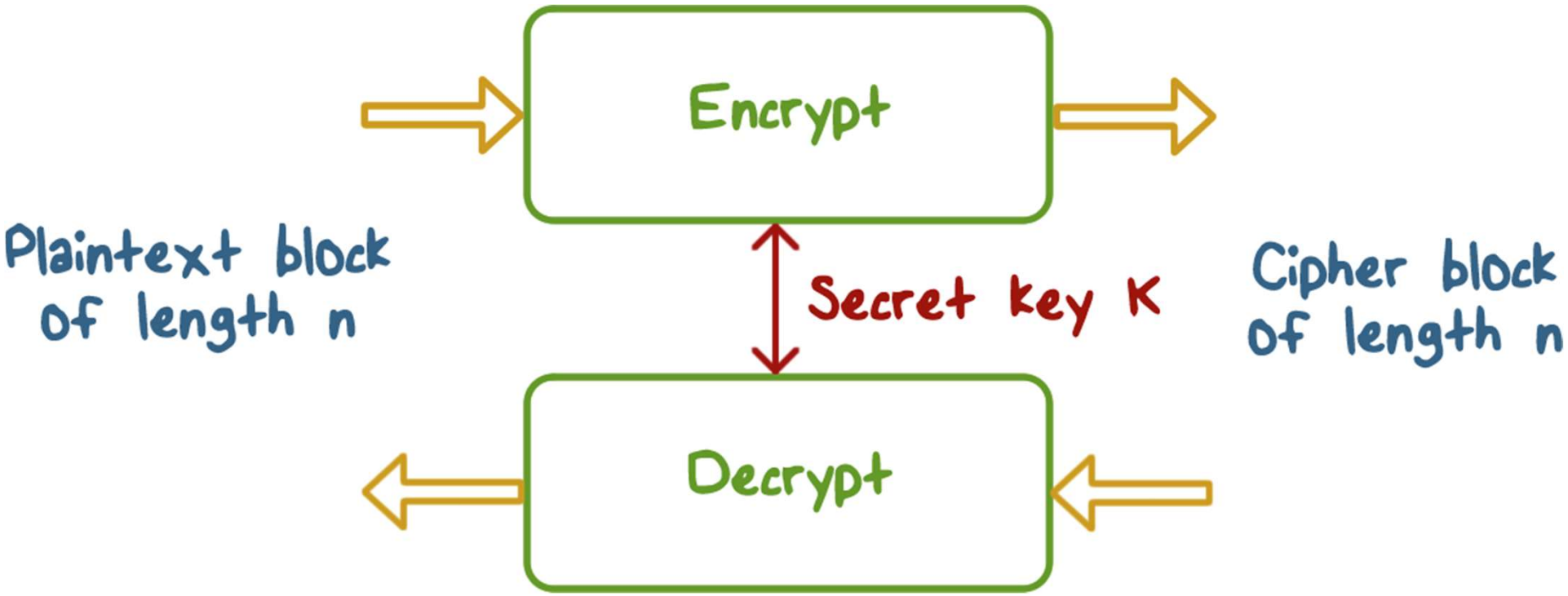
Symmetric Encryption

Lesson Introduction

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

Block Cipher Scheme



Block Cipher Primitives

Confusion:

Ensure that if one gets the cipher text, they cannot read the real meaning.

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

- Achieved with **substitution**

letters will not be encoded with itself

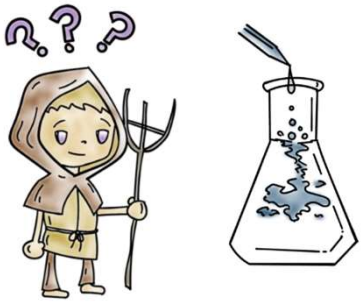
Block Cipher Primitives

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**

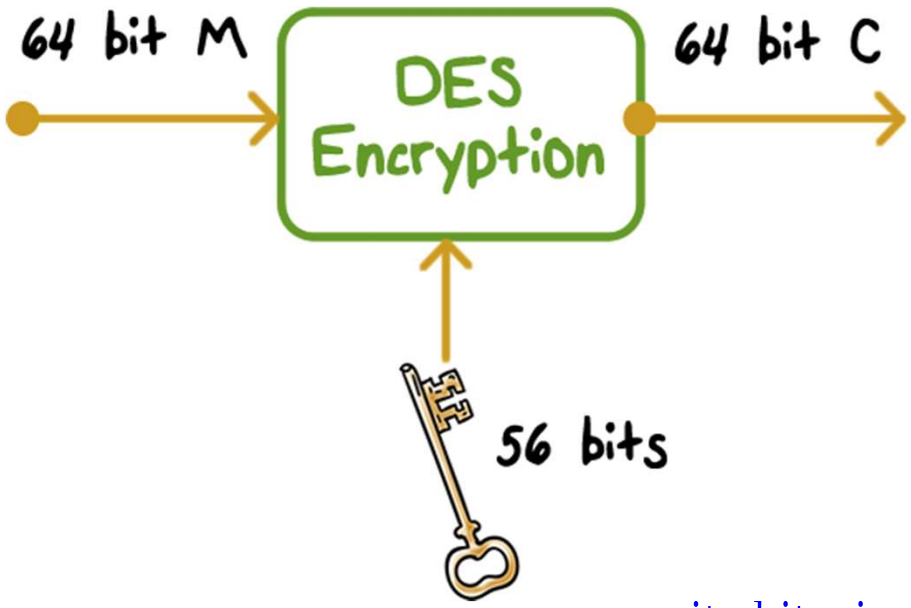
Block Cipher Primitives



- 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

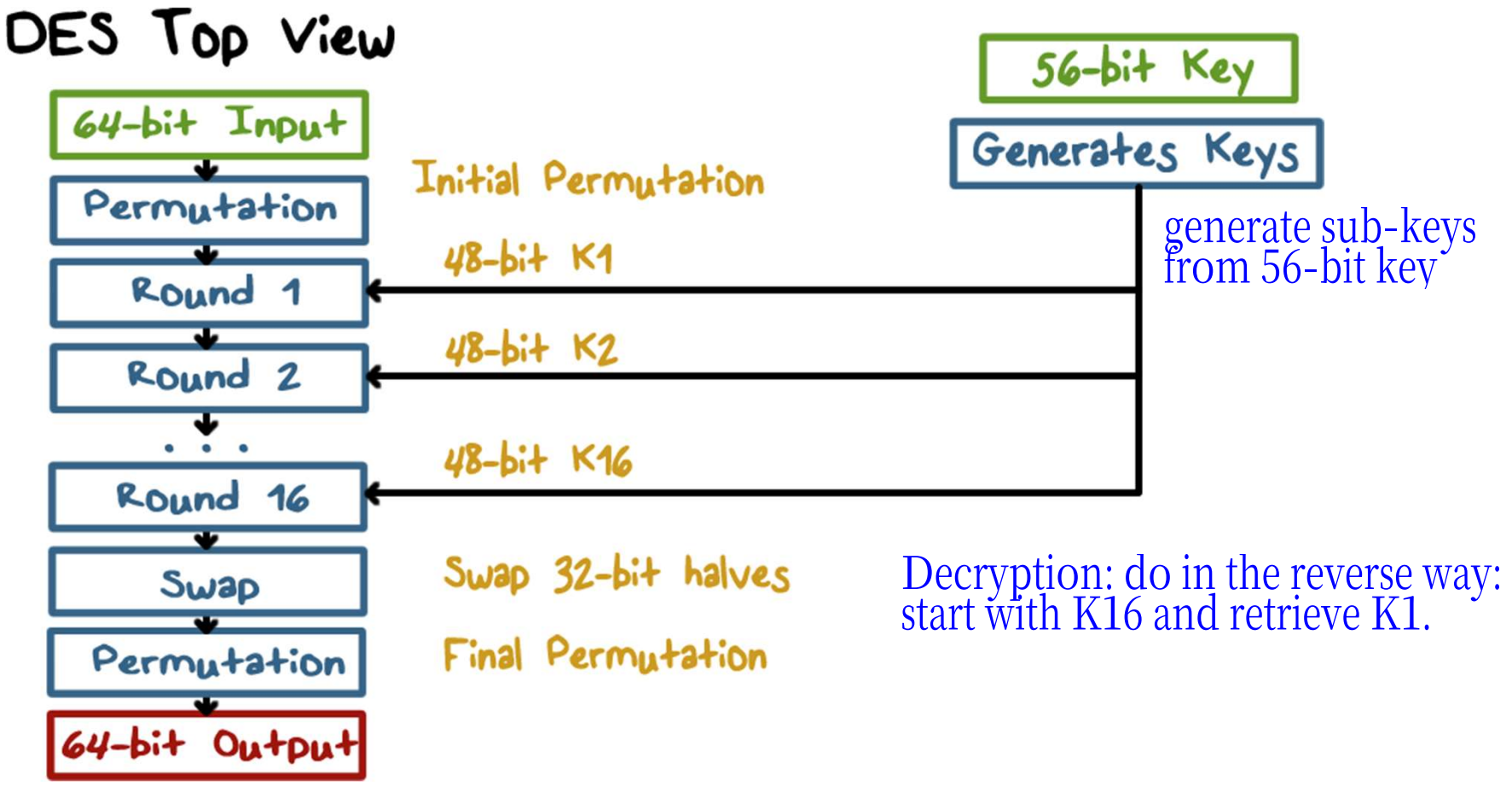
Data Encryption Standard



- 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

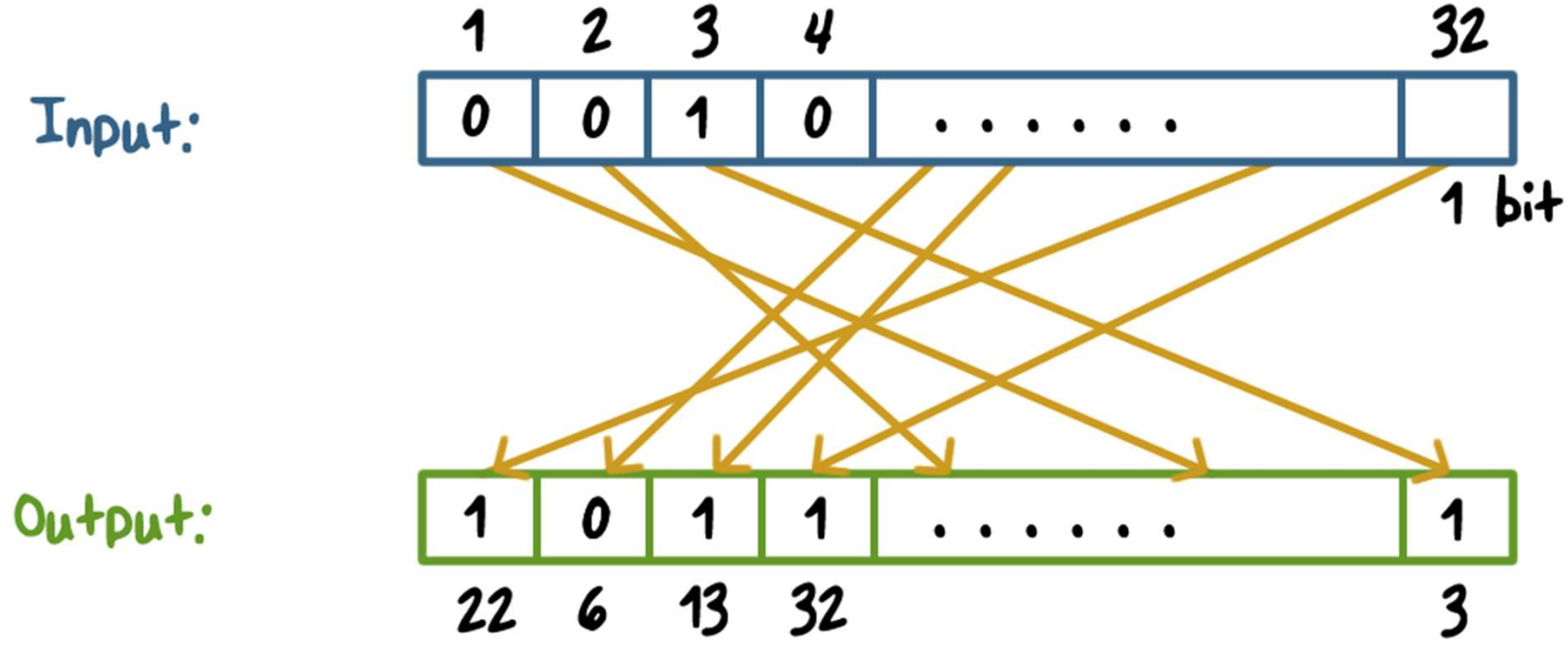
parity bit: gives you an easy way to do checking.

Data Encryption Standard

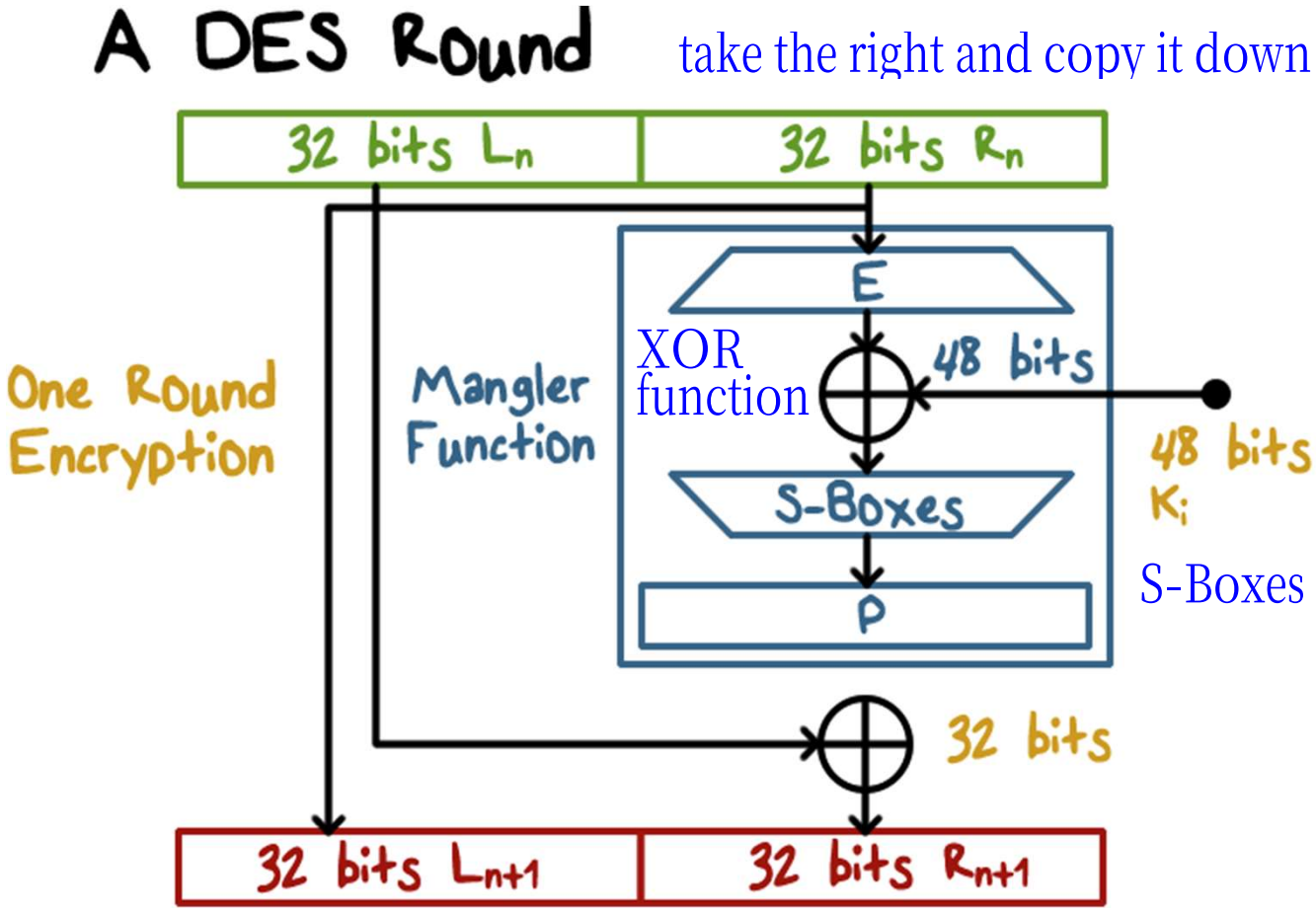


Data Encryption Standard

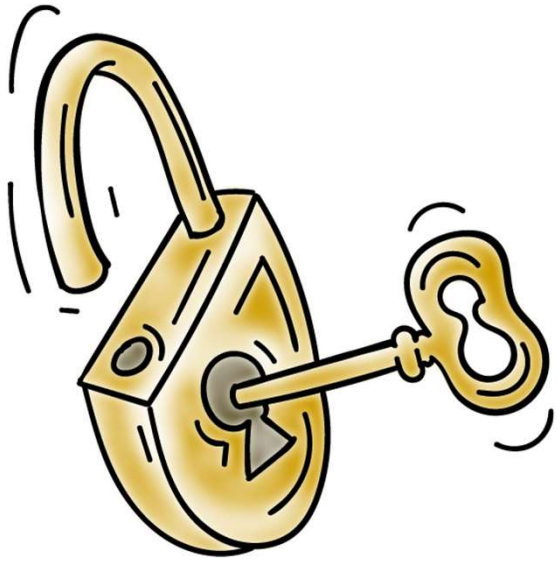
Bit Permutation (1-to-1)



Data Encryption Standard

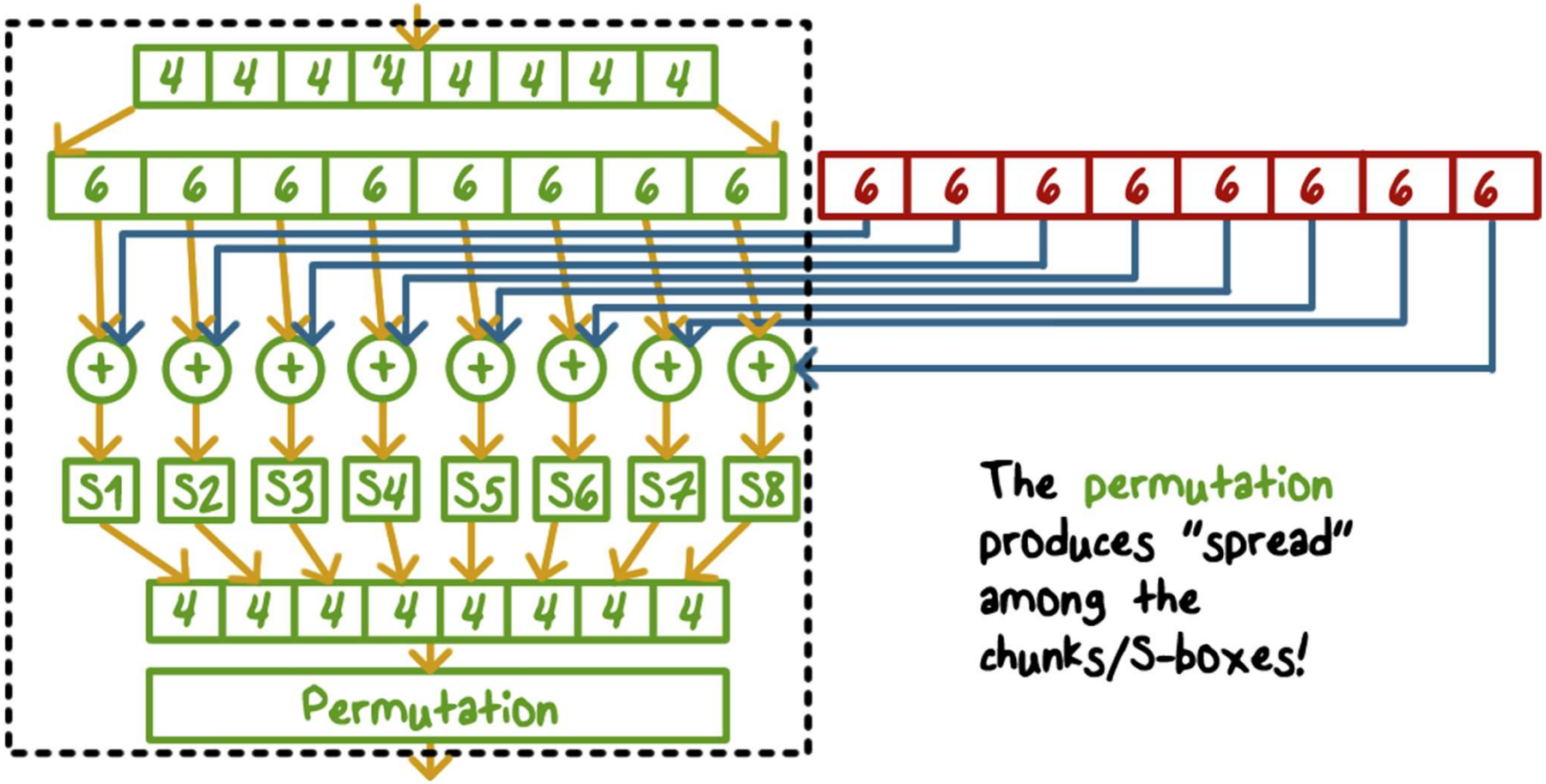


Decryption



- **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

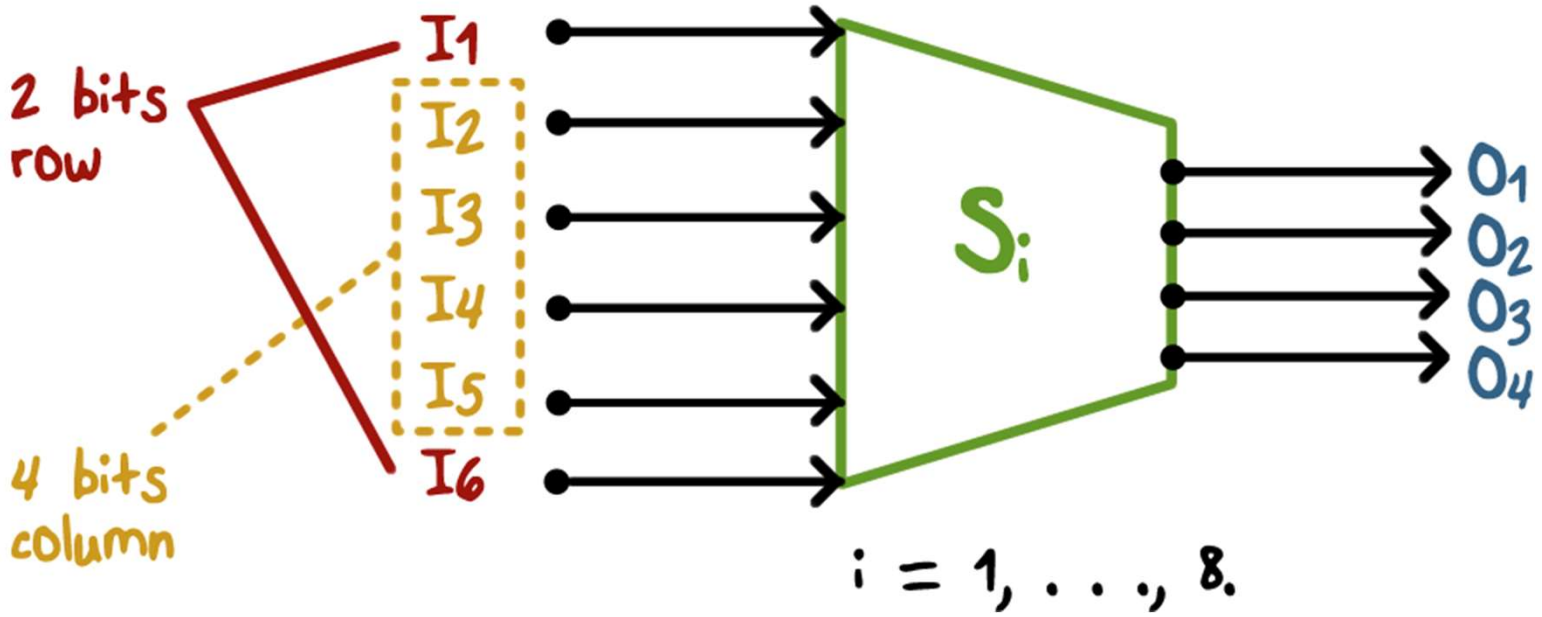
Mangler Function



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

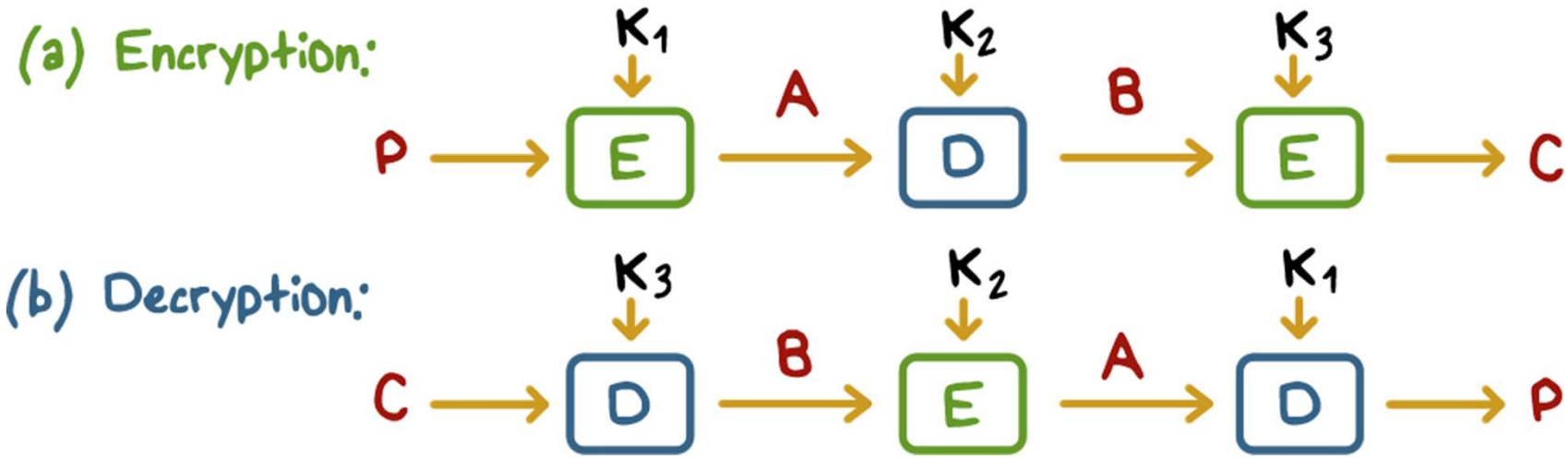


Security of DES

- **Key space is too small** (2^{56} keys)
  - Exhaustive key search relative easy with today's computers

    The overall security of this was called into question.
- **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

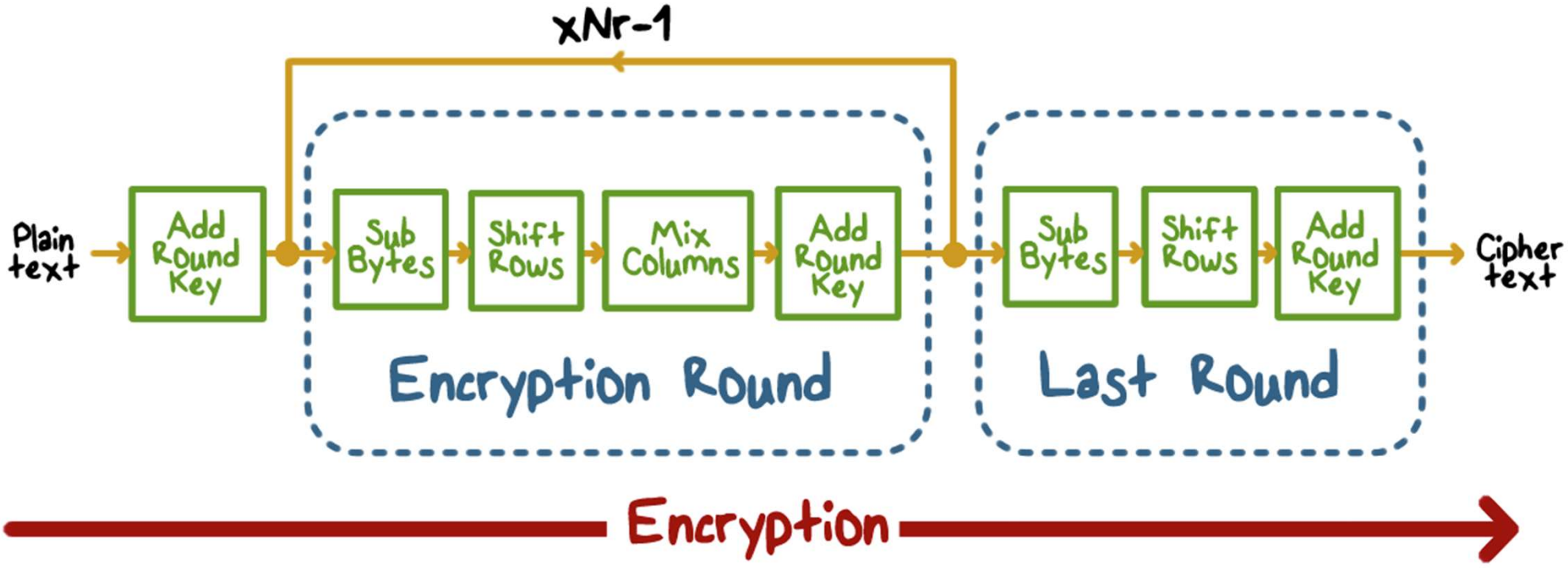
Usually, two keys are used.

Triple DES is still not secure. From the processing perspective, it needs more space and more processing time.

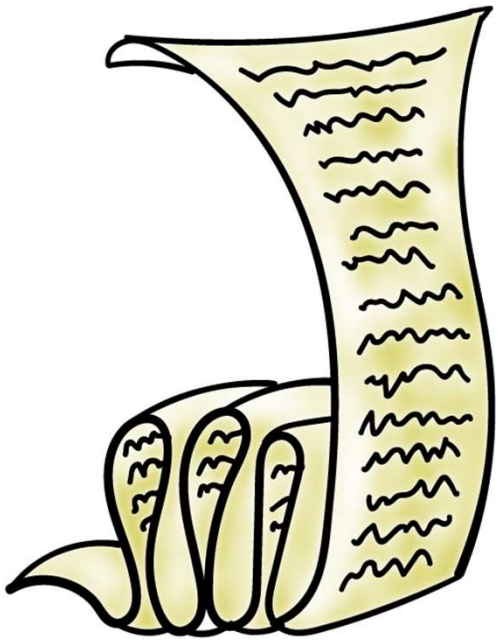
Advanced Encryption Standard

- 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

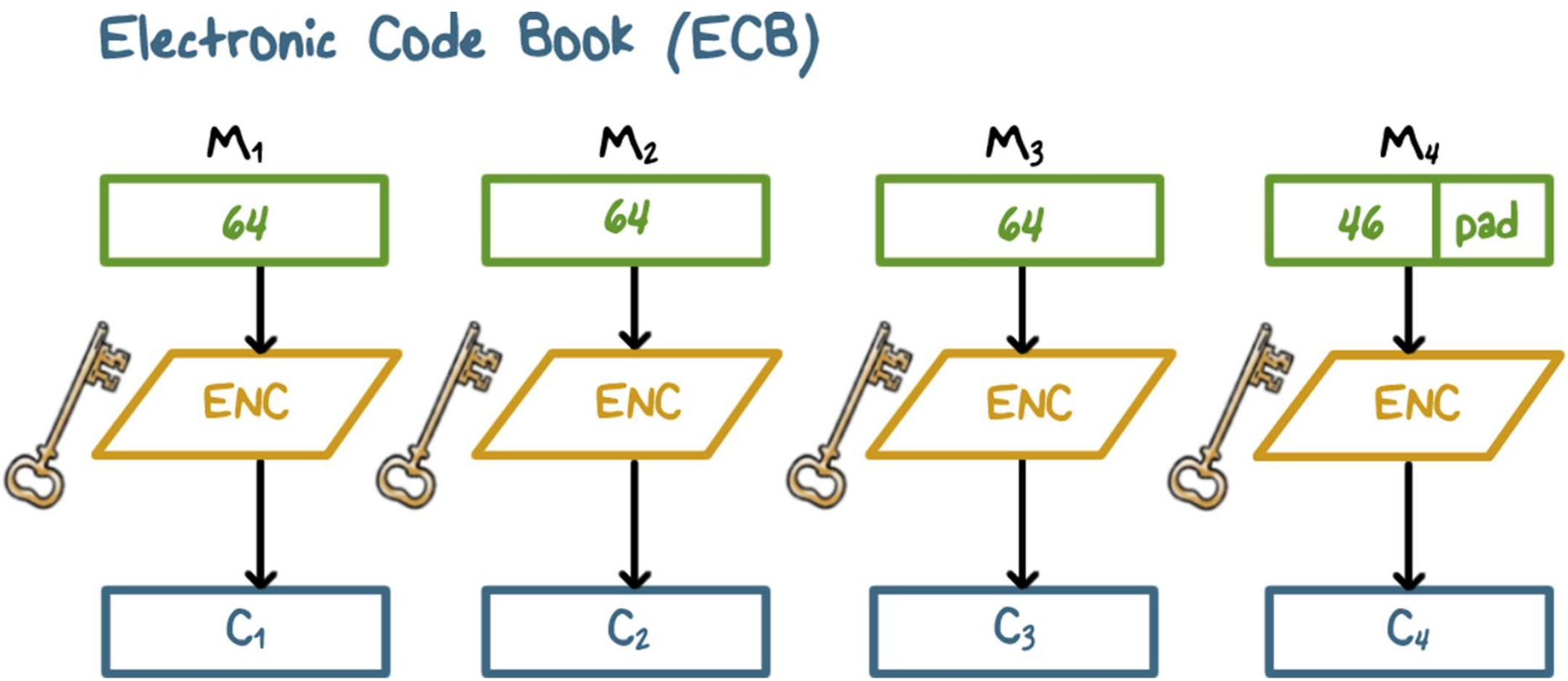


Encrypting a Large Message

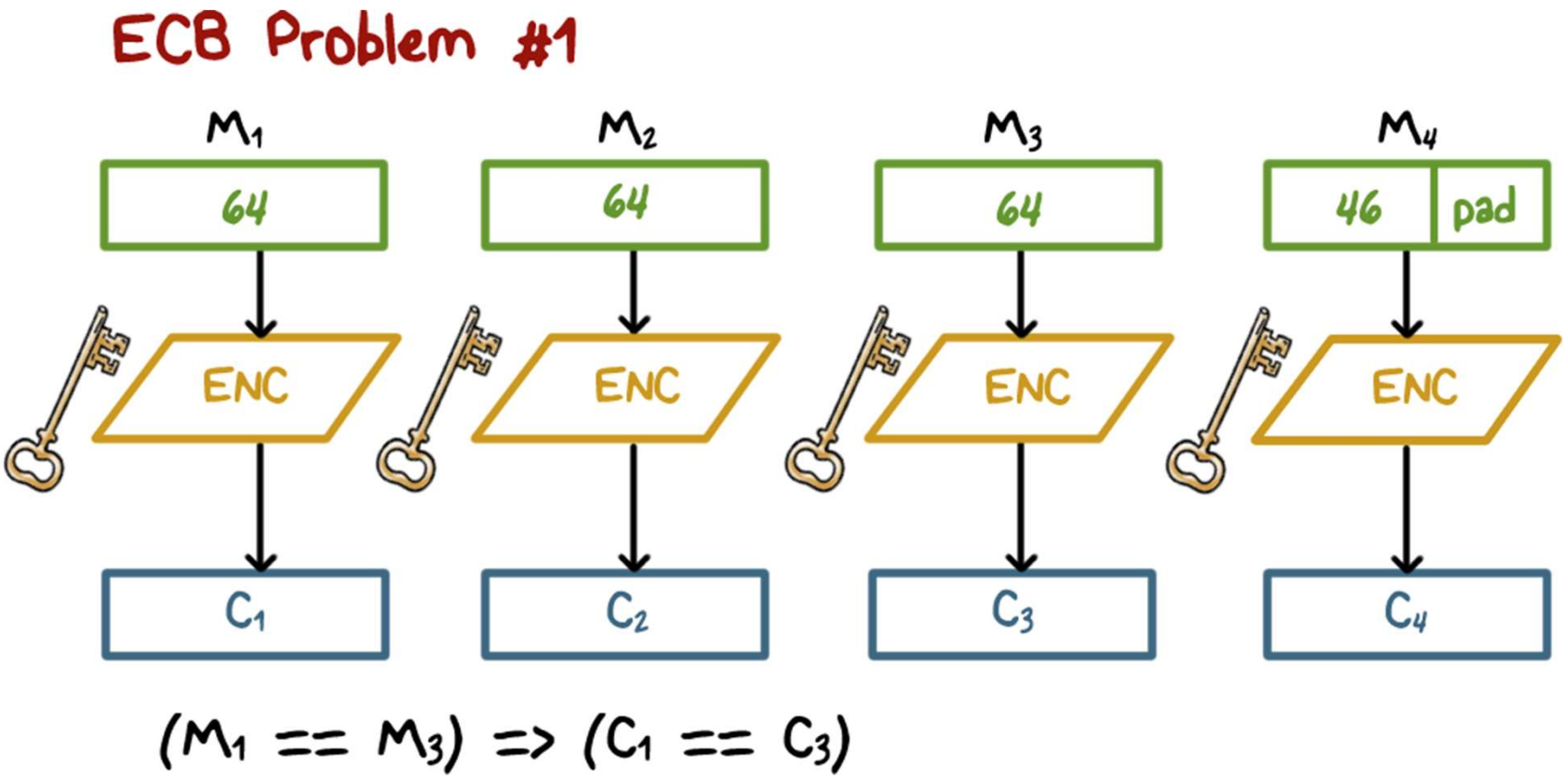


- Break a message into blocks
- Apply block cipher on the blocks
- Is that it?

Encrypting a Large Message



Encrypting a Large Message



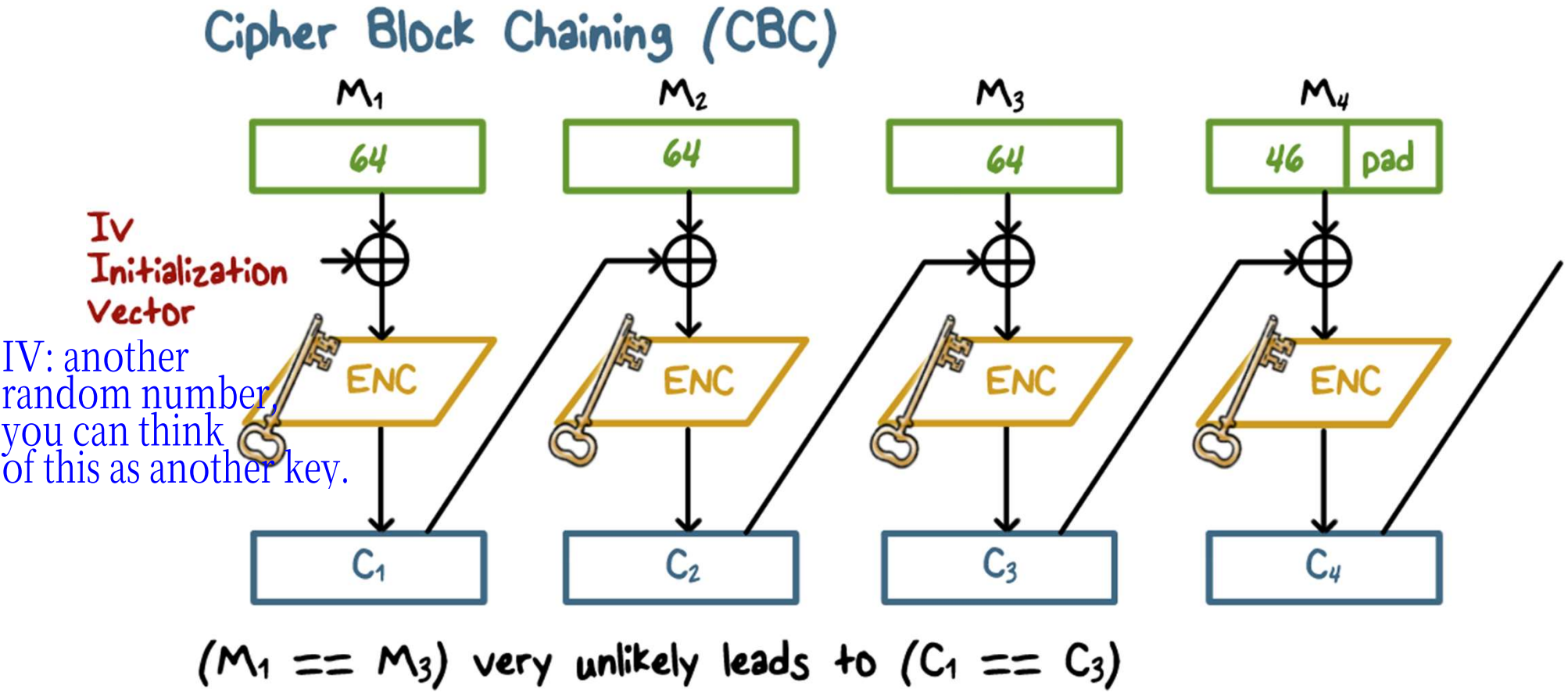
A potential vulnerability: providing the association between same characters in the plain text.

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



IV: another random number, you can think of this as another key.

All of these are associated, you can take a single piece out.

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**

run through it with the second key and checks if it gives me with the same CBC. If yes, the I can decrypt it with the first key.

So you can ensure that the plaintext is the same as the one from the original sender.

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)

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