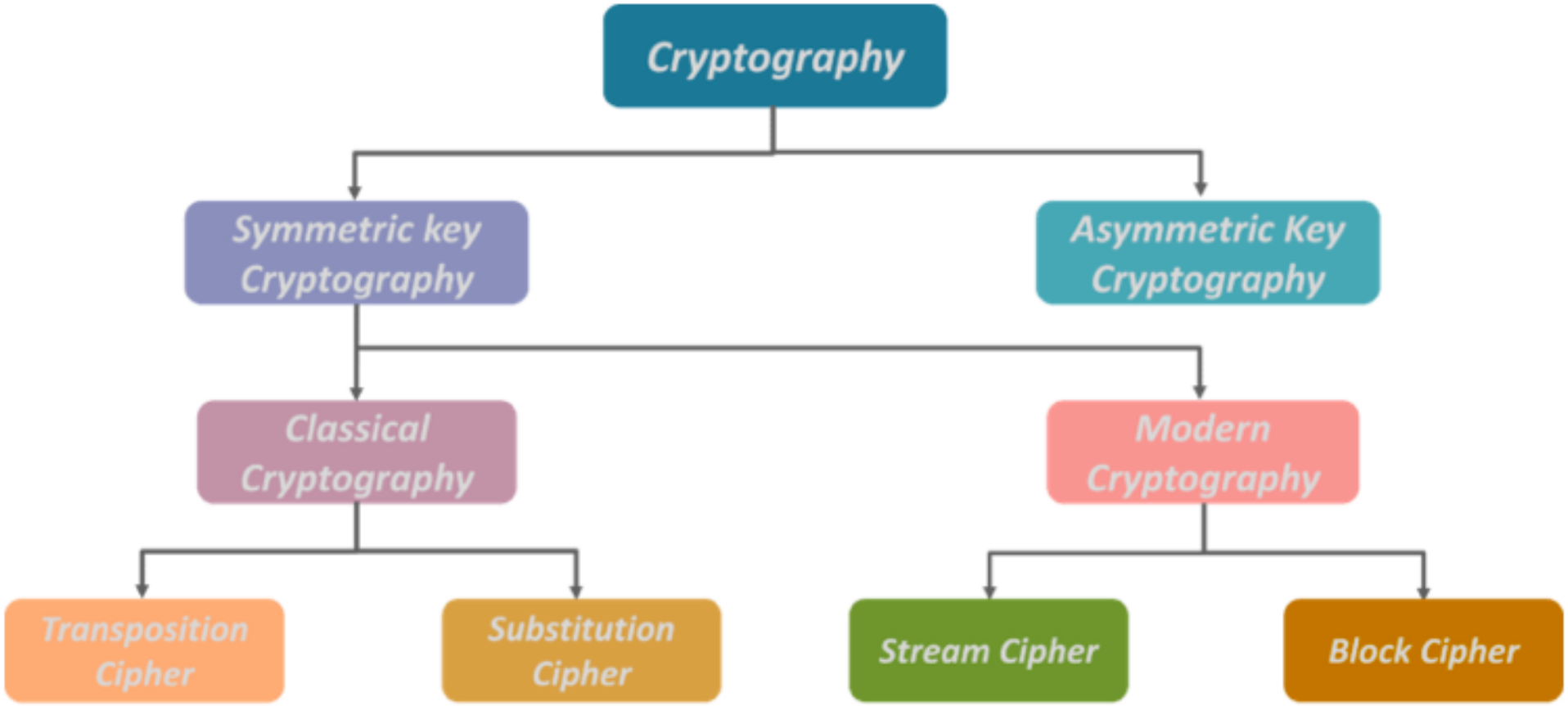
Practical Cryptography

Handout 3 - Symmetric Key Cryptography

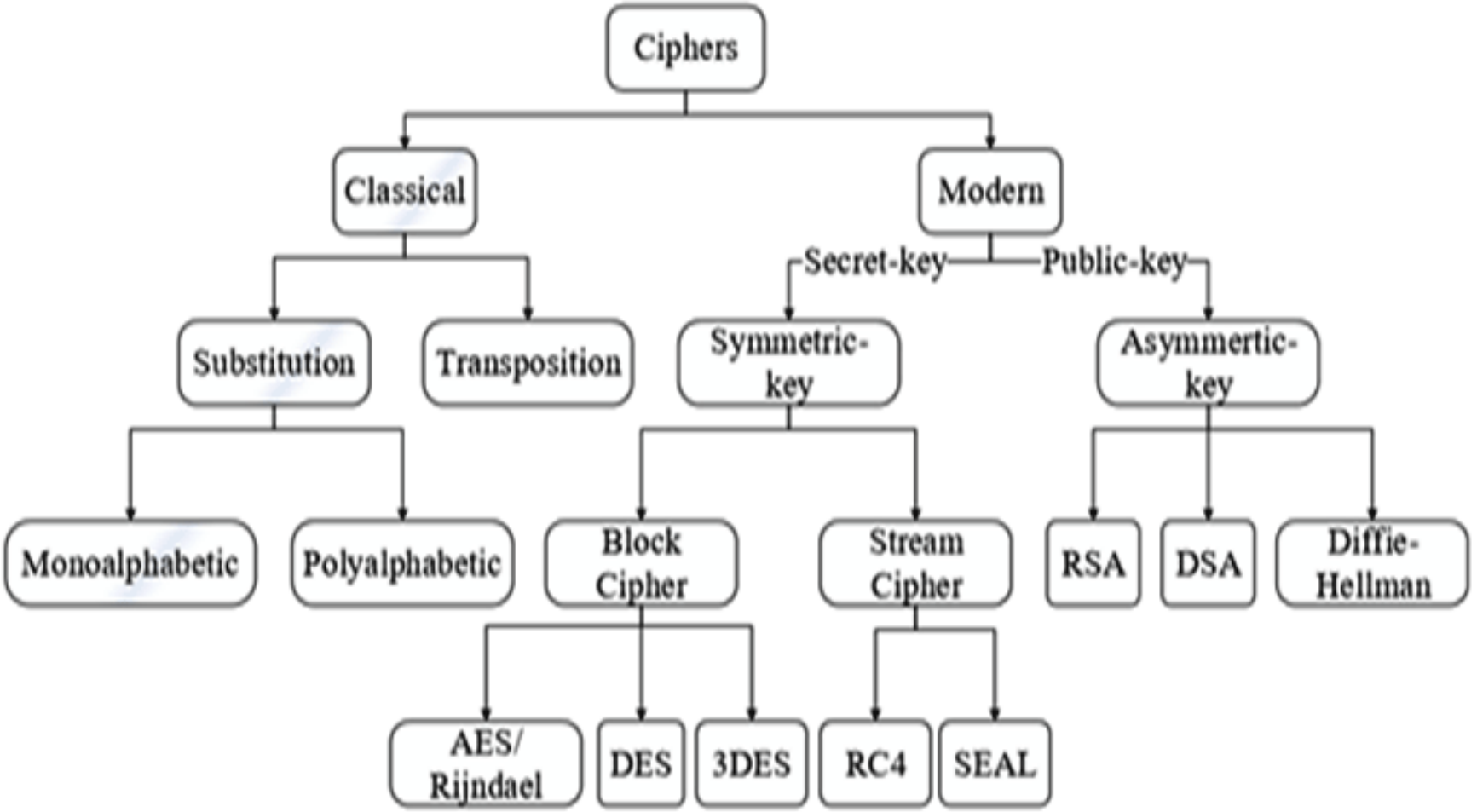
Kasun de Zoysa
kasun@ucsc.cmb.ac.lk

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

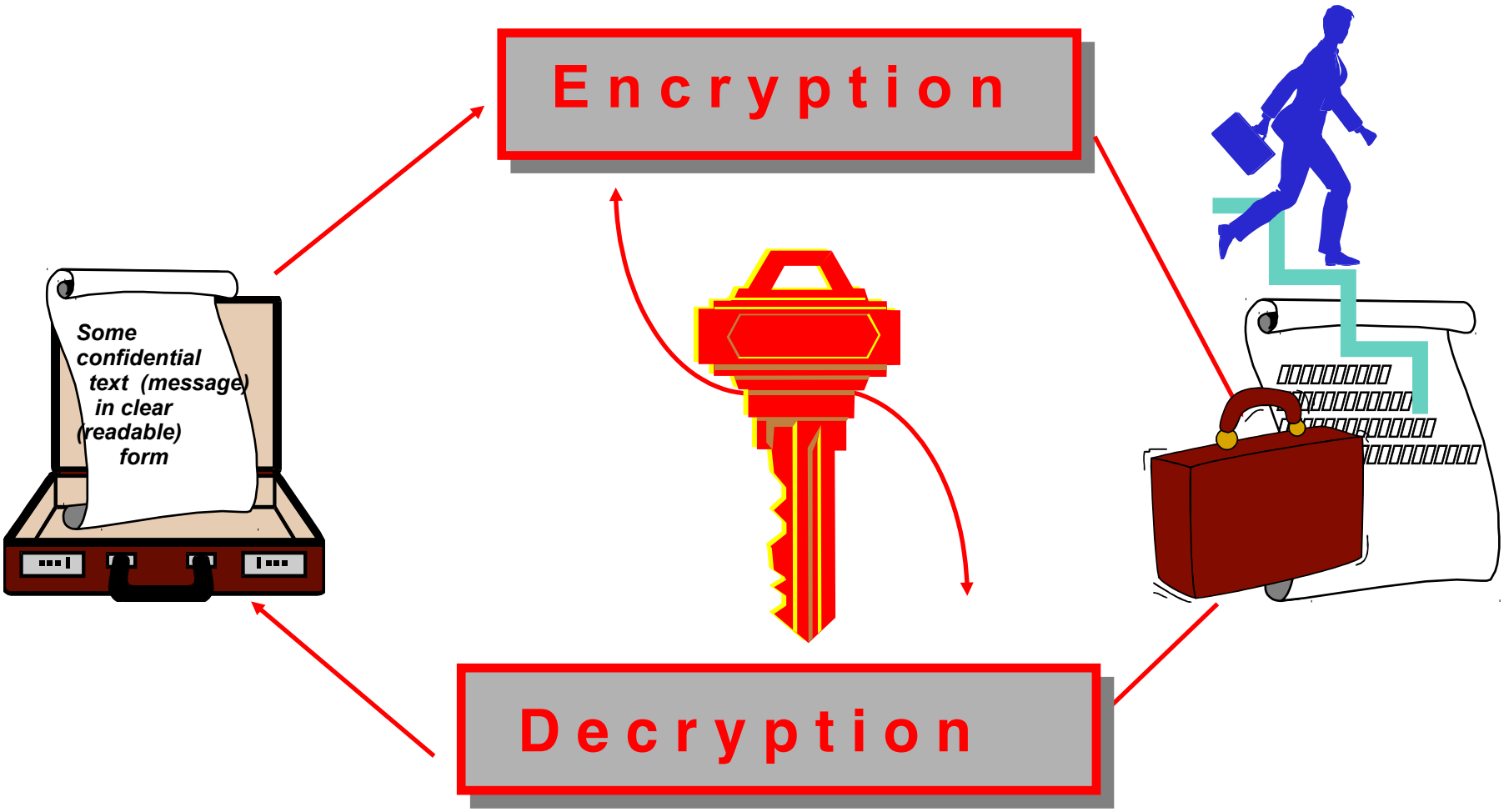
Cryptography



Ciphers


Ciphers
Classical
Modern
Secret-key
Public-key
Substitution
Transposition
Symmetric-key
Asymmetric-key
Monoalphabetic
Polyalphabetic
Block Cipher
Stream Cipher
RSA
DSA
Diffie-Hellman
AES/ Rijndael
DES
3DES
RC4
SEAL

Symmetric key Cryptograms



Requirements for Symmetric Key Cryptography

Two requirements for secure use of symmetric encryption:

- a strong encryption algorithm
- a secret key, *K*, known only to sender / receiver

$$Y = \mathrm{E}K(X)$$
$$X = \mathrm{D}K(Y)$$

  - Assume encryption algorithm is known
  - Implies a secure channel to distribute key

Symmetric Key / Private Key Cryptosystem

- Uses a single Private Key shared between users
- Strengths
  - Speed/ Efficient Algorithms – much quicker than Asymmetric
  - Hard to break when using a large Key Size
  - Ideal for bulk encryption / decryption
- Weaknesses
  - Poor Key Distribution (must be done out of band – ie phone, mail, etc)
  - Poor Key Management / Scalability (each user needs a unique key)
  - Cannot provide authenticity or non-repudiation – only confidentiality

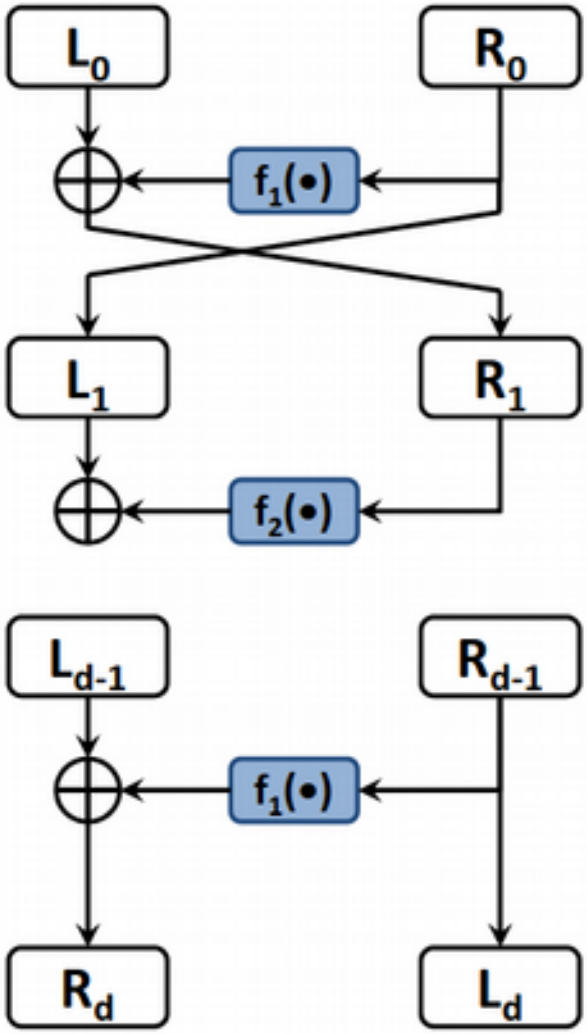
Block Ciphers

- **Block size:** in general larger block sizes mean greater security.
- **Key size:** larger key size means greater security (larger key space).
- **Number of rounds:** multiple rounds offer increasing security.
- **Encryption modes:** define how messages larger than the block size are encrypted, very important for the security of the encrypted message.

Feistel Network

- Several block ciphers are based on the structure proposed by Feistel in 1973
- A Feistel Network is fully specified given
  - the block size: n = 2w
  - number of rounds: d
  - d round functions f1, …, fd: {0,1}w {0,1}w
- Used in DES, IDEA, RC5, and many other block ciphers.
- Not used in AES

Feistel Network

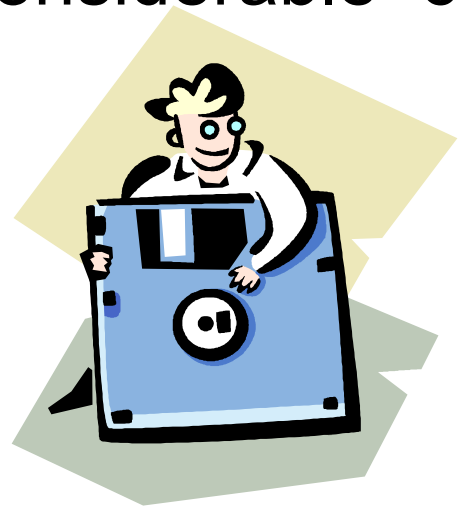


- **Encryption:**
  - $L_1 = R_0 \quad R_1 = L_0 \oplus f_1(R_0)$
  - $L_2 = R_1 \quad R_2 = L_1 \oplus f_2(R_1)$
  - ...
  - $L_d = R_{d-1} \quad R_d = L_{d-1} \oplus f_d(R_{d-1})$
- **Decryption:**
  - $R_{d-1} = L_d \quad L_{d-1} = R_d \oplus f_d(L_d)$
  - ...
  - $R_0 = L_1; \quad L_0 = R_1 \oplus f_1(L_1)$

5

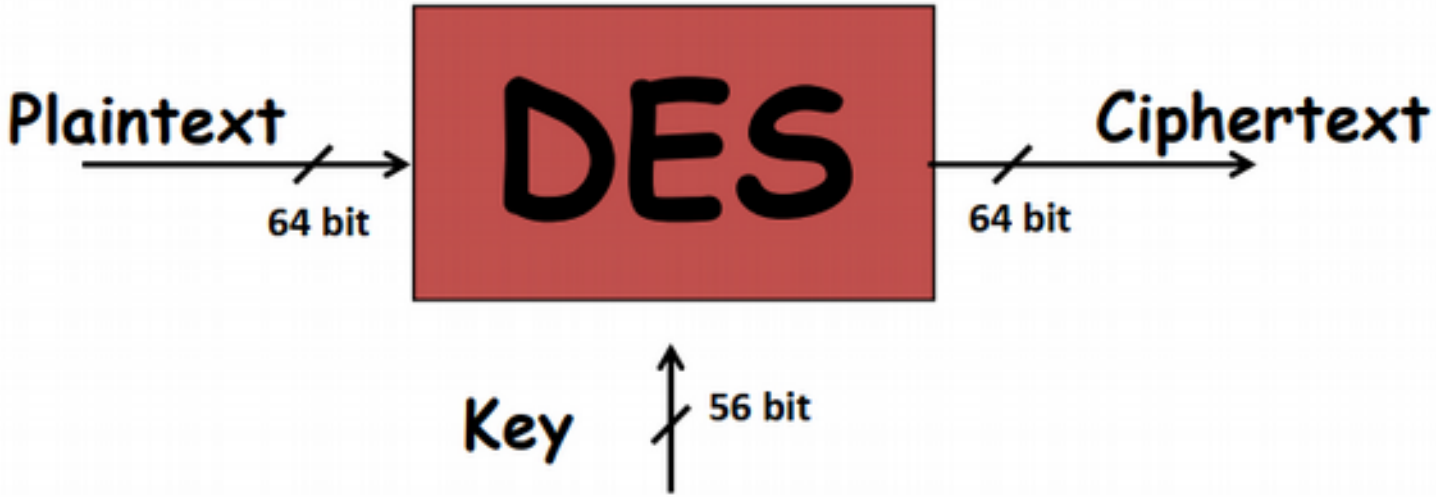
Data Encryption Standard (DES)

- Most widely used block cipher in world
- Adopted in 1977 by NBS (now NIST) as FIPS PUB 46
- Encrypts 64-bit data using 56-bit key
- Has widespread use
- Has been the subject of considerable controversy over its security



DES Features

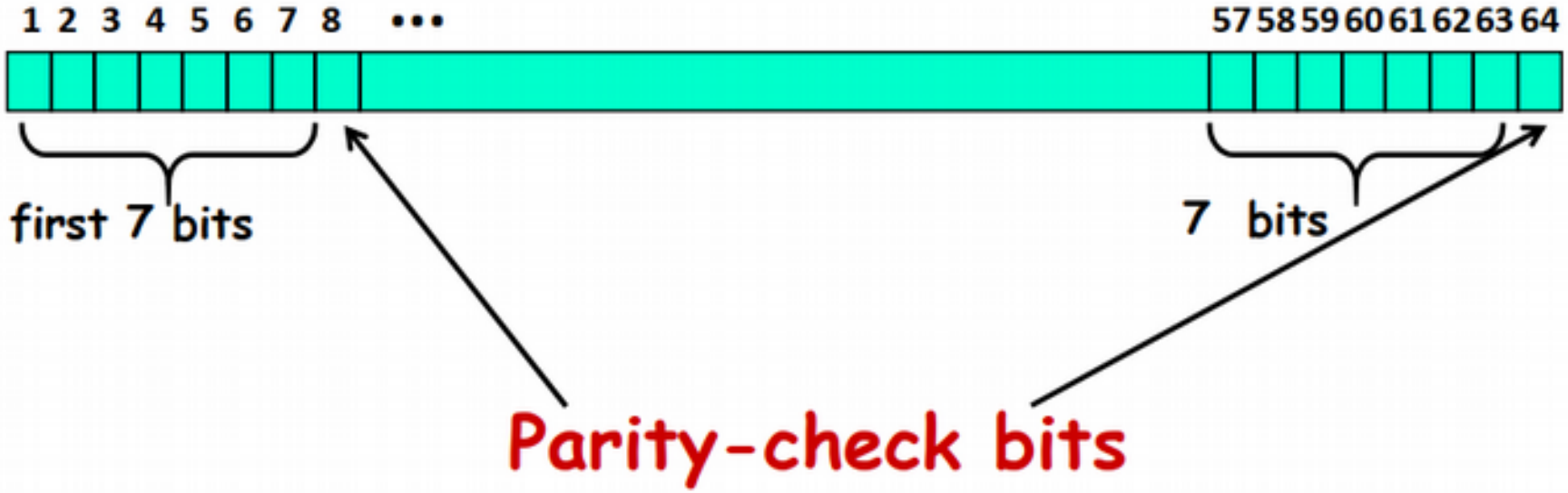
- Features:
  - Block size = 64 bits
  - Key size = 56 bits (in reality, 64 bits, but 8 are used as parity-check bits for error control, see next slide)
  - Number of rounds = 16
  - 16 intermediary keys, each 48 bits



8

Key length in DES

- In the DES specification, the key length is 64 bit:
- 8 bytes; in each byte, the 8th bit is a parity-check bit



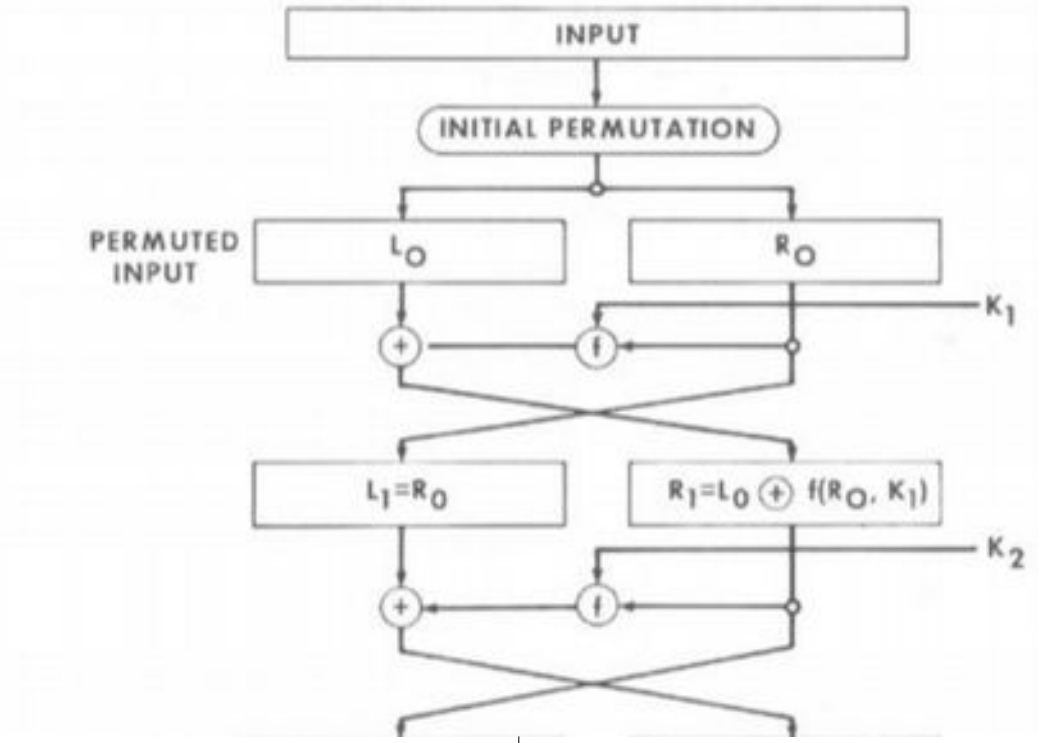
Each parity-check bit is the XOR of the previous 7 bits

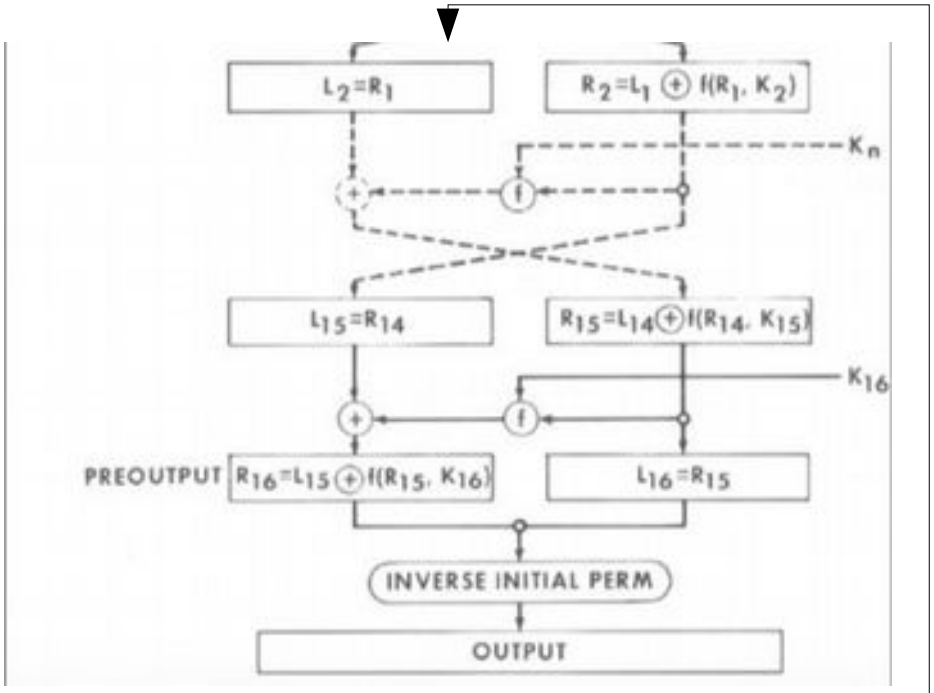
DES – Key Size

- 56-bit keys have 2^{56} = 7.2 x 1016 values
- Brute force search looks hard
- Recent advances have shown that this is possible
  - in 1997 on Internet in a few months
  - in 1998 on DES Cracker dedicated h/w (EFF) in a less than 3 days (cost: $250,000)
  - in 1999 on Internet in a few hours
    - in 2010 above on Internet in a few minutes

Now we have alternatives to DES

DES


INPUT
INITIAL PERMUTATION
PERMUTED INPUT
L_0
R_0
K_1
$L_1=R_0$
$R_1=L_0 \oplus f(R_0, K_1)$
K_2


$L_2=R_1$
$R_2=L_1 \oplus f(R_1, K_2)$
K_n
$L_{15}=R_{14}$
$R_{15}=L_{14} \oplus f(R_{14}, K_{15})$
K_{16}
PREOUTPUT
$R_{16}=L_{15} \oplus f(R_{15}, K_{16})$
$L_{16}=R_{15}$
INVERSE INITIAL PERM
OUTPUT

Triple-DES with Two-Keys

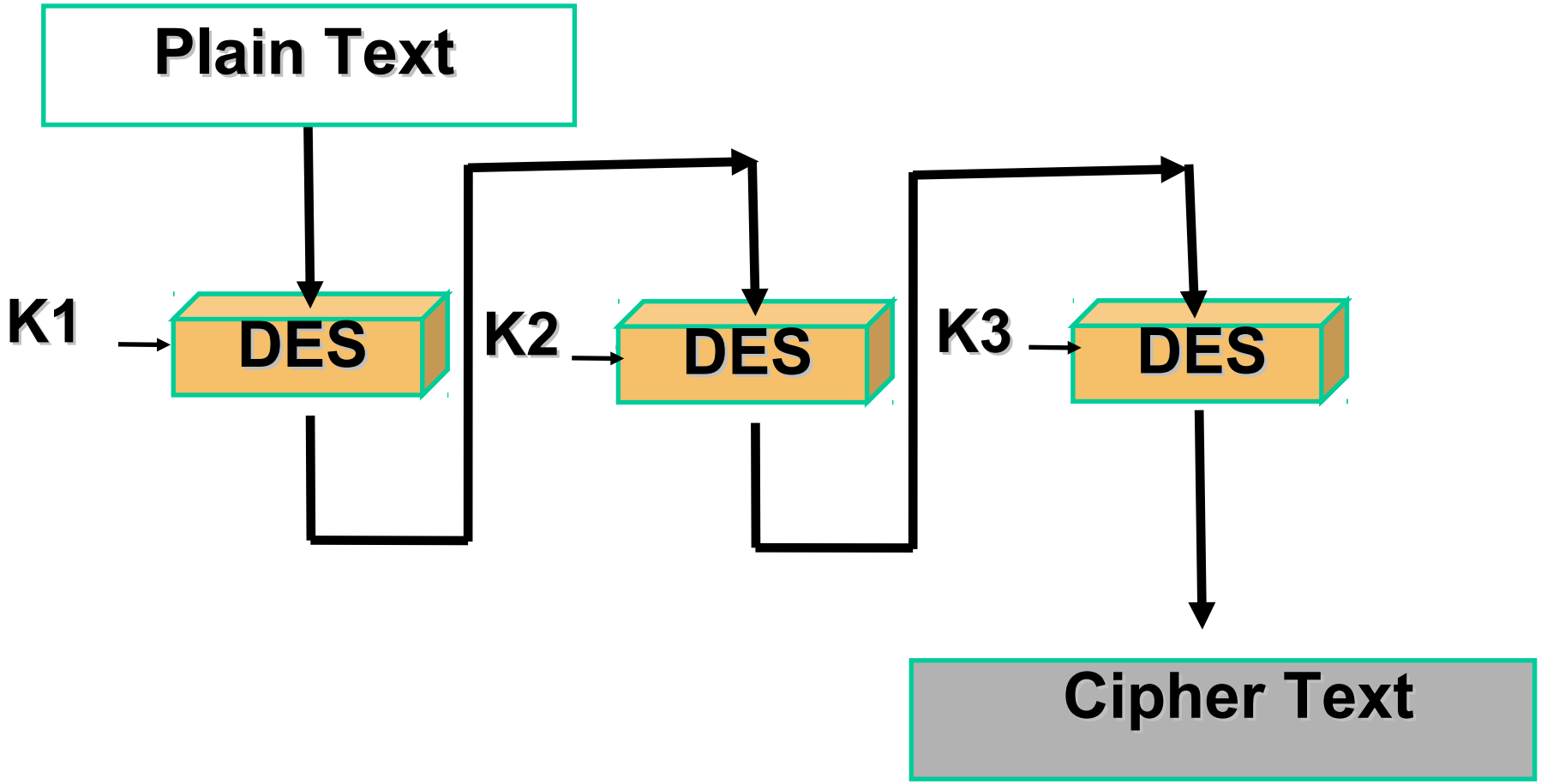
- Use 3 encryptions
  - would seem to need 3 distinct keys
  - But can use 2 keys with E-D-E sequence

C = EK1[DK2[EK1[P]]]

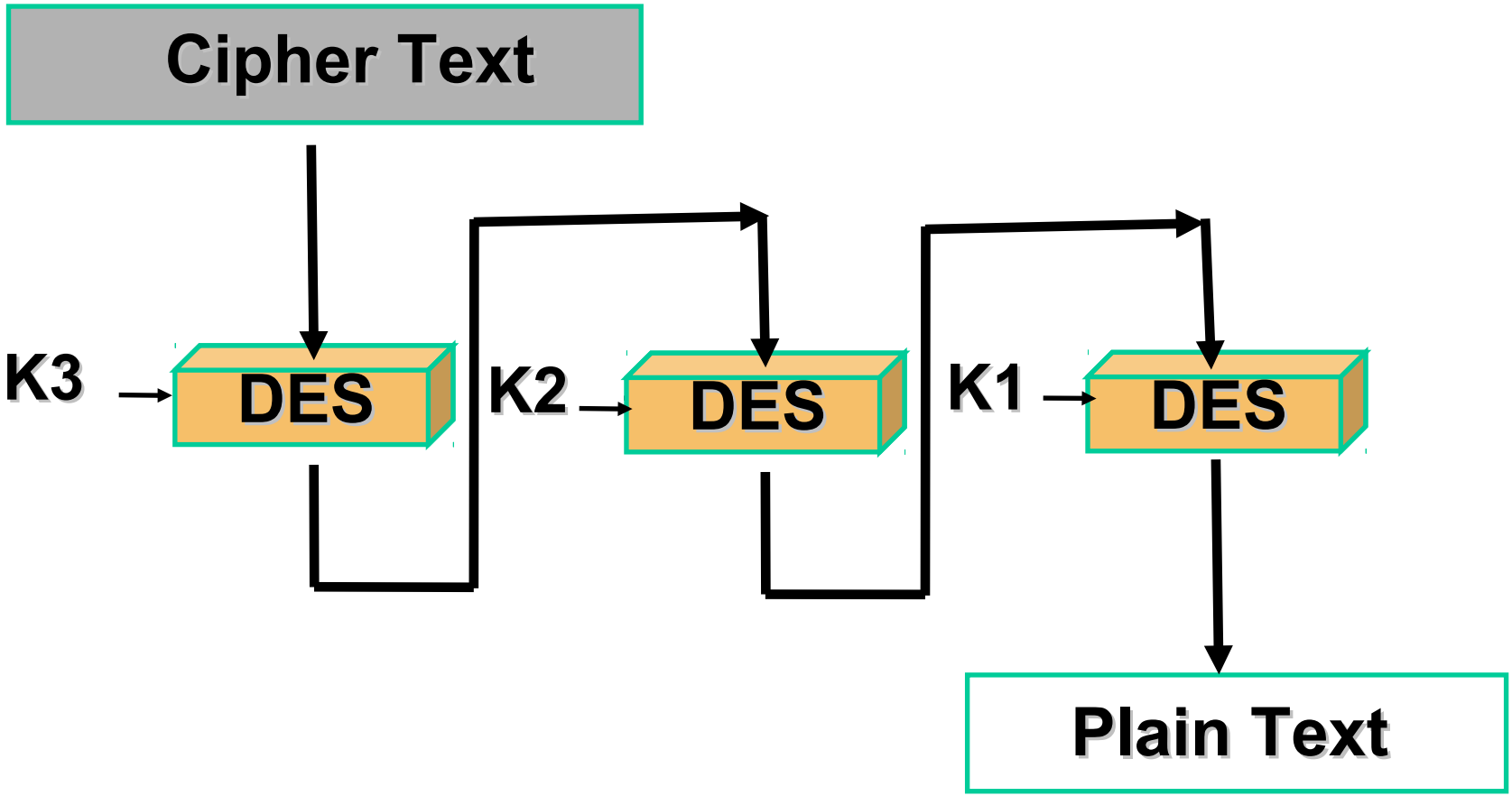
  - Note: encrypt & decrypt equivalent in security
  - if K1=K2 then can work with single DES

- Standardized in ANSI X9.17 & ISO8732
- No current known practical attacks

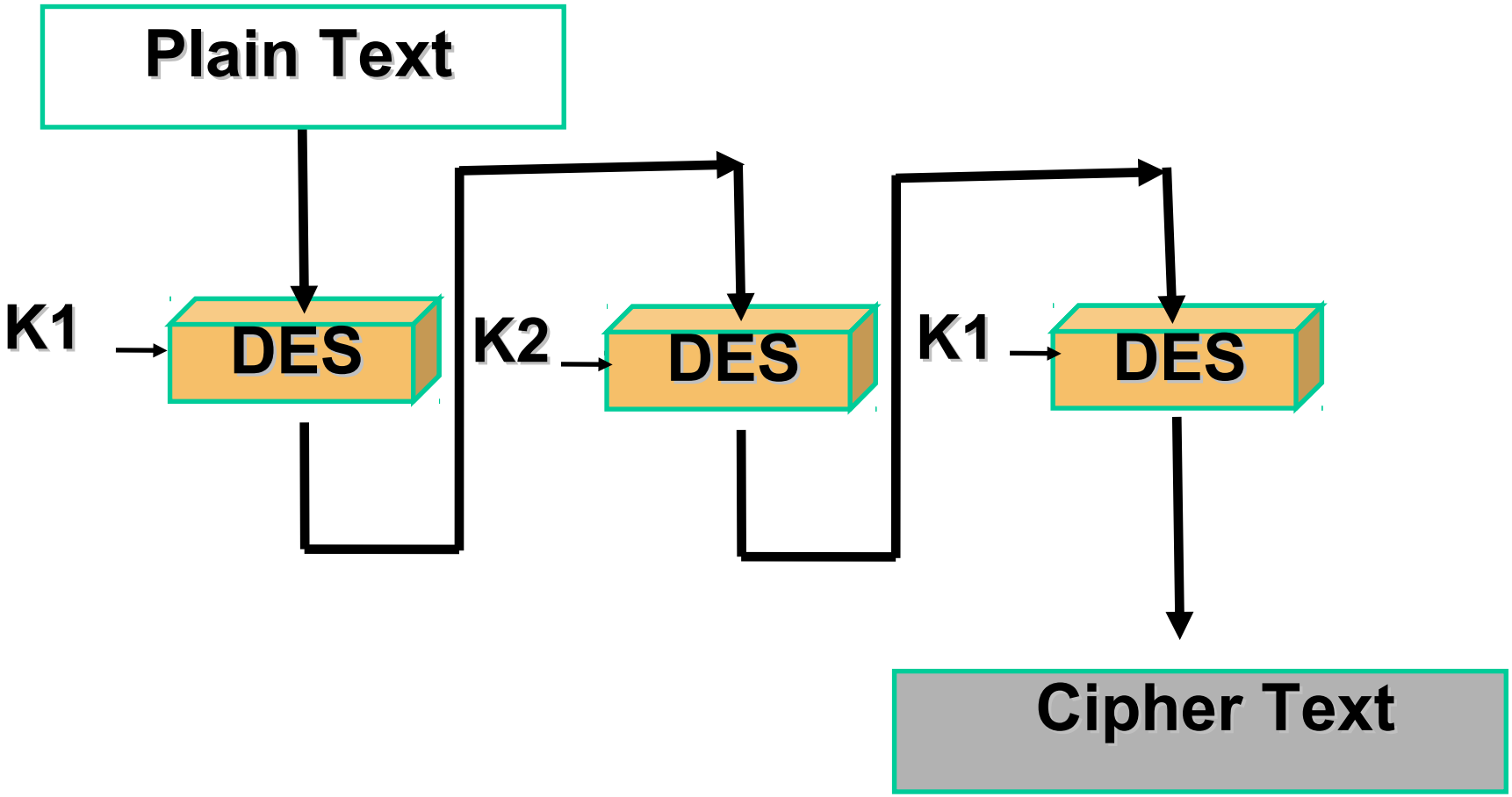
Triple DES - Encryption



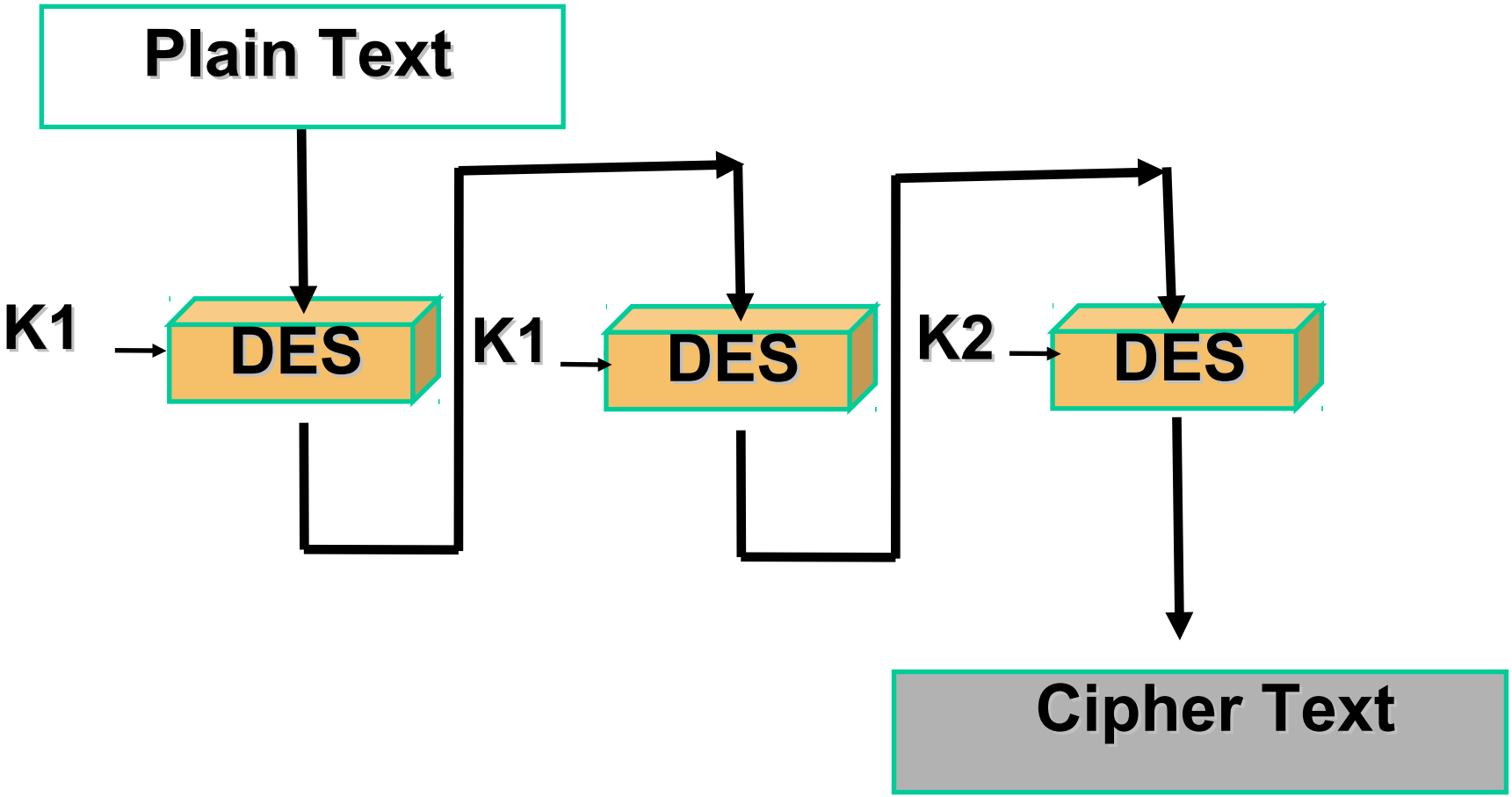
Triple DES - Decryption



Triple DES with Two Keys



Triple DES Backward Compatibility

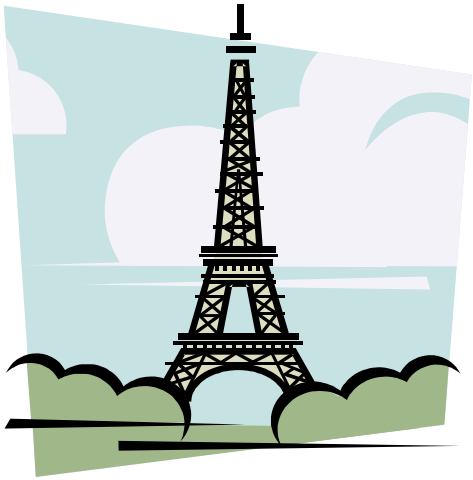


DES- AES

- Clearly, a replacement for DES was needed
  - have theoretical attacks that can break it
  - have demonstrated exhaustive key search attacks
- Can use Triple-DES – but slow with small blocks
- NIST issued a call for ciphers in 1997
- 15 candidates accepted in June 1998
- 5 were short listed in August 1999
- Rijndael was selected as the AES in October 2000
- Issued as FIPS PUB 197 standard in November 2001

AES Requirements

- Private key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger & faster than Triple-DES
- Active life of 20-30 years (+ archival use)
- Provide full specification & design details
- NIST has released all submissions & unclassified analyses

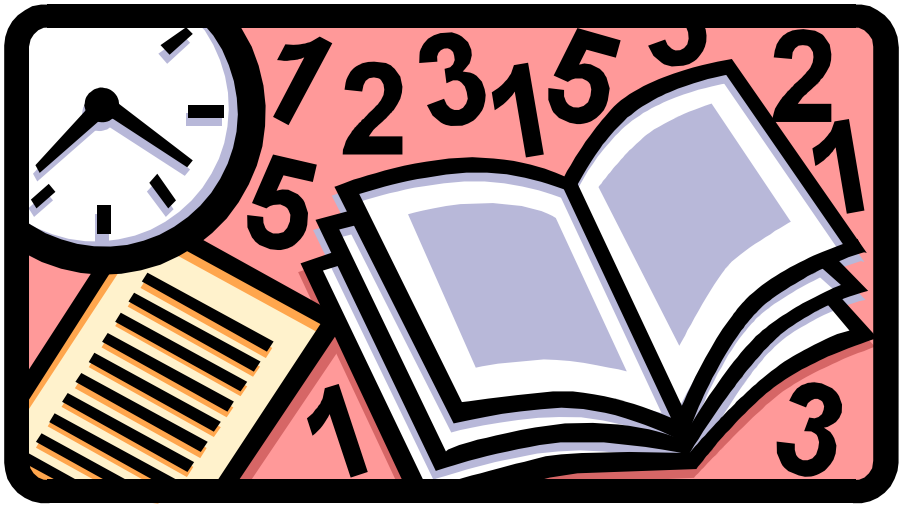


AES Shortlist

- After testing and evaluation, shortlist in August 1999:
  - MARS (IBM) - complex, fast, high security margin
  - RC6 (USA) - v. simple, v. fast, low security margin
  - Rijndael (Belgium) - clean, fast, good security margin
  - Serpent (Euro) - slow, clean, v. high security margin
  - Twofish (USA) - complex, v. fast, high security margin

Advance Encryption Standard (AES)

- In 2001, National Institute of Standards and Technology (NIST) issued AES known as FIPS 197
- AES is based on Rijndael proposed by Joan Daemen, Vincent Rijmen from Belgium

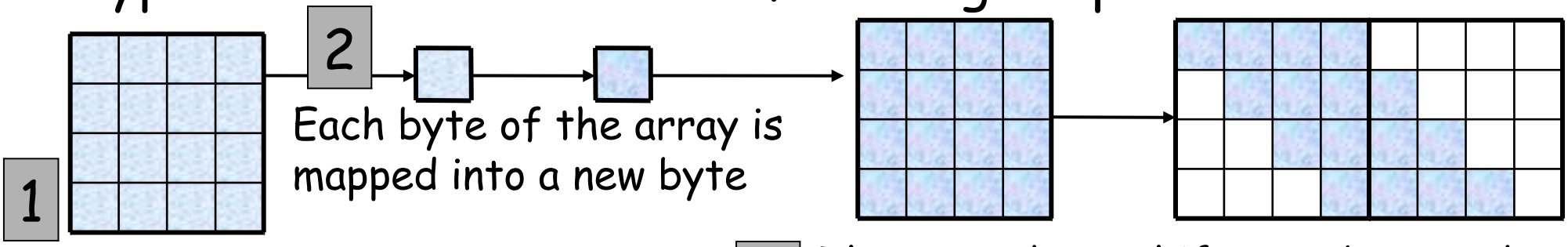


Advance Encryption Standard (AES)

- AES has block length 128
- Supported key lengths are 128, 192 and 256
- AES requires 10 rounds of processing
- Key is expanded into 10 individual keys
- Decryption algorithm uses the expanded keys in reverse order
- Decryption algorithm is not identical to the encryption algorithm

Advance Encryption Standard (AES)

A Typical round includes the following steps



1 The input to each round is a 4-by-4 array of 128 bit plain text.

2 Each byte of the array is mapped into a new byte

3 The second raw shifts one byte and the third raw shifts two bytes and fourth raw shifts three bytes

4 Each column of bytes is mapped into a new column of four bytes

5 An expanded key performs a byte-by-byte exclusive-OR

6 The output of each round is a 4-by-4 byte array of 128 bites

Block Ciphers - Modes of Operation

- Block ciphers encrypt fixed size blocks
  - E.g. DES encrypts 64-bit blocks, with 56-bit key

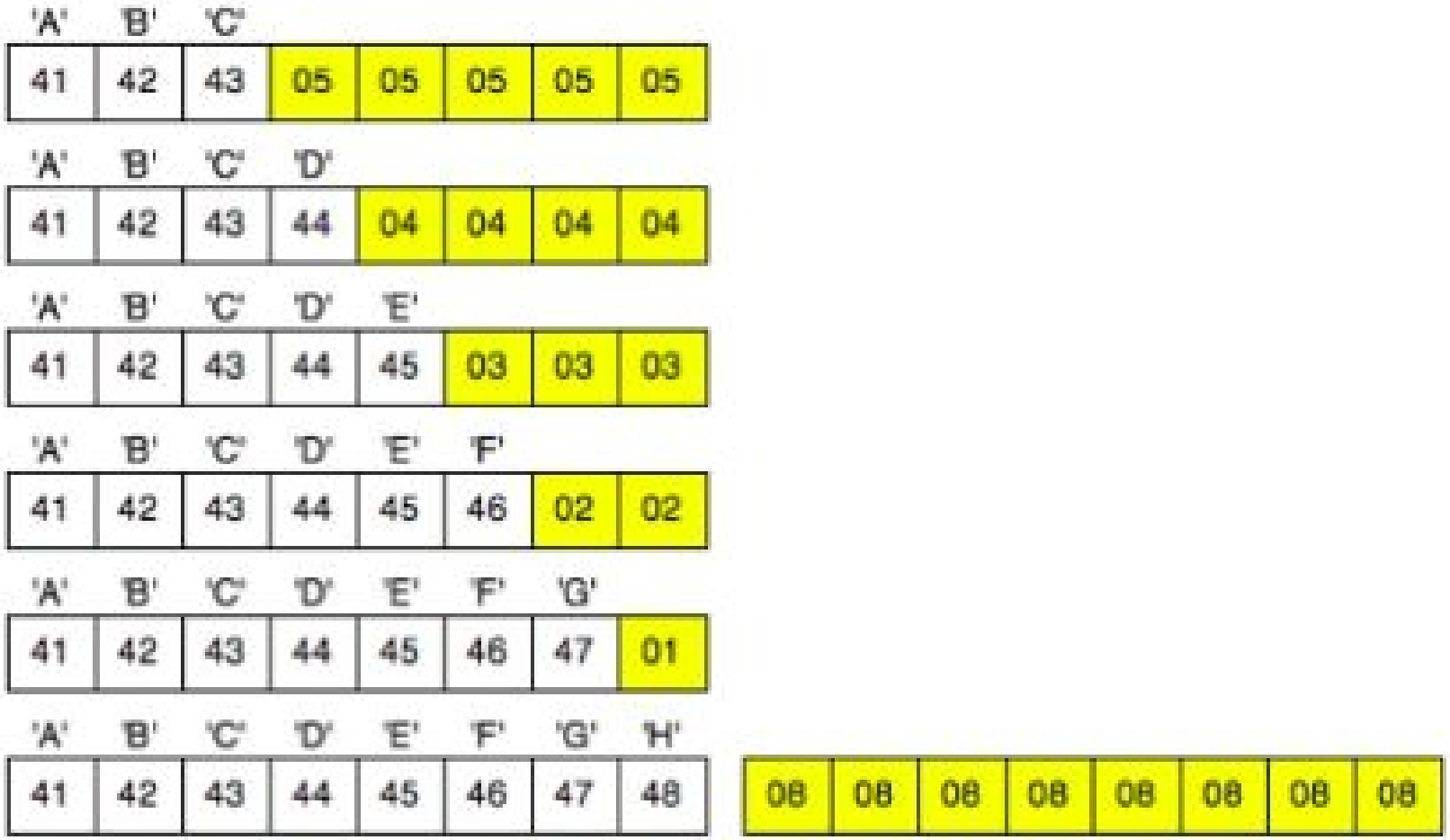
- Given that one needs to encrypt arbitrary amount of information, how do we use in practice,
  - Four modes were defined for DES in ANSI standard
  - **ANSI X3.106-1983 Modes of Use**
  - Subsequently now have 5 for DES and AES

PKCS5 Padding Scheme

- Assume block cipher is 64-bits
- Any message not a multiple of 8 bytes is padded
- Valid pad:
- 1 byte needed: 0x1
- 2 bytes needed: 0x2 0x2
- 3 bytes needed: 0x3 0x3 0x3
- ....
- No padding: 0x8 0x8 0x8 0x8 0x8 0x8 0x8 0x8

  (If the length of the original data is an integer multiple of the block size B, then an extra block of bytes with value B is added.)

PKCS5 Padding Scheme



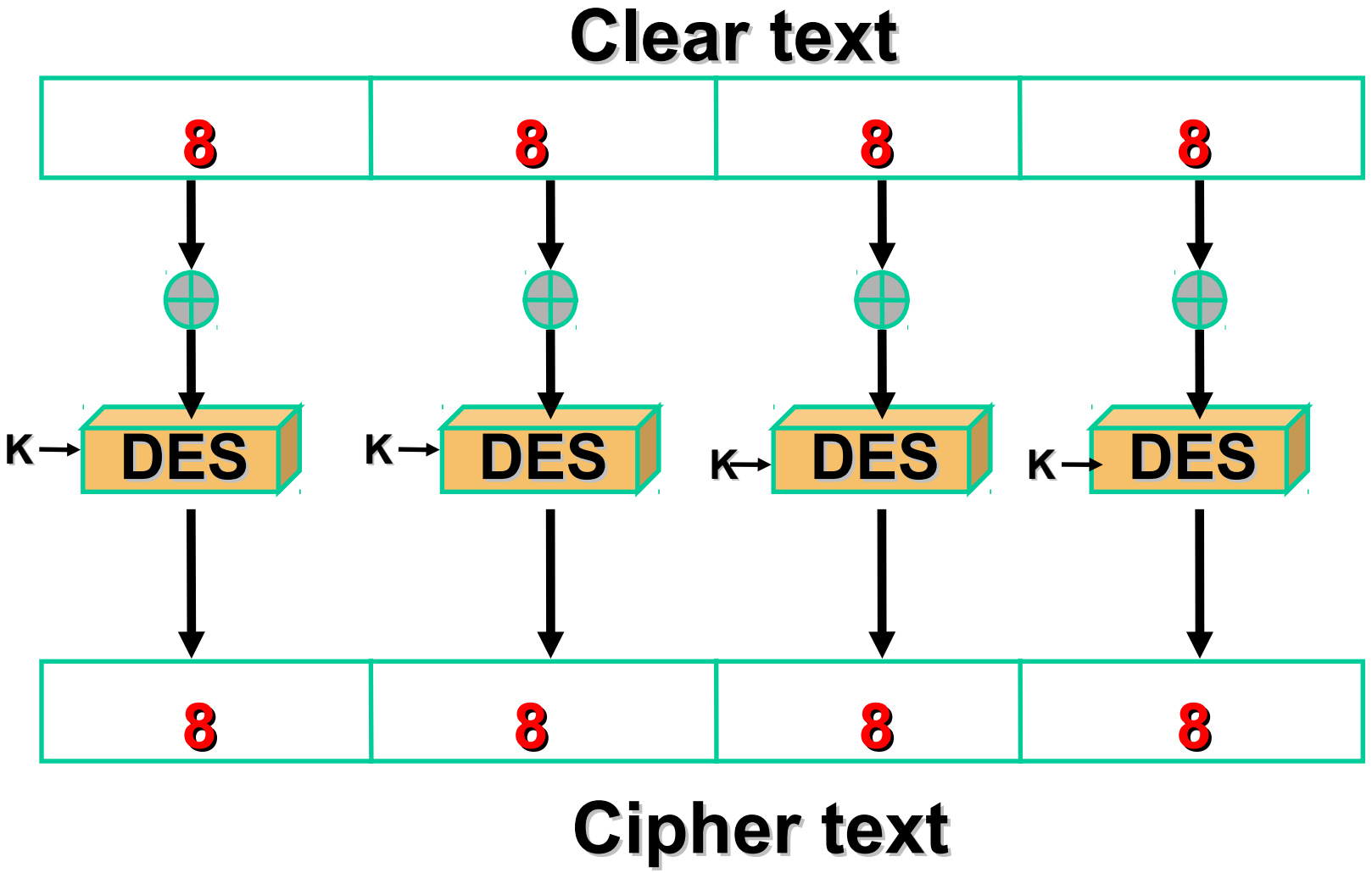
Electronic Codebook Book (ECB)

- Message is broken into independent blocks which are encrypted
- Each block is a value which is substituted, like a codebook, hence name
- Each block is encoded independently of the other blocks

$$C_i = DES_K(P_i)$$

- Uses: secure transmission of single values

Electronic Code Book Mode (ECB)

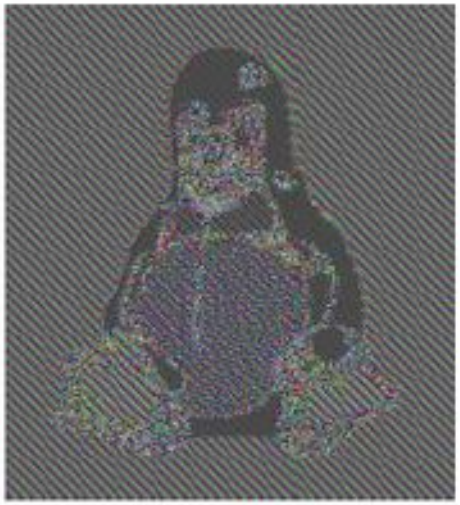


Advantages and Limitations of ECB

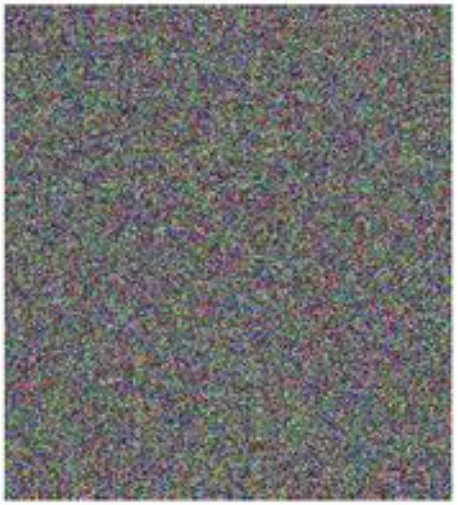
- Repetitions in message may show in ciphertext if aligned with message block particularly with data such graphics or with
- Messages that change very little
- Weakness due to encrypted message blocks being independent
- Main use is sending a few blocks of data

ECB vs CBC

Original



Encrypted using ECB mode



Encrypted using other modes

Electronic codebook (ECB), Cipher block chaining (CBC), Cipher feedback (CFB), Output feedback (OFB)

Cipher Block Chaining (CBC)

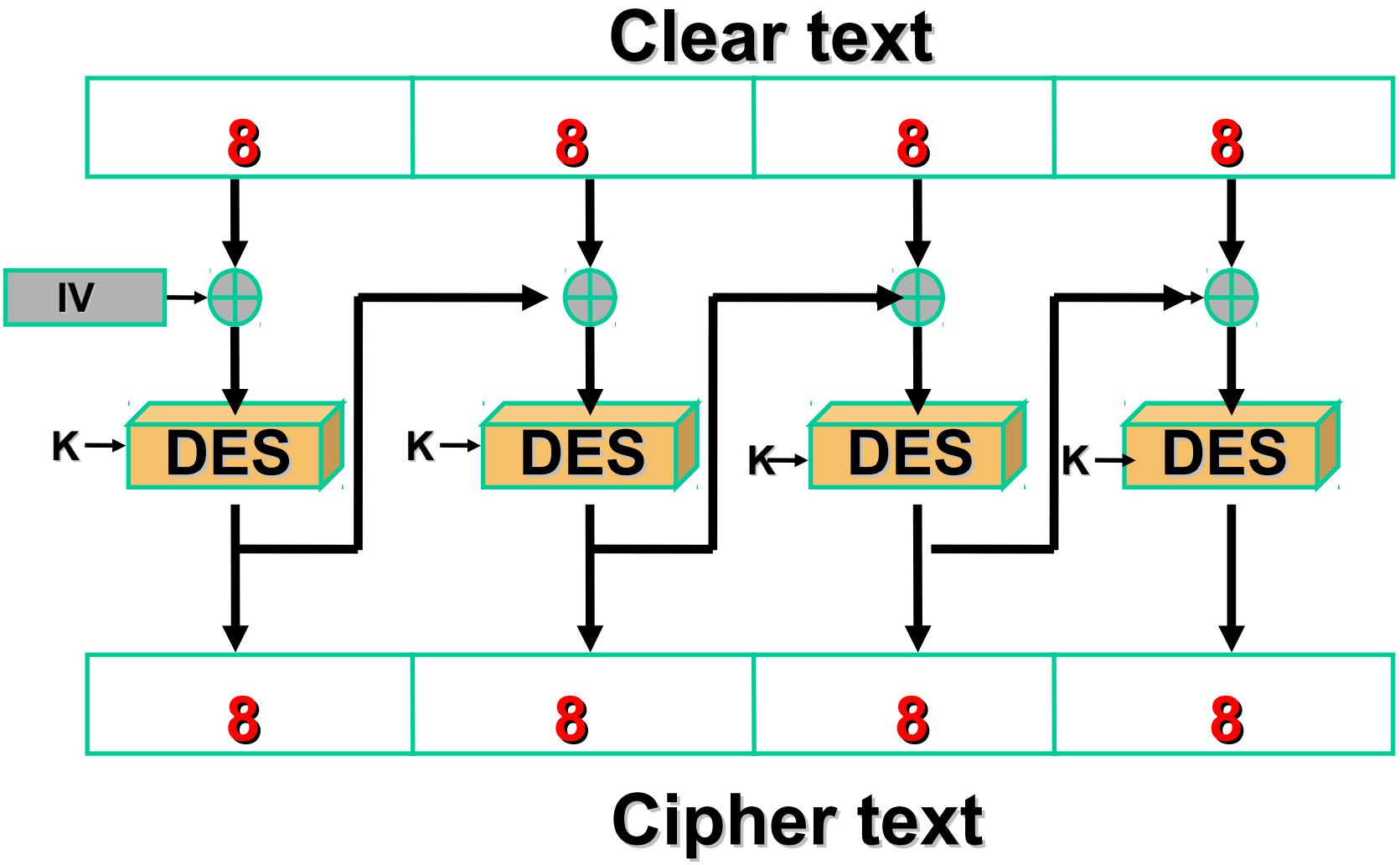
- Message is broken into blocks
- But these are linked together in the encryption operation
- Each previous cipher blocks is chained with current plaintext block, hence name
- Use Initial Vector (IV) to start process

$$C_i = DES_K(P_i \; XOR \; C_{i\text{-}1})$$

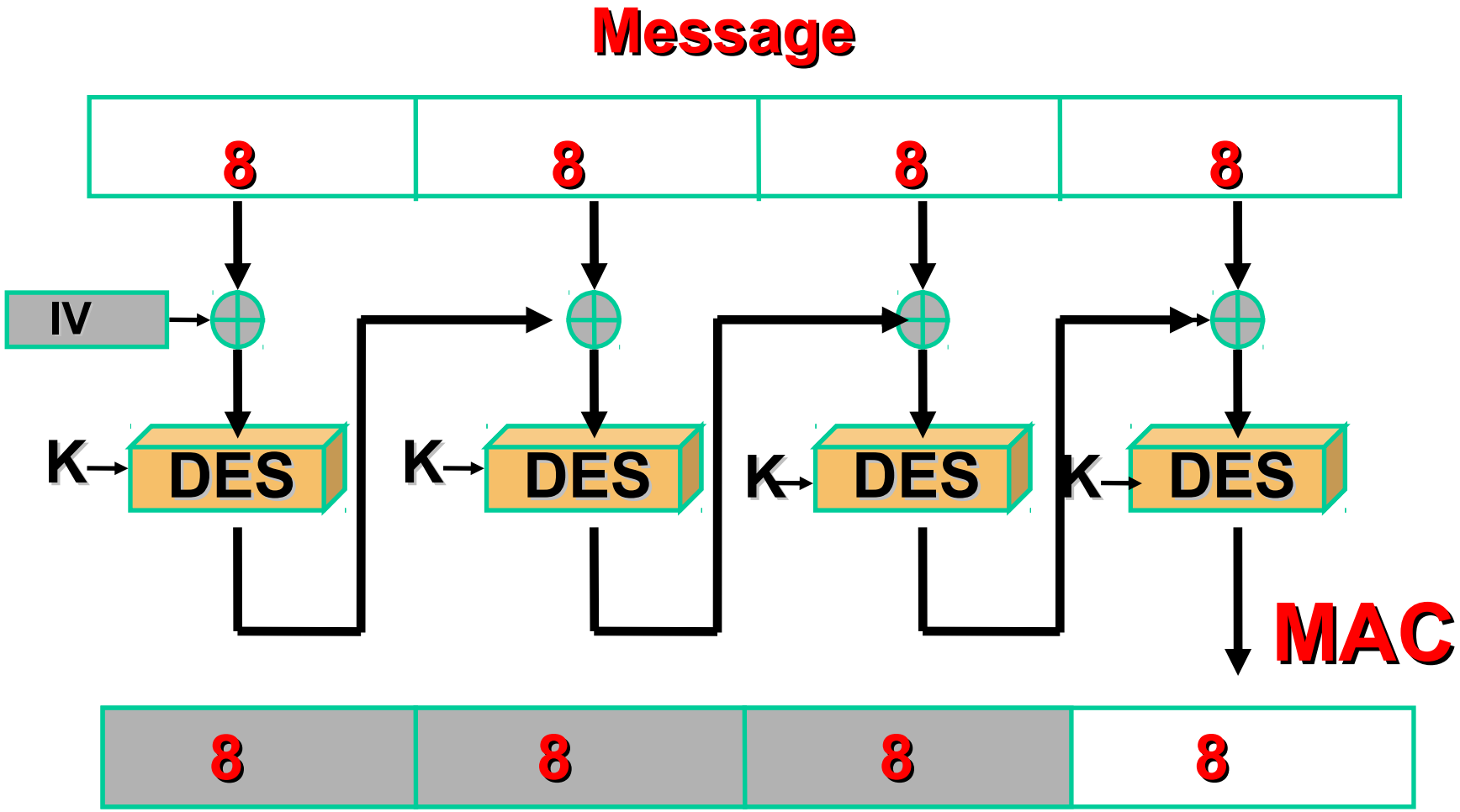
$$C_{-1} = IV$$

- Uses: bulk data encryption, authentication

Cipher Block Chaining Mode (CBC)



MAC based on CBC



Advantages and Limitations of CBC

- Each ciphertext block depends on **all** preceding message blocks thus a change in the message affects all ciphertext blocks after the change as well as the original block

- Need **Initial Value** (IV) known to sender & receiver however if IV is sent in the clear, an attacker can change bits of the first block, and change IV to compensate hence either IV must be a fixed value or it must be sent encrypted in ECB mode before rest of message

- At end of message, handle possible last short block by padding either with known non-data value (e.g. nulls) or pad last block with count of pad size

OpenSSL

 # encrypt file.txt to file.enc using 256-bit AES in CBC mode

>openssl enc -aes-256-cbc -in file.txt -out file.enc

 # decrypt binary file.enc
>openssl enc -d -aes-256-cbc -in file.enc

 # see the list under the 'Cipher commands' heading
>openssl -h

Discussion