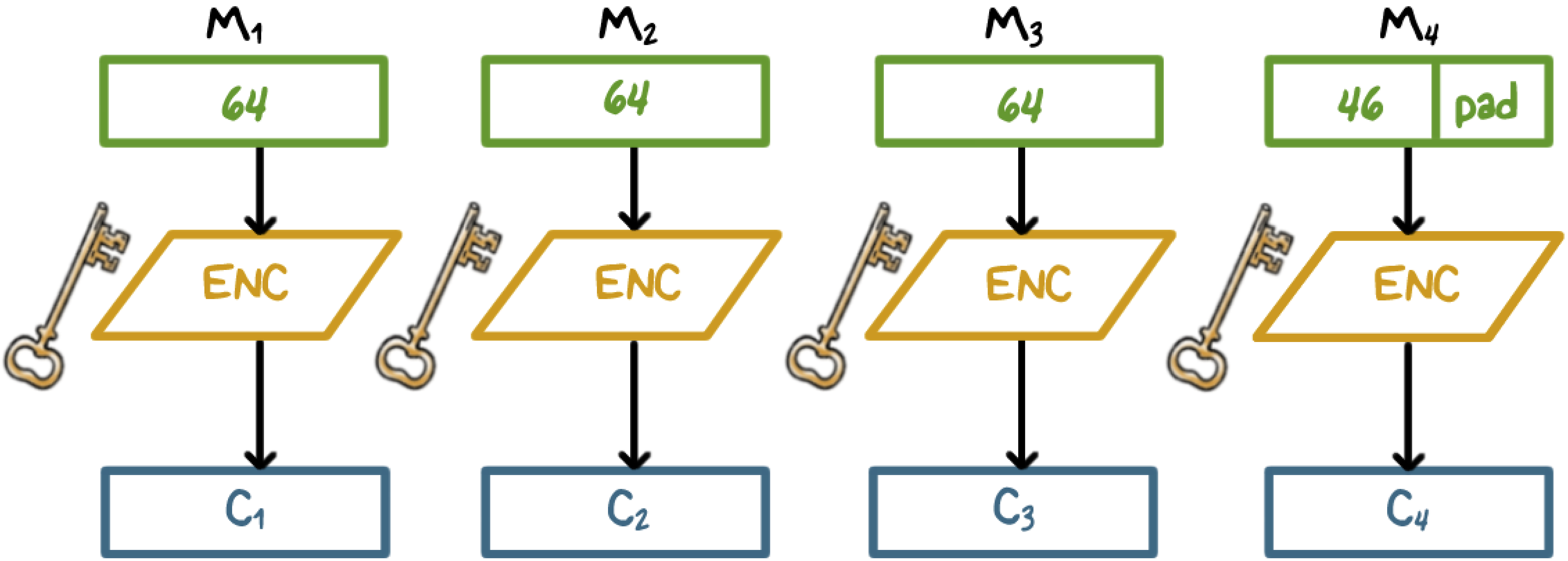
Encrypting a Large Message



- Break a message into blocks
- Apply block cipher on the blocks
- Is that it?

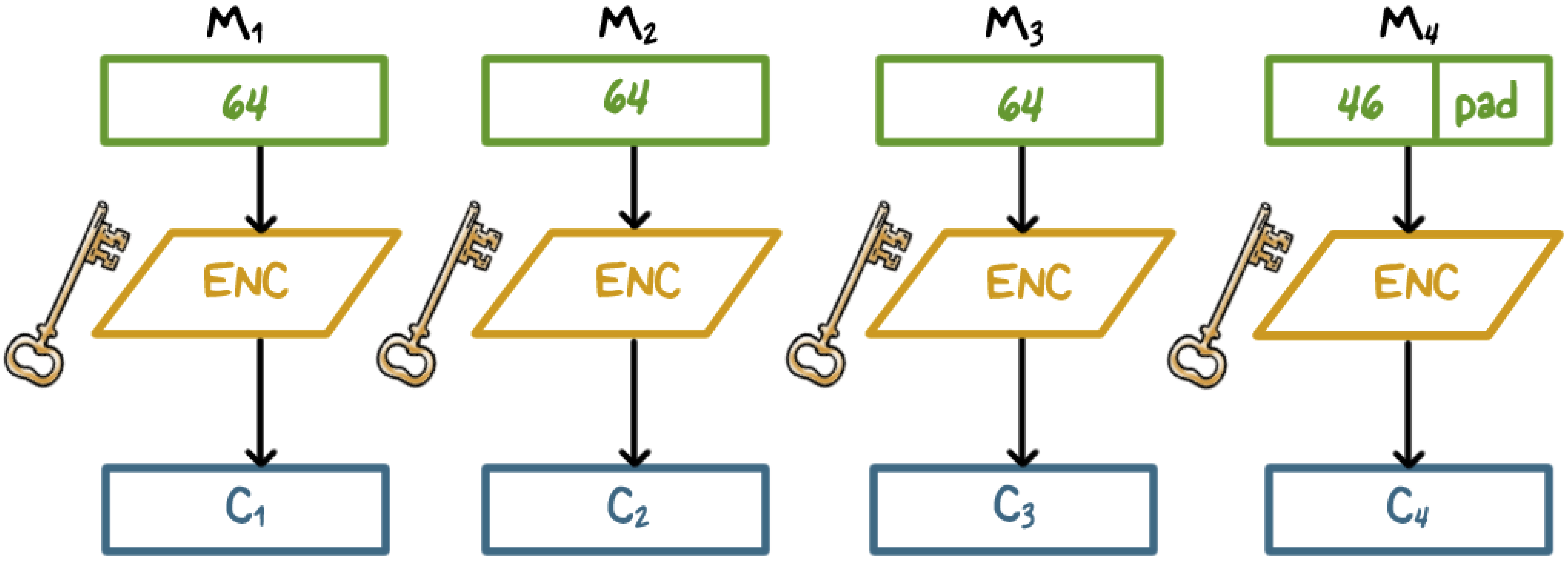
Encrypting a Large Message

Electronic Code Book (ECB)



Encrypting a Large Message

ECB Problem #1



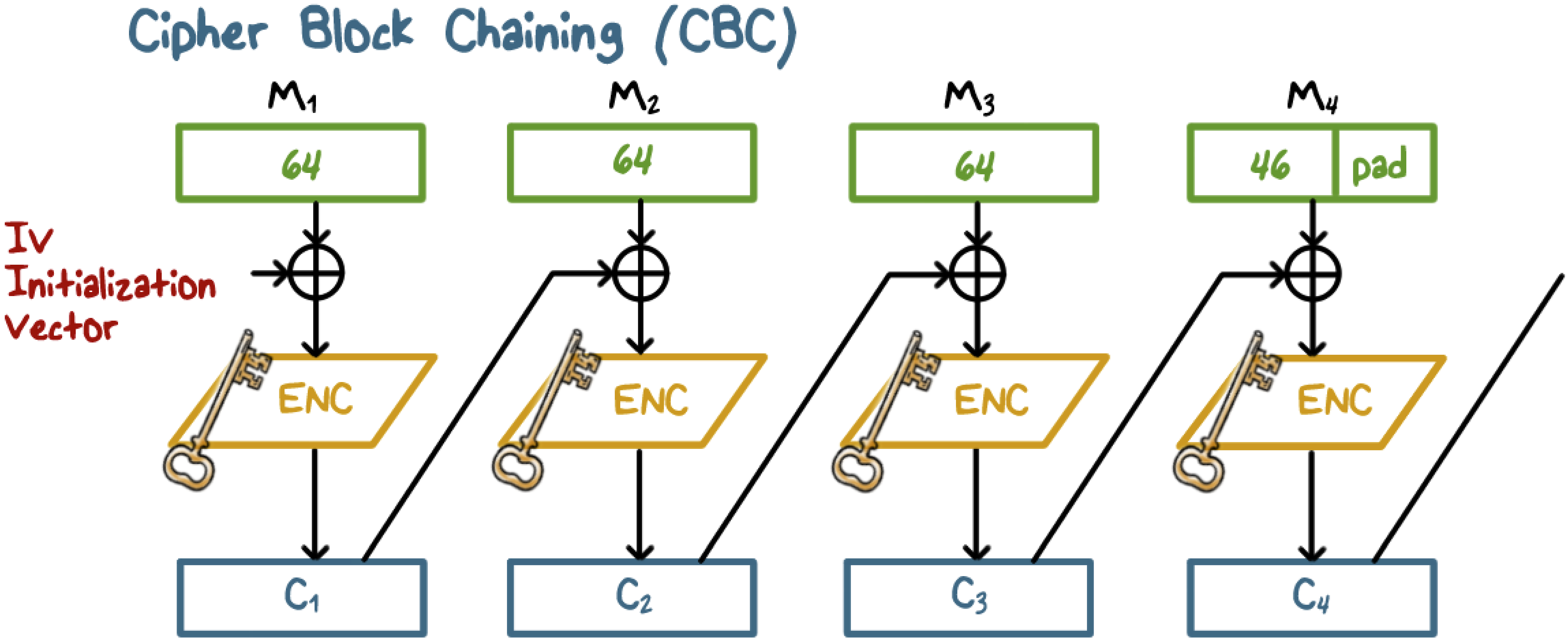
$(M_1 == M_3) \Rightarrow (C_1 == C_3)$

Encrypting a Large Message

ECB Problem #2

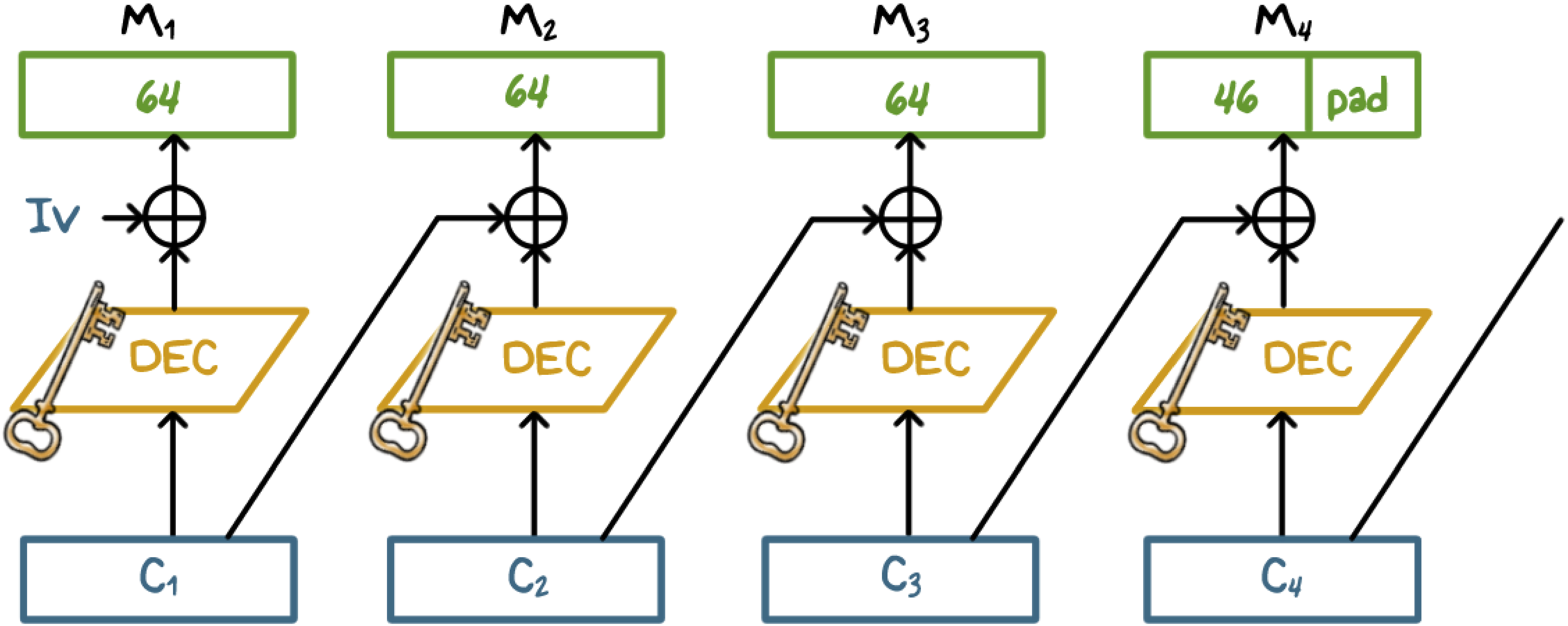
- **Lack the basic protection against integrity attacks** on the ciphertext at message level (i.e., multiple cipher blocks)
- Without additional integrity protection
  - **cipher block substitution** and rearrangement attacks
  - **fabrication** of specific information

Encrypting a Large Message

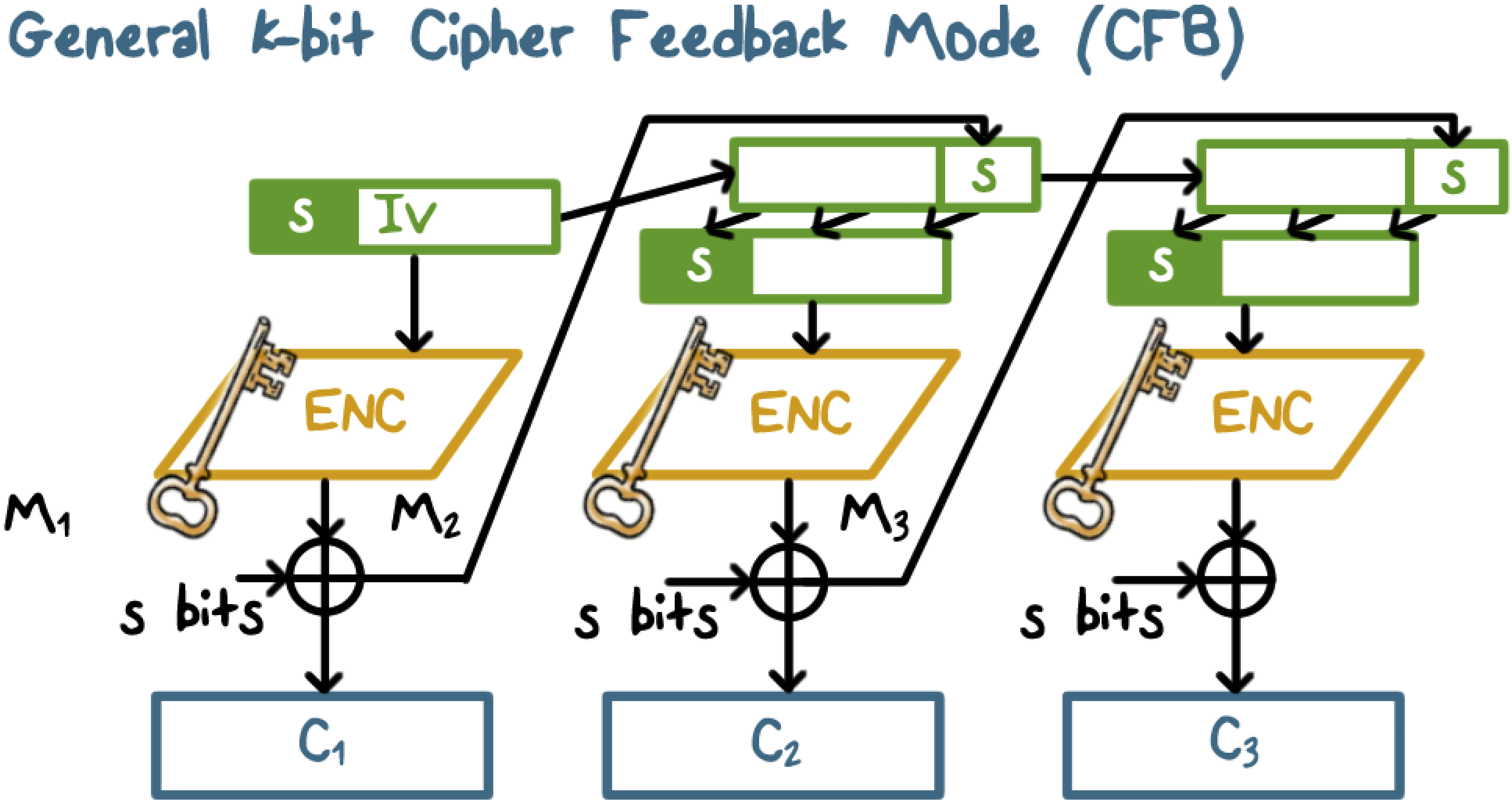


$(M_1 == M_3)$ very unlikely leads to $(C_1 == C_3)$

CBC Decryption



General K-Bit Cipher Feedback Mode (CFB)

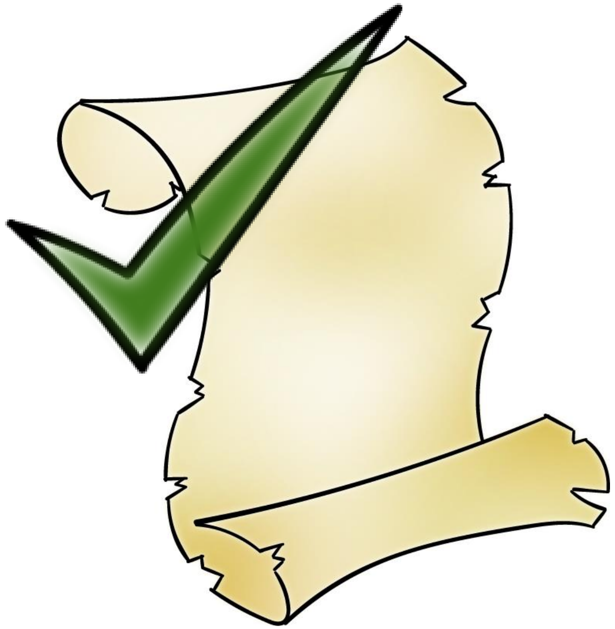


Protecting Message Integrity

- **Only send last block of CBC** (CBC residue) along with the plaintext

- Any modification in plaintext result in a CBC residue computed by the receiver to be different from the CBC residue from the sender
    - **Ensures integrity**

Protecting Message Integrity



- Simply sending all CBC blocks (for confidentiality) replicating last CBC block (for integrity) **does not work**
- **Should use two separate secret keys:** one for encryption and the other for generating residue (two encryption passes)
- Or, **CBC** (message|hash of message)

CBC Quiz

Put a check next to the statements that are true:

□ CBC is more secure than ECB

□ We can have both confidentiality and integrity protection with CBC by using just one key

Symmetric Encryption

Lesson Summary

- Need both confusion and diffusion
- DES: input 64-bit, key 56-bit; encryption and decryption same algorithms but reversed per-round key sequence
- AES: input 128-bit, key 128/192/256 bits; decryption the reverse/inverse of encryption
- Use cipher-block-chaining to encrypt a large message
- Last CBC block can be use as MIC; use different keys for integrity and confidentiality