1. Cryptographic algorithms

Cryptographic process

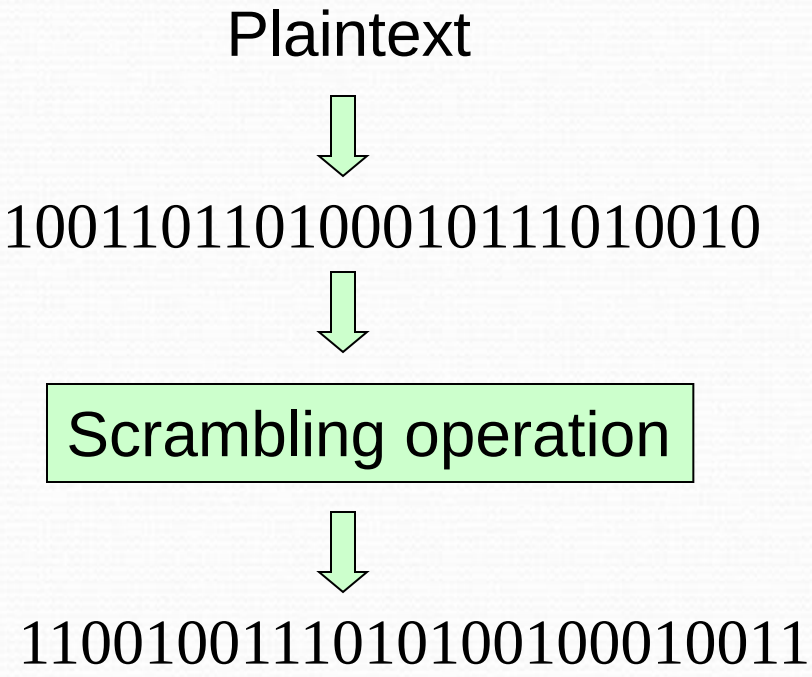
A cryptographic algorithm itself is just one component part of an entire process, which must all come together before we can claim that any particular application is using a “secure cipher system”.

This process includes:

- Selection (or design) of an algorithm
- Deciding how an algorithm is to be used
- Implementing an algorithm within a communications system
- Devising a key management scheme

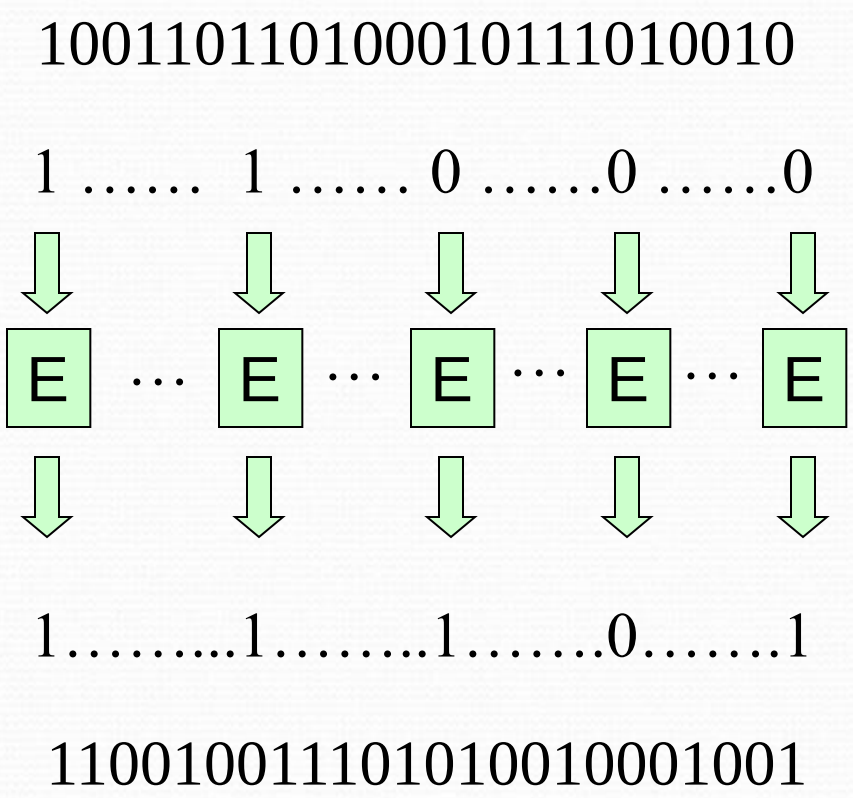
A symmetric classification

Most ciphers systems effective scramble one sequence of binary digits into another:

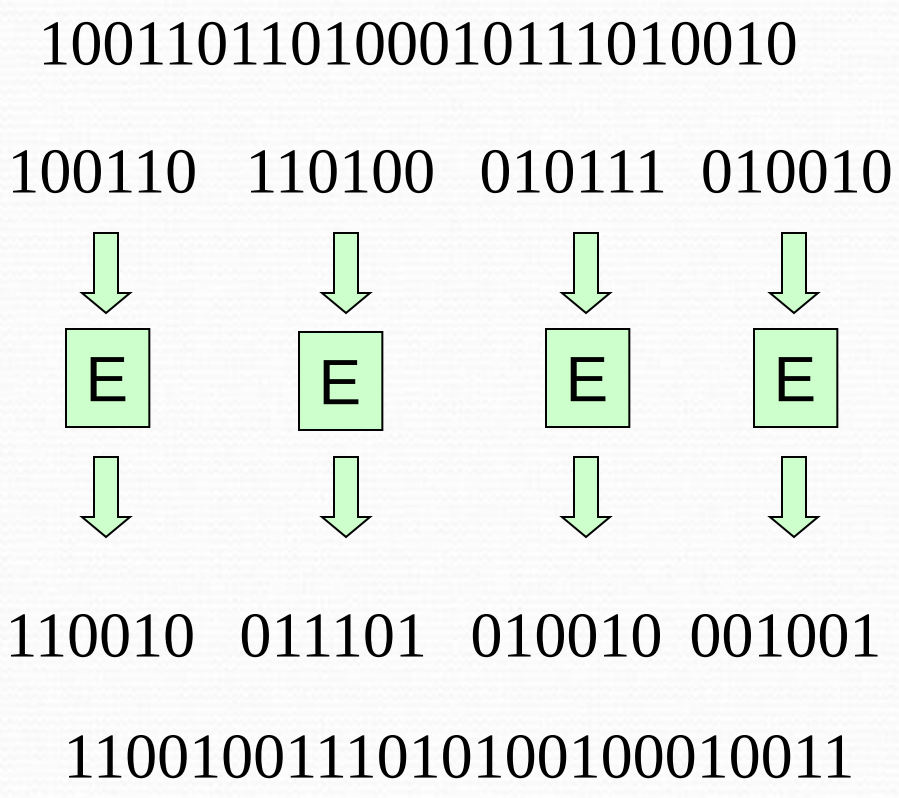


A symmetric classification

Stream cipher



Block cipher



A public key classification?

Can public key cipher systems also be classified into stream and block ciphers?

Error propagation

A decryption process involves **error propagation** if a ciphertext input that has one incorrect bit produces a plaintext output that has more than one incorrect bit.

1. To what extent does error propagation occur in basic stream and block ciphers?
2. Does error **propagation** have anything to with error **prevention** or error **correction**?
3. Is error propagation a good thing?

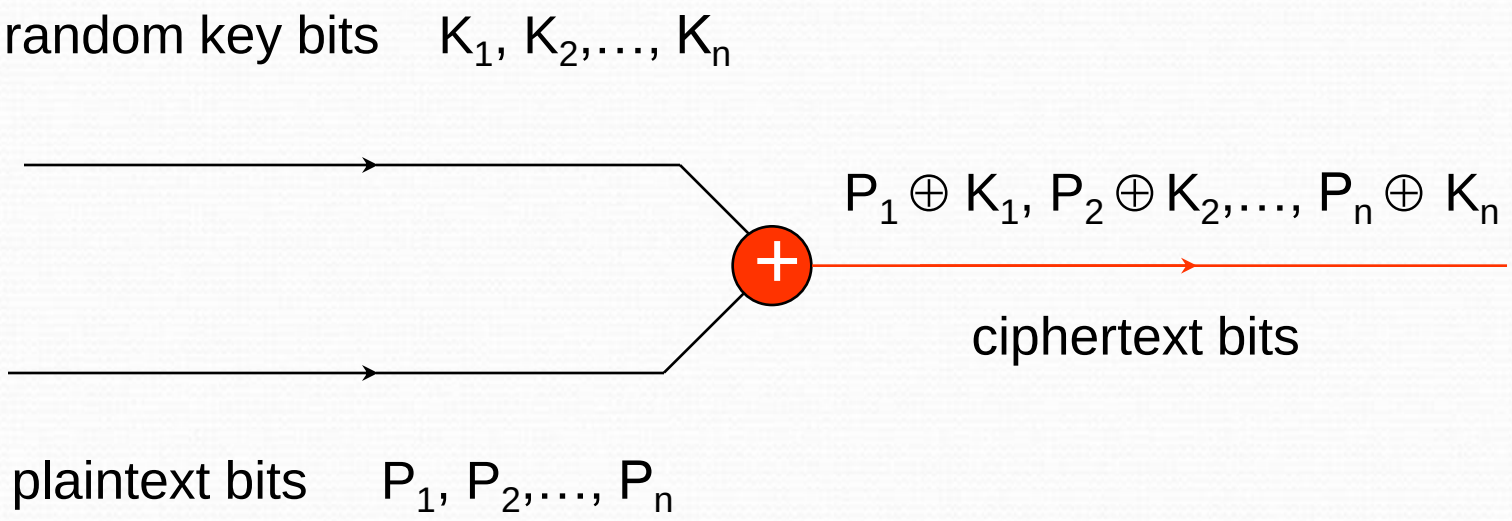
2. Stream ciphers

The idea behind stream ciphers

Recall from Unit 5…

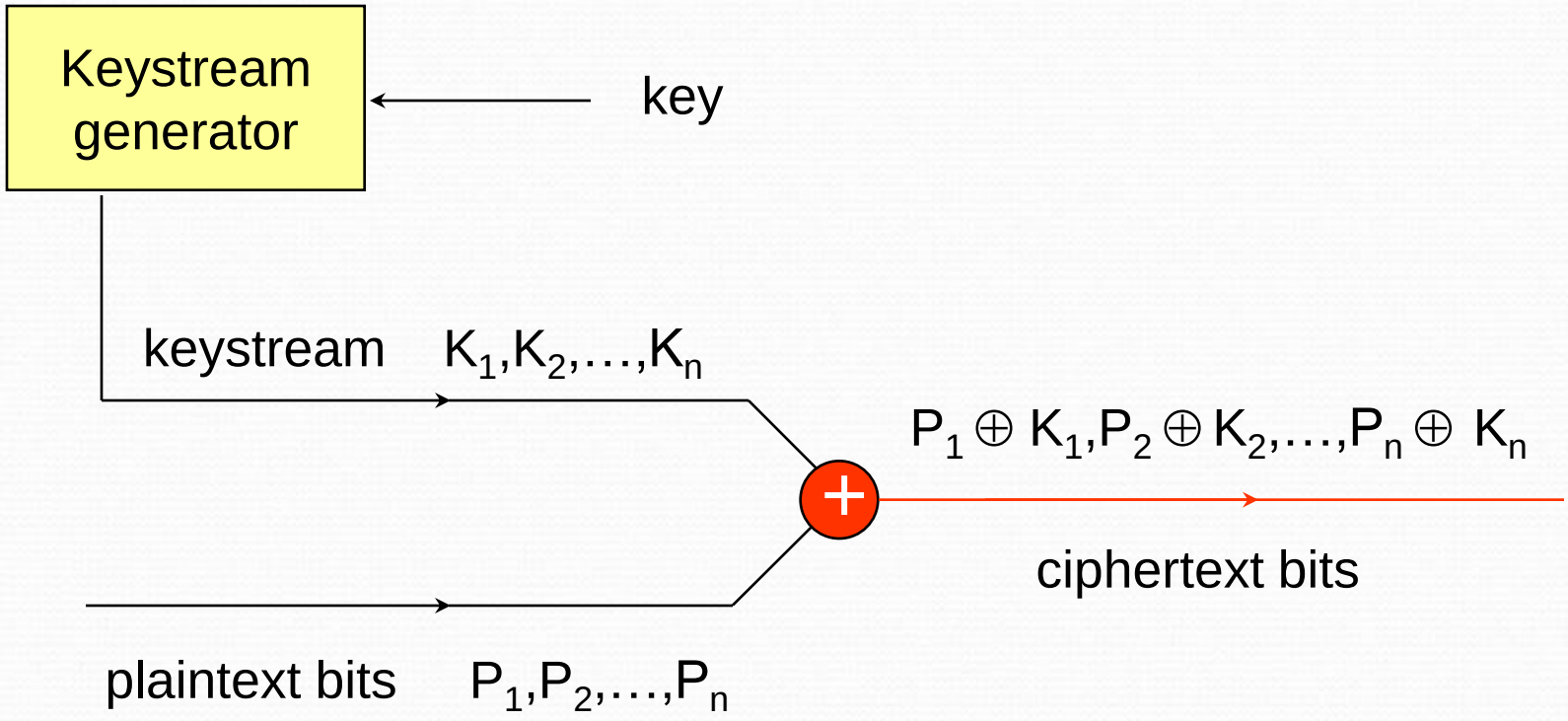
Stream ciphers attempt to simulate the one-time pad by using short keys to generate longer keys that can then be used in a one-time pad encryption

Vernam cipher



How do you decrypt using the Vernam cipher?

Model of a stream cipher



Model of a stream cipher

1. How do you decrypt using a stream cipher?

2. How would go about trying to break:

   a) The Vernam Cipher?

   b) A typical stream cipher?

Security of stream ciphers

Designing good stream ciphers and assessing their security primarily involves analysis of the keystream generator and the properties of the resulting keystream.

Research is focussed on designing keystream generators that produce a keystream that “looks random” (even though it clearly cannot be since it is generated using a deterministic process based on a finite key).

Sequences of ones and zeros that “look random” (but aren’t) are usually referred to as **pseudorandom** sequences.

Pseudorandomness

1. How can you tell if a sequence has good pseudorandom properties?

2. What do you call a stream cipher whose keystream is truly random?

Properties of stream ciphers

PROS

- No error propagation
- Ease of implementation
- Fast

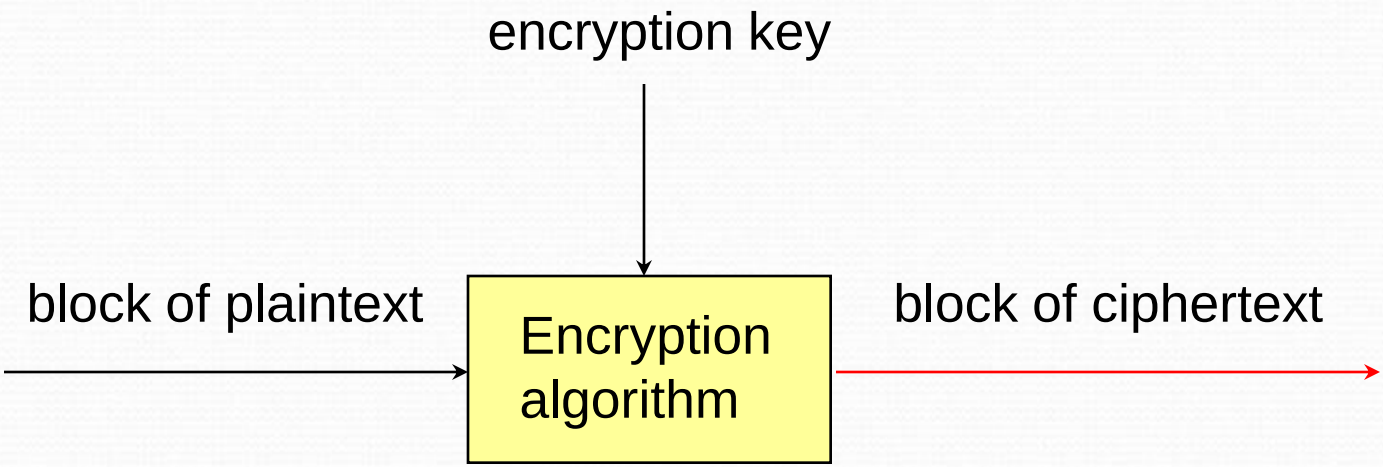
CONS

- (No error propagation)
- Requirement for synchronisation

How might you go about resynchronising the keystream of a stream cipher that had got out of sequence?

3. Block ciphers

Model of a block cipher



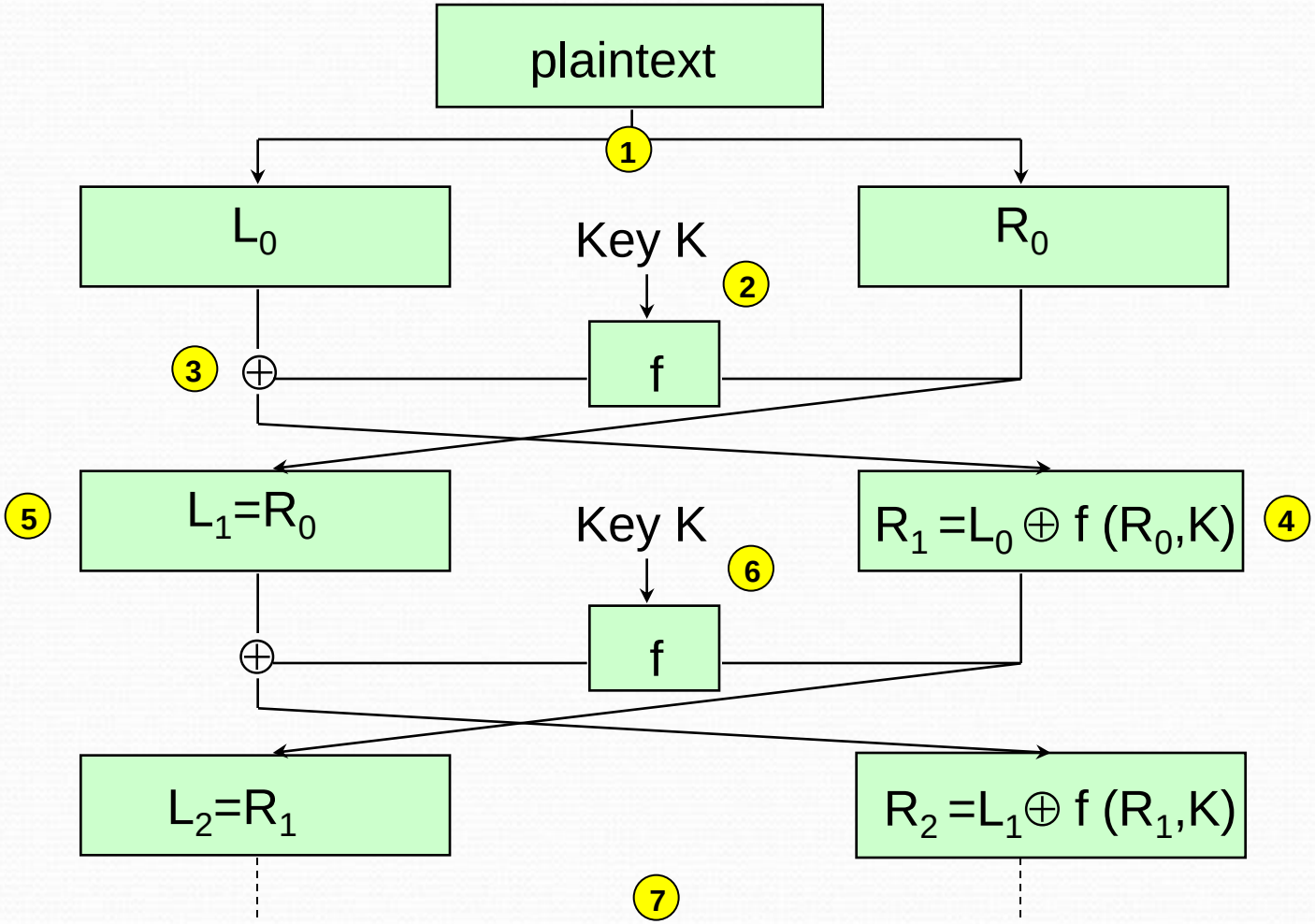
Block size

Normally the block size is fixed, and the block of ciphertext produced by the block cipher is usually also the same length as the plaintext block size.

Typical block sizes are 64 (DES) and 128 (AES).

1. What happens if the block size is “too short”?
2. What happens if the block size is “too long”?
3. Why are most block sizes multiples of 8?

A Feistel cipher



Feistel ciphers

1. How do you choose the round function f ?
2. How do you decrypt using a Feistel cipher?
3. How many rounds should a Feistel cipher have?

In order to understand the cleverness of the Feistel cipher design methodology, attempt Exercises 1 and 2 for this unit.

DES

Parameter	DES specification
Type of design	Feistel Cipher
Number of rounds	16
Block size	64
Length of key	56
Public / proprietary	Published as FIPS 46

Brief history of DES

- In 1973 the National Bureau of Standards (NBS) in the United States published a call for proposals for an encryption algorithm standard.
- IBM was encouraged to submit an encryption algorithm that they had been developing for a second call in 1974.
- After a due consultation process (including NSA) this algorithm was adopted as a federal standard in 1976 and published as DES in 1977.
- DES became mandatory for Federal Agencies in 1977 and after adoption as ANSI X3.92, a banking standard, found widespread use throughout the international financial industry.
- DES essentially now becomes a de facto encryption standard.
- Although DES was predicted to have a 15-year lifespan, the NSA removed its endorsement of DES in 1988.
- The NBS reaffirmed the use of DES in the same year, largely to appease the financial industry, which by then relied heavily upon it.
- NIST finally acknowledged that DES no longer offered adequate cryptographic protection by issuing a new call for an algorithm in 1998.

Design criticisms

Criticism	Comment
Secret design criteria	Design criteria of round function / key schedules secret. (although actual design public) Fear of trapdoors has proved unfounded.
Weak keys	Certain DES keys are **weak**. (encryption and decryption has same effect) Few such keys and their use easily avoided.
Inadequate key length	56 bits an inadequate key length. Criticised even in 1975 Unsubstantiated claims that NSA insisted on the "small" key length.

Searching for a DES key

Suppose that we have a machine consisting of one million processors, each of which can test one million keys per second.

How long is it likely to take before we find a DES key during an exhaustive key search?

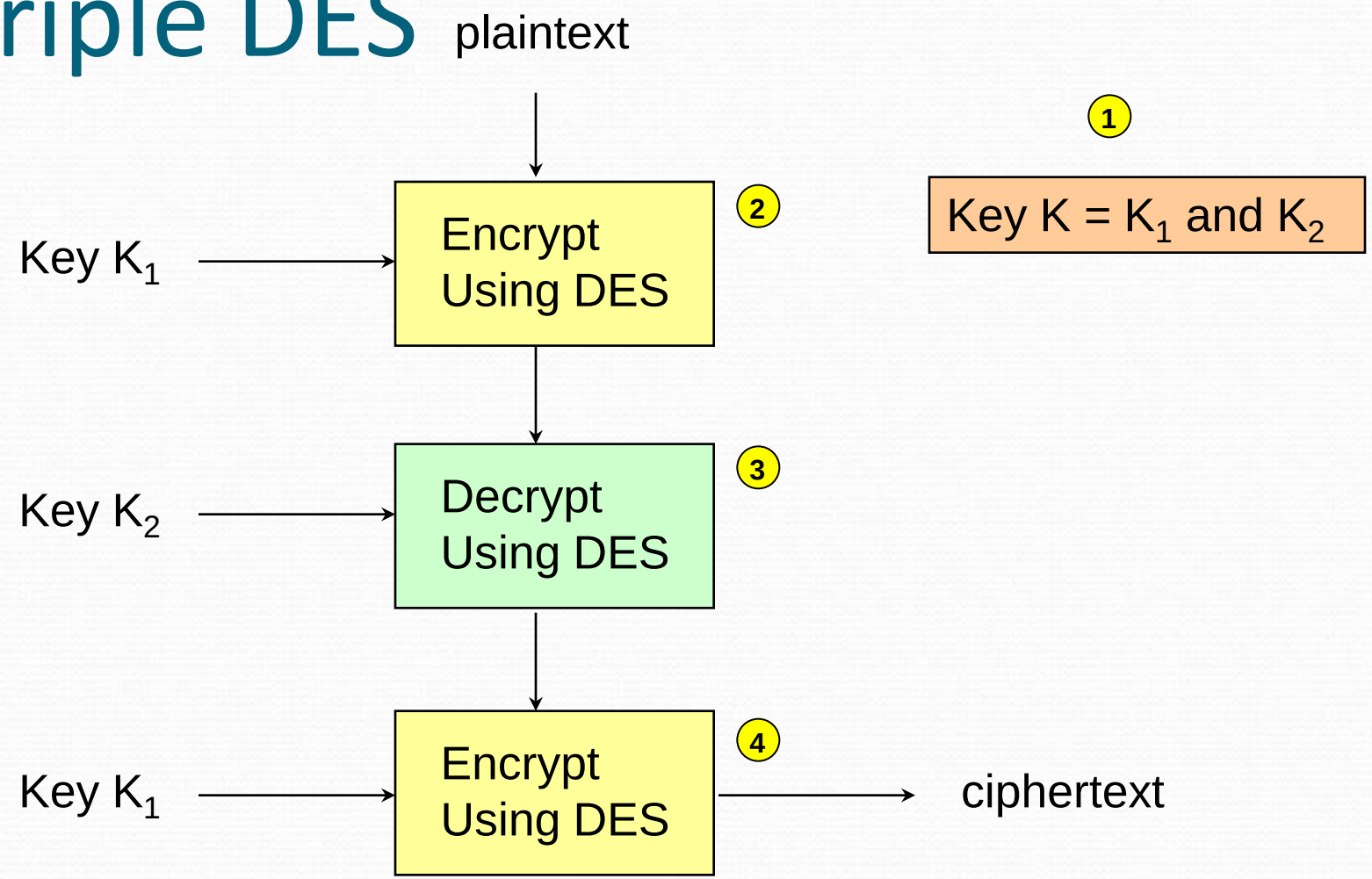
Searching for a DES key

Year	Source	Implemented?	(Estimated) Cost in US$	(Estimated) Search time
1977	Diffie Hellman	No	20 million	20 hours
1993	Wiener	No	10.5 million 1.5 million 600 000	21 minutes 3.5 hours 35 hours
1997	Internet	Yes	Unknown	140 days
1998	Electronic Frontier Foundation [www.eff.org]	Yes	210 000	56 hours

DES today

- Well accepted that a DES key can be found by anyone determined enough.
- Differential and linear cryptanalysis provide academic attacks on DES.
- DES is still in use in many applications.
- Triple DES or AES are commonly recommended instead of DES .

Triple DES


plaintext
1
Key K = K1 and K2
Key K1
Encrypt Using DES
2
Key K2
Decrypt Using DES
3
Key K1
Encrypt Using DES
4
ciphertext

Triple DES

1. Could you encrypt at step 3 of Triple DES instead of decrypting?
2. Could you use a third key K_3 at step 4 of Triple DES, rather than reapplying K_1 ?

Design requirements of AES

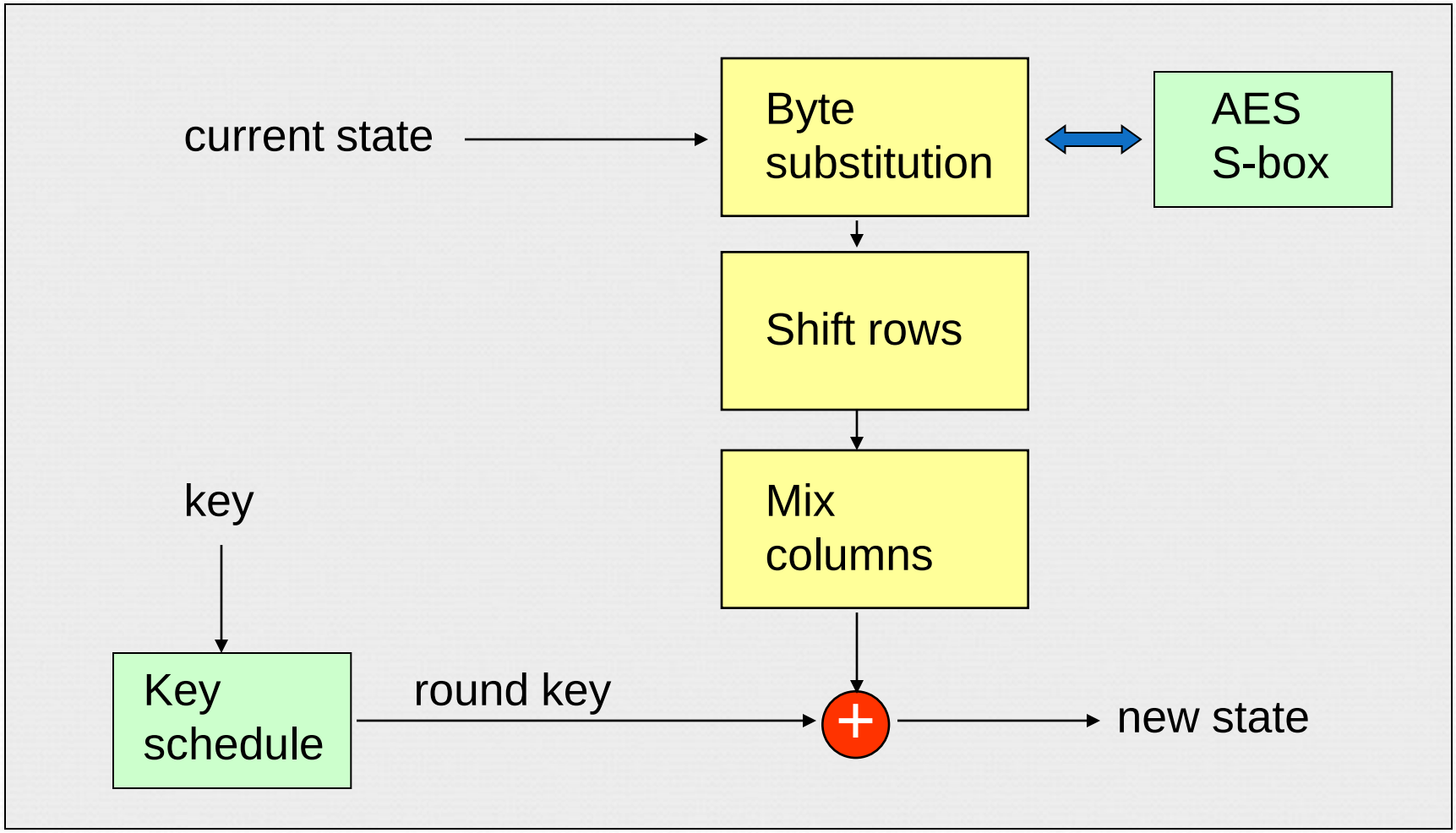
In 1998 NIST issued a call for proposals for a new block cipher standard, to be referred to as the **Advanced Encryption Standard** or **AES**.

- The selection would be a public process and the chosen algorithm and design details would be made freely available for public use.
- The block size should be 128 bits.
- The block cipher would be designed to offer variable key lengths of 128, 192 and 256 bits, to allow for future developments in exhaustive key search efforts.
- The block cipher had to operate at a faster speed than Triple DES across a number of different platforms.

Development of AES

- 15 candidate proposals, quickly reduced to 11 in August 1998.
- In April 1999, after a public consultation process, this was reduced to five candidates: MARS, RC6, Rijndael, SERPENT and TWOFISH.
- In October 2000 the winning algorithm Rijndael was selected.
- Federal Information Processing Standard FIPS 197, the Advanced Encryption Standard, published early 2001. This standard specifies AES (Rijndael) as a FIPS-approved symmetric encryption algorithm that may be used by U.S. Government organizations (and others) to protect sensitive information.
- AES now widely adopted and supported.

AES encryption round



AES

Do you think that the standardisation of AES means the end of Triple DES?

4. Modes of operation

Modes of operation

Modes of operation of a block cipher are operational rules for a generic block cipher that each result in different properties being achieved.

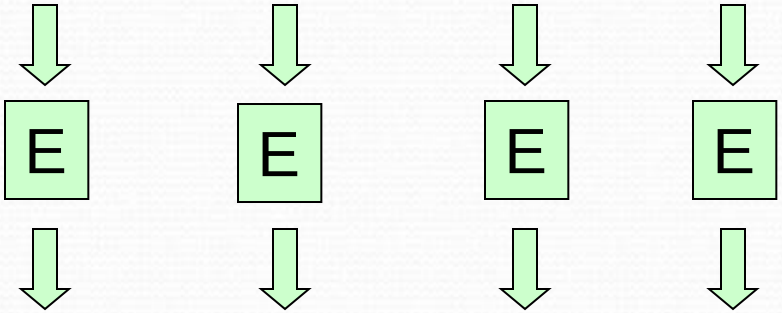
In theory any block cipher could be used in any mode of operation, and the decision concerning which mode of operation to use will in practice be influenced by the application and the properties desired.

The three modes of operation that we will study are not the only modes of operation proposed for block ciphers, but they are three of the most commonly used.

Electronic Code Book (ECB)

100110110100010111010010

100110 110100 010111 010010



110010 011101 010010 001001

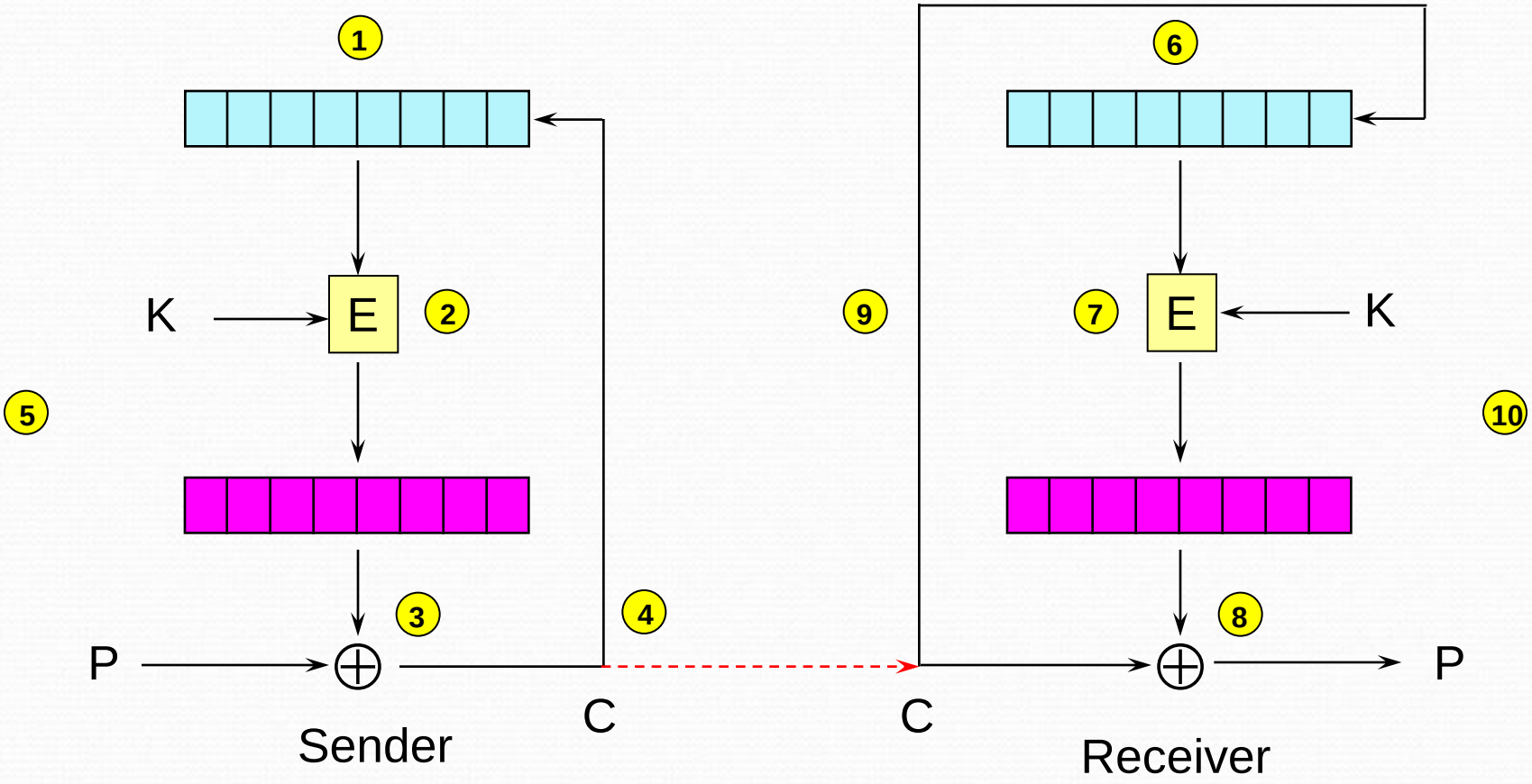
110010011101010010010011

Problem with ECB mode

Try to come up with at least three reasons why ECB mode is rarely used and generally regarded as an insecure mode of operation.

Make sure you attempt Exercise 6 for this unit in order to further appreciate these problems with ECB mode.

Cipher Feedback Mode (CFB)



Cipher Feedback Mode

Paradoxically, when using CFB mode you never actually use the encryption algorithm to **decrypt** anything!

CFB mode is actually using a block cipher to make a sort of stream cipher. The encryption algorithm is used as a keystream generator to produce key material.

This key material is then added to the plaintext very much in the style of a stream cipher.

The receiver also uses the encryption algorithm to generate the same keystream that is needed to decrypt the ciphertext.

CFB in practice

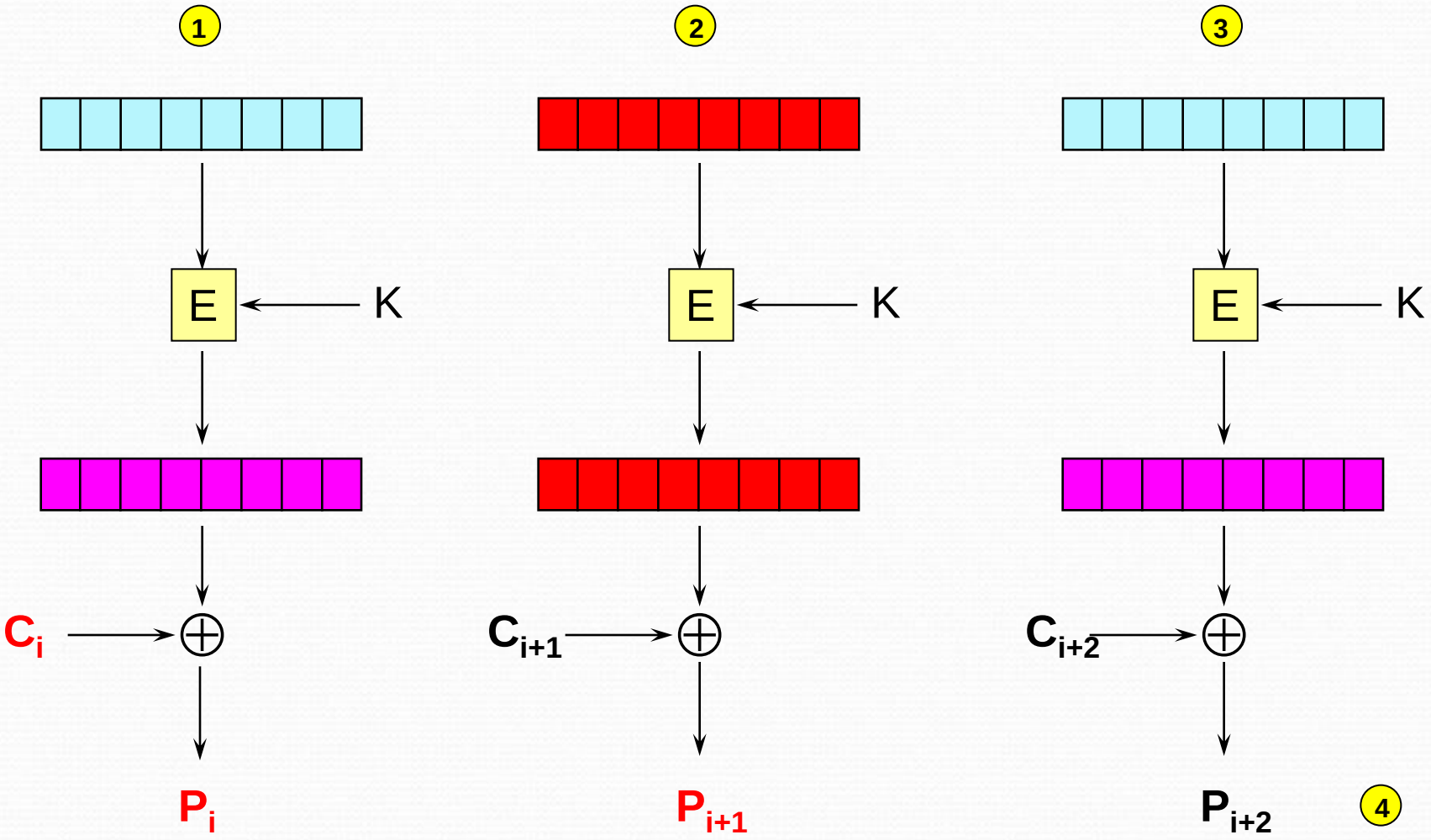
Most practical implementations of CFB mode process the plaintext in units of bits that are smaller than the block size.

For example, when using an encryption algorithm with a block size of 64 in **8-bit CFB mode** the plaintext is processed 8 bits at a time. This produces only 8 bits of ciphertext.

When these 8 bits of ciphertext are fed back, they are not sufficient to replace the current register contents, so the existing entries are shifted along, with the 8 furthest bits dropping out.

For what reasons might you want to use 8-bit CFB mode, rather than full block CFB mode?

Effect of error in CFB mode

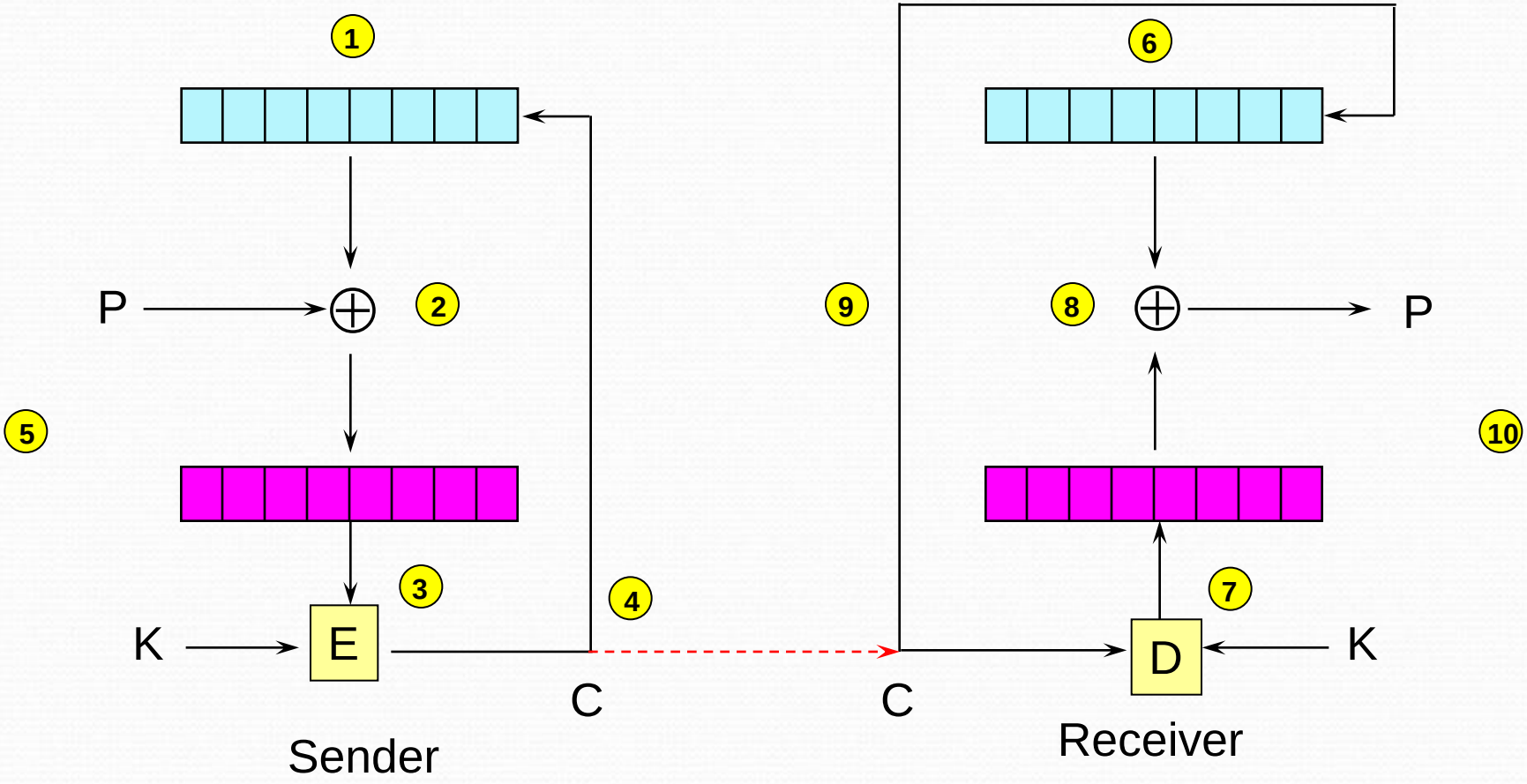


Properties of CFB mode

- Message dependence
- Limited error propagation
- No block synchronisation required
- Efficiency

To what extent do these properties also hold for 8-bit CFB mode?

Cipher Block Chaining (CBC)



CBC mode

All CBC mode is doing is adding each plaintext block to the previous ciphertext block, and then encrypting the result with the key:

$$C_1 = E_K (P_1 \oplus IV)$$

$$C_i = E_K (P_i \oplus C_{i-1})$$

It is worth working through a simple example to get the hang of CBC mode: one such example is provided in Chapter 7 of Piper and Murphy.

Decrypting using CBC mode

First step (7 on diagram):

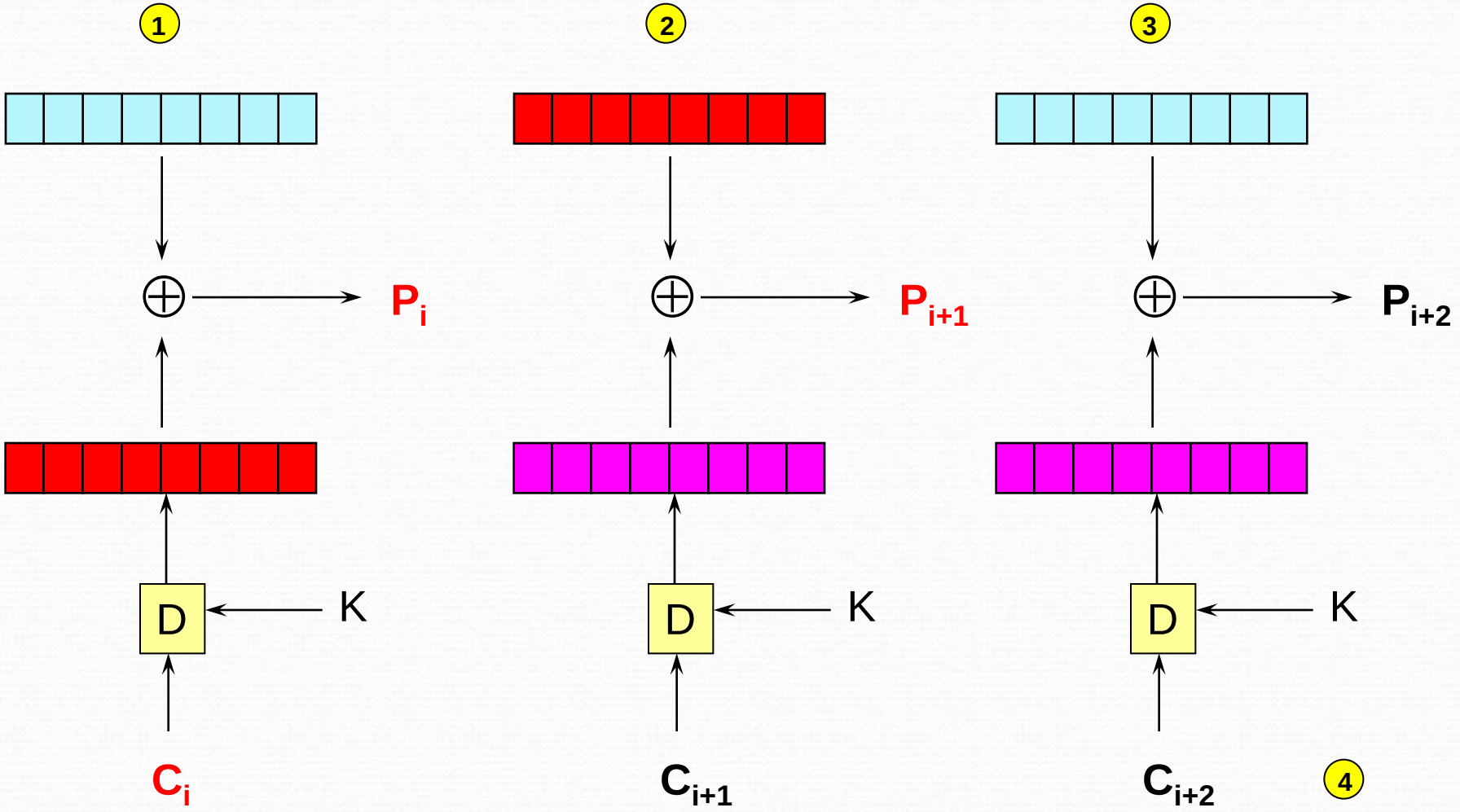
$$D_K(C_i) = D_K(E_K(P_i \oplus C_{i-1}))$$

$$= P_i \oplus C_{i-1}$$

Second step (8 on diagram):

$$(P_i \oplus C_{i-1}) \oplus C_{i-1} = P_i$$

Effect of error in CBC mode



Properties of CBC mode

Recall the properties of CFB mode.

To what extent does CBC mode offer these properties?

We will see later that CBC mode can also be used to design a Message Authentication Code