



I. Criptografía avanzada

Iván García y Pablo López

1. Cifrado simétrico
 - a. Cifrados de flujo vs de bloque
 - b. AES
 - i. ECB
 - ii. CBC
2. Cifrado asimétrico
 - a. RSA

CRIPTOGRAFÍA SIMÉTRICA

¿Qué es la criptografía simétrica?

Es un tipo de cifrado en el que se utiliza la misma clave tanto para cifrar como para descifrar un mensaje.

VENTAJAS

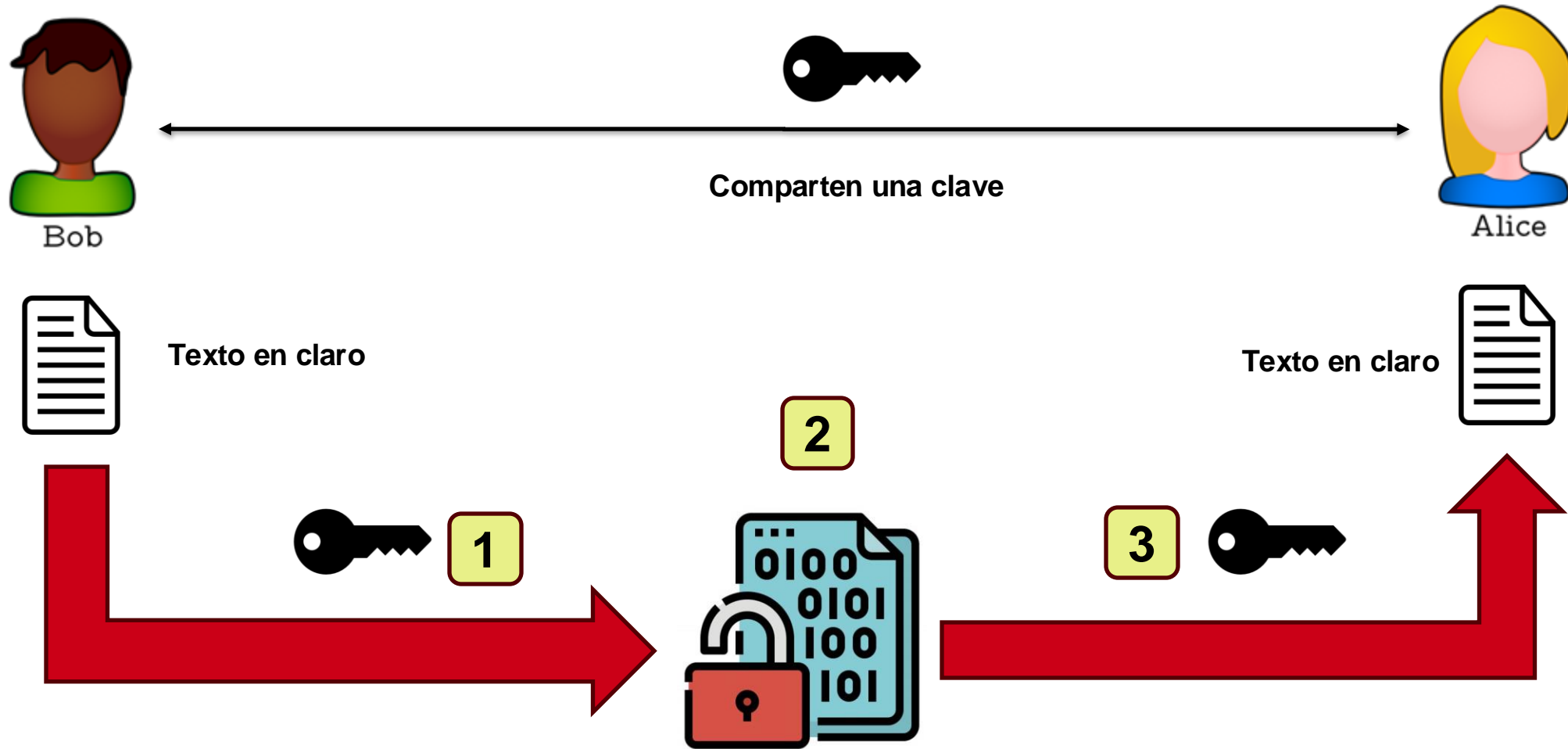
- Muy fácil de usar
- Muy útil
- Rápida y eficiente
- Segura

DESVENTAJAS

- ¿Cómo compartimos la clave?
- Demasiadas claves

CRIPTOGRAFÍA SIMÉTRICA

¿Cómo funciona?



CRIPTOGRAFÍA SIMÉTRICA: FLUJO VS BLOQUE

FLUJO

- Cifrado bit a bit
- XOR(bit, key_bit)
- Más ligero
- Más difícil de implementar de forma segura
- No reutilizar claves

BLOQUE

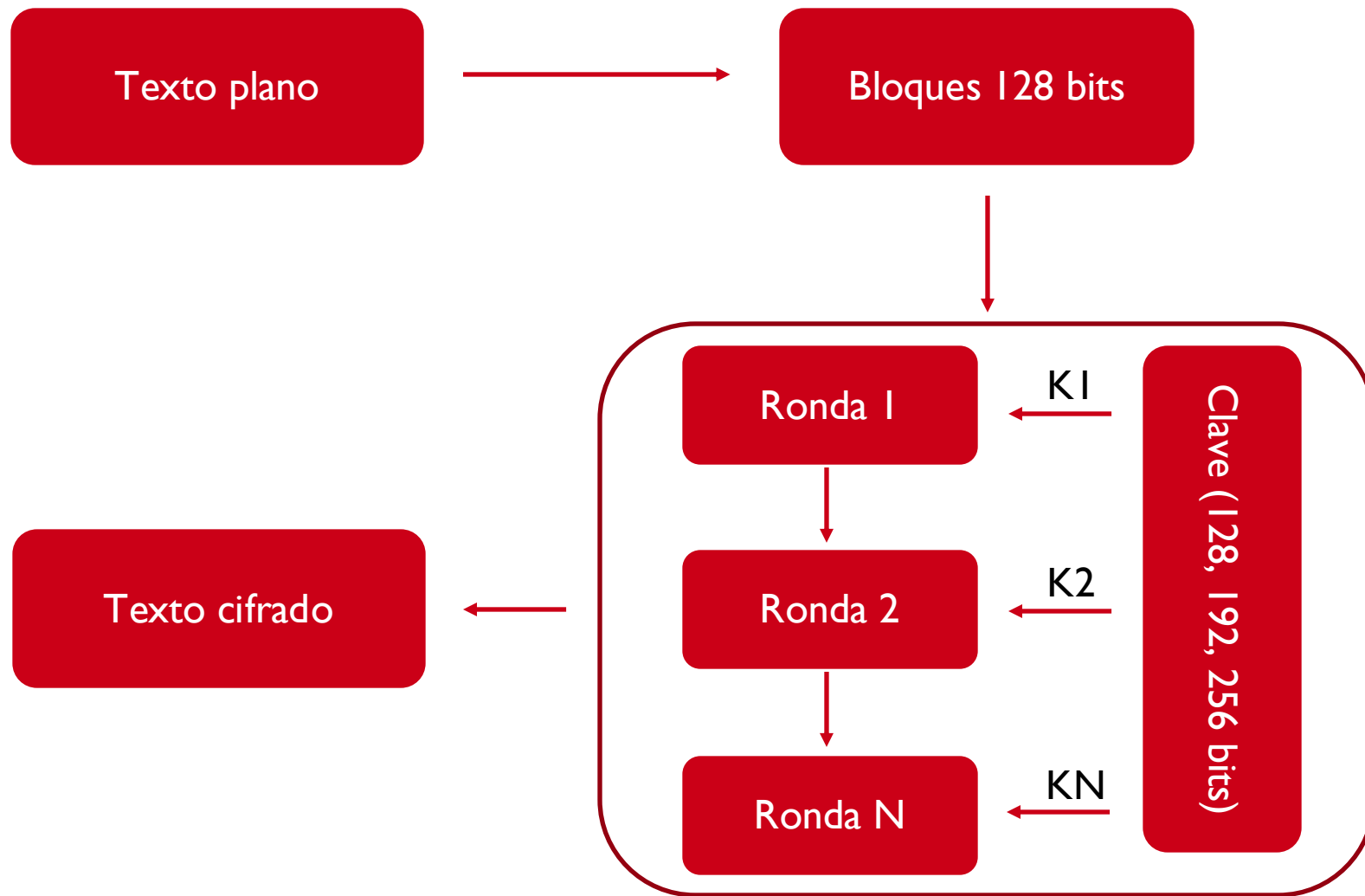
- Cifrado por bloques
- XOR(block, key_block)
- Más pesado
- Más fácil de implementar de forma segura
- Se pueden reutilizar claves (¡EN ALGUNOS MODOS!)

AES – (Advanced Encryption Standard)

¿Qué es?

- Algoritmo de cifrado simétrico de bloque
- Se utiliza como estándar global de encriptación
- Se utiliza en aplicaciones como WhatsApp y Signal
- 128, 192 y 256 bits
- Evolución de DES --> más lento y más inseguro (clave corta)

¿Cómo funciona AES?



¿Cómo funciona AES?

- En cada ronda, se calcula una **nueva clave** a partir de la **original**
- El **número de rondas** depende de la **longitud de la clave**

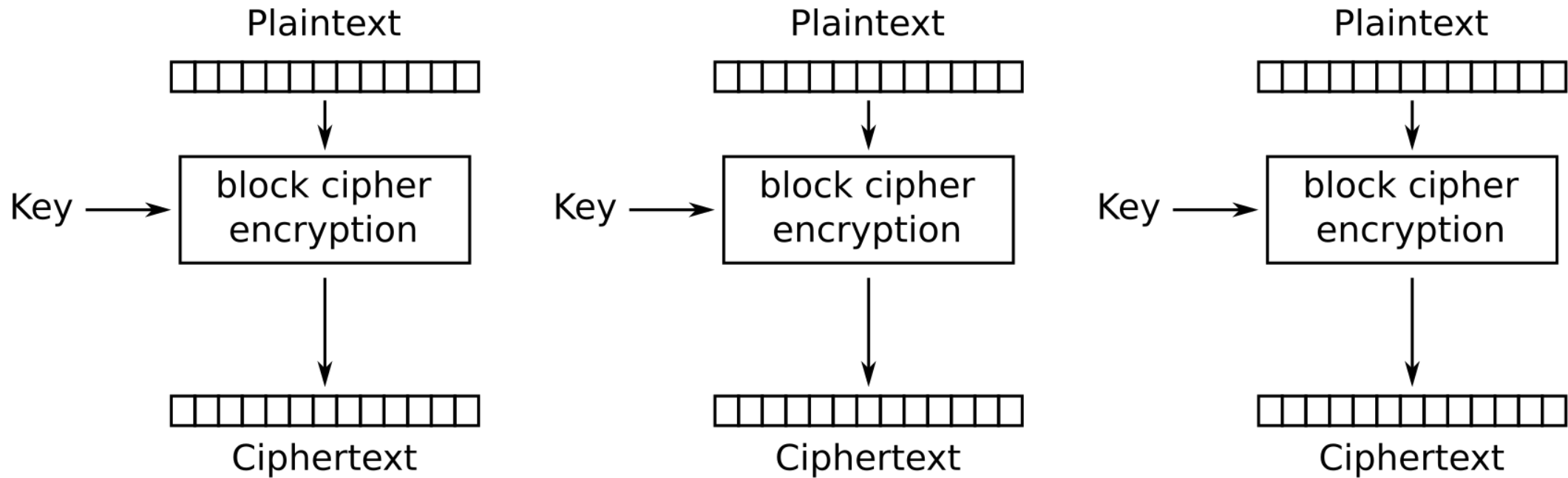
LONGITUD DE LA CLAVE (bits)	RONDAS
128	10
192	12
256	14

- Lo que ocurre en cada una de esas rondas, queda a vuestra curiosidad

Crypto en python

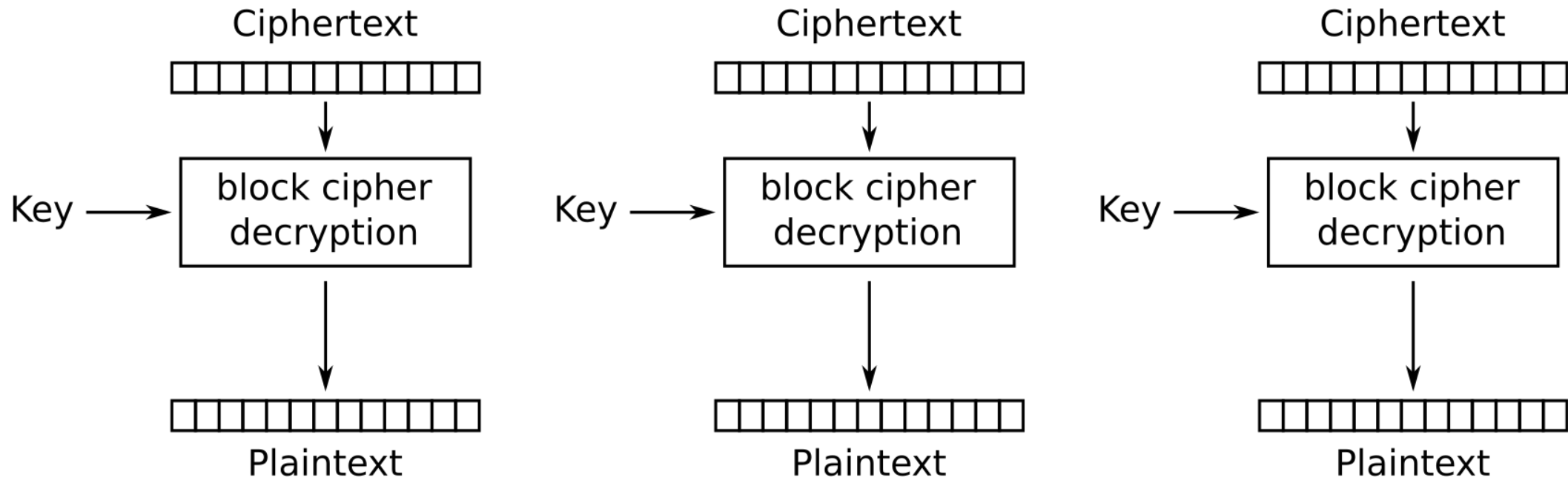
```
1 #pip install pycryptodome
2 from Crypto.Cipher import AES
3 from Crypto.Util.Padding import pad, unpad
4
5
6 BLOCK_SIZE = 32
7 KEY = 'd9a6c7230dfe48ed' #16 bytes
8 text = 'My_first_message'
9 text = pad(text.encode(), BLOCK_SIZE) # Padding para alinear al tamaño del bloque
10 cipher = AES.new(KEY.encode(), AES.MODE_ECB) # nuevo cifrador en modo ECB
11 c = cipher.encrypt(text) # cifrar mensaje
12 print(c)
13 print(unpad(cipher.decrypt(c), BLOCK_SIZE).decode()) # descifrar y eliminar el padding del mensaje
14
15 # OUTPUT
16 '''
17 b'\xbd\x7f\xdes9T;\xff{\x89n\xc1\xdf#\xa0x1\xda\xc8h\xad\xf9e\x8aW\x08\n\xcf\xe3\xa5\xd1\x1a'
18 My_first_message
19 '''
```

¿Qué es el modo ECB?



Electronic Codebook (ECB) mode encryption

¿Qué es el modo ECB?



Electronic Codebook (ECB) mode decryption

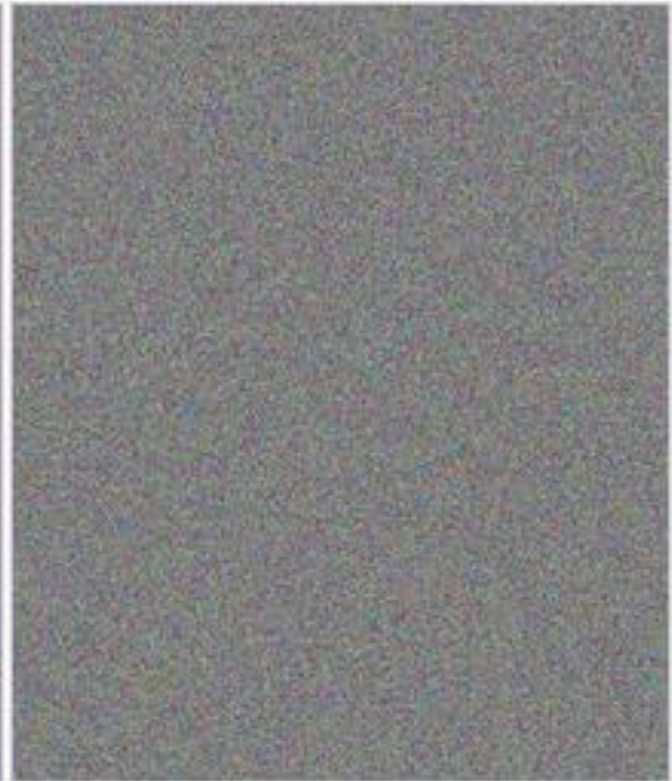
¿Qué es el modo ECB?



Original image

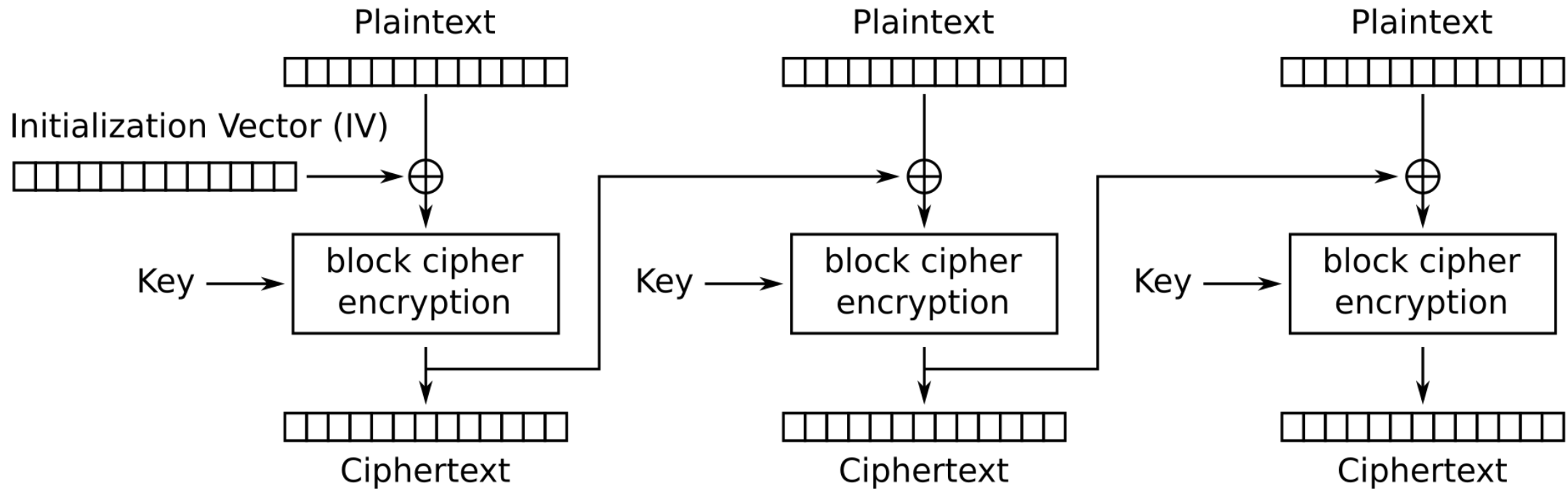


Using ECB allows patterns to be easily discerned



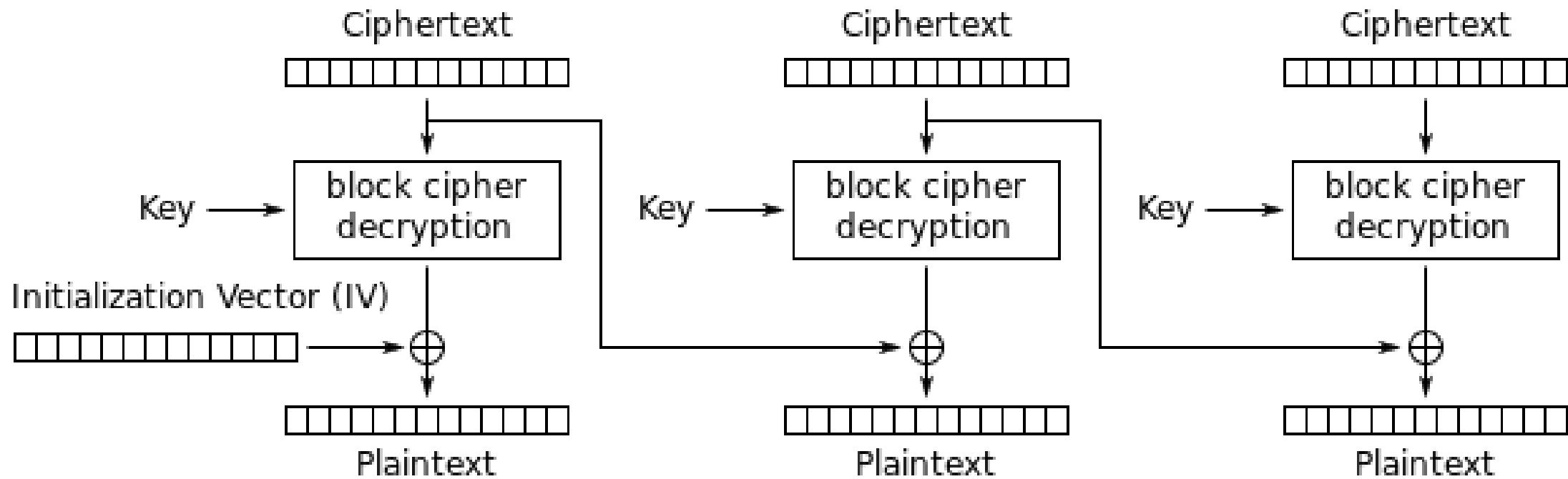
Modes other than ECB result in pseudo-randomness

¿Qué es el modo CBC?

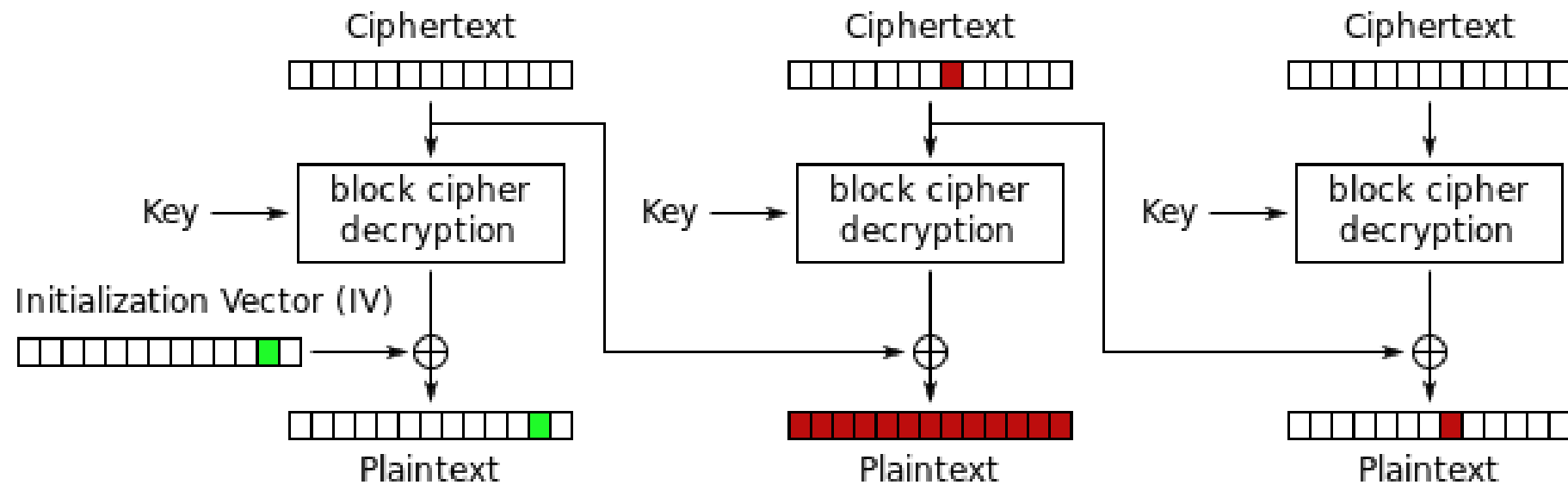


Cipher Block Chaining (CBC) mode encryption

¿Qué es el modo CBC?



Bit flipping



Bit flipping

$$C'_{i-1} = C_{i-1} \oplus x$$

$$P'_i = D_K(C_i) \oplus C'_{i-1}$$

$$P'_i = D_K(C_i) \oplus C_{i-1} \oplus x$$

$$P'_i = P_i \oplus x$$

$$\text{Let } x = P_i \oplus y$$

$$P'_i = P_i \oplus P_i \oplus y$$

$$P'_i = y$$

Existen muchos modos más...

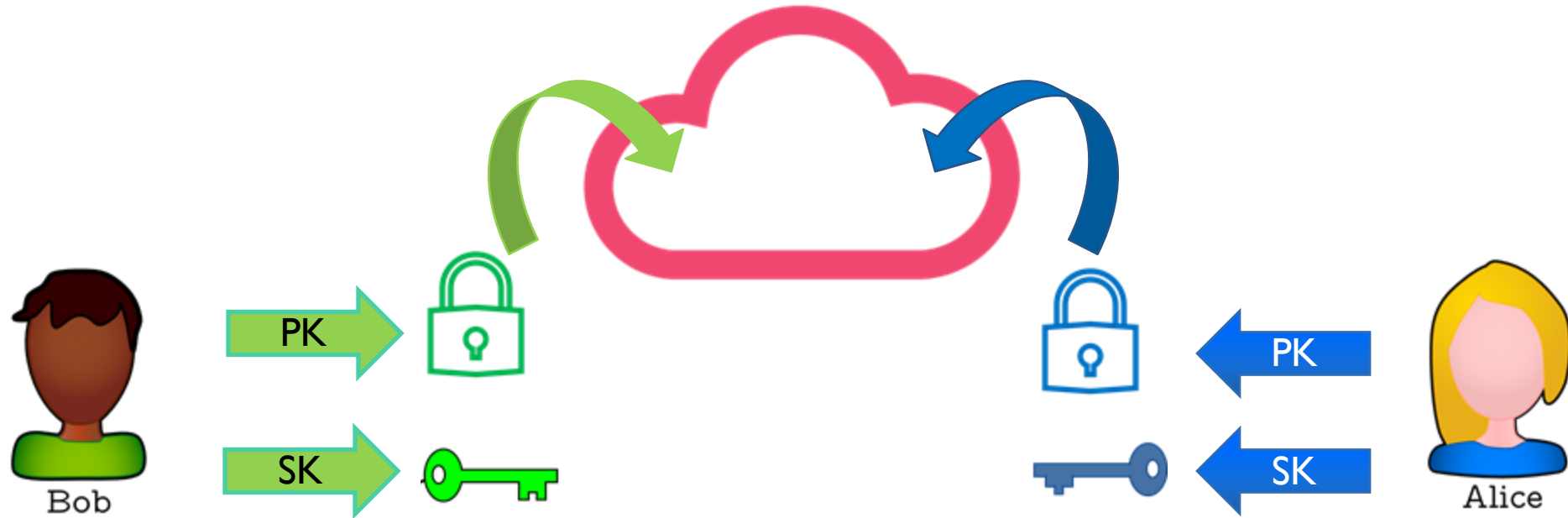
- PCBC
- OFB
- CTR
- CFB
- ...

CRIPTOGRAFÍA ASIMÉTRICA

¿Qué es la criptografía asimétrica?

Es un tipo de cifrado que utiliza una clave pública para cifrar y otra privada para descifrar.

RSA o el Gamal

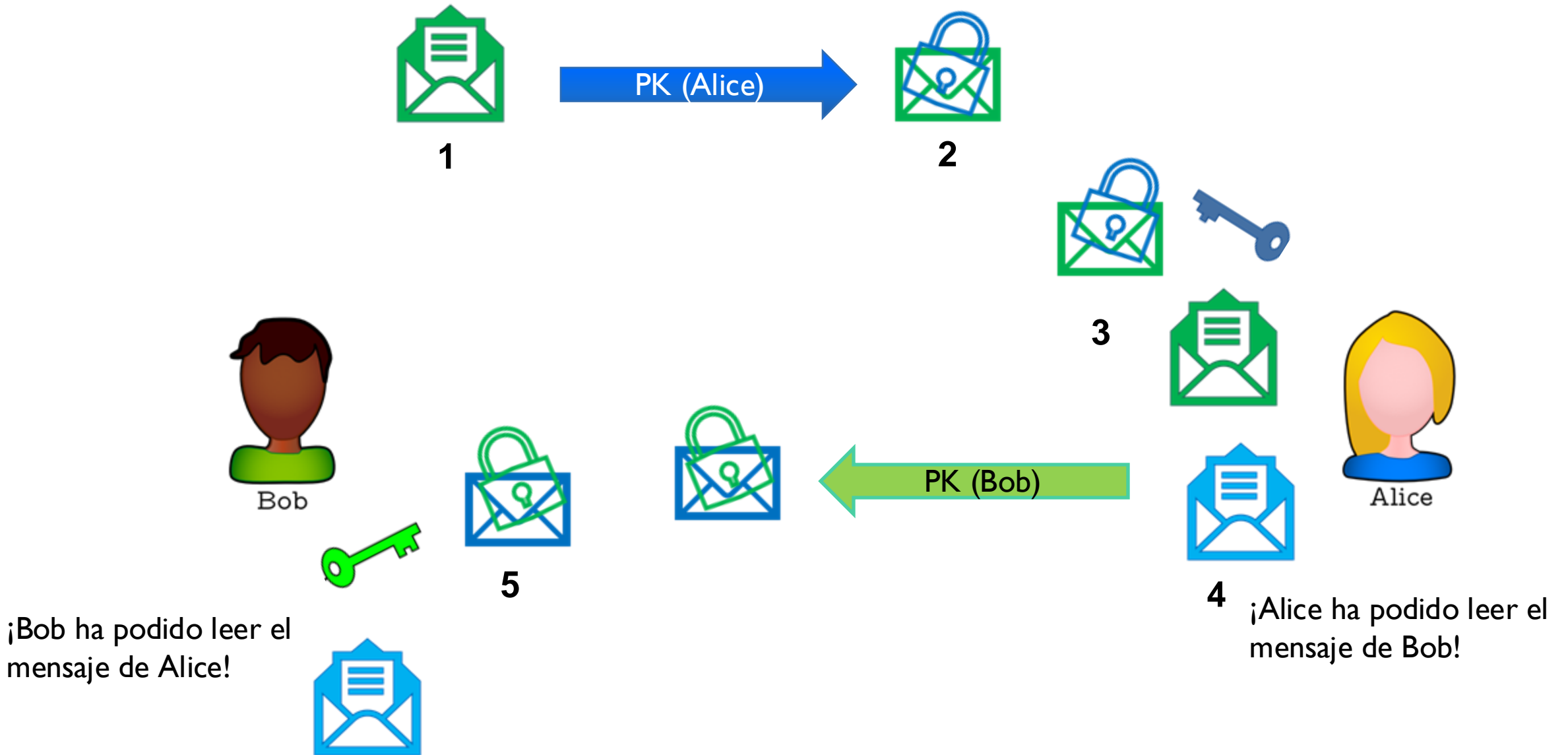


CRIPTOGRAFÍA ASIMÉTRICA

**Bob quiere mandar un mensaje seguro a Alice
pero no han acordado ninguna clave secreta
previamente**

¿Cómo lo hacen?

CRIPTOGRAFÍA ASIMÉTRICA



BASES DE LA ARITMÉTICA MODULAR

Modular/Clock Arithmetic



Modulus 12

$$a \equiv b \pmod{n}$$

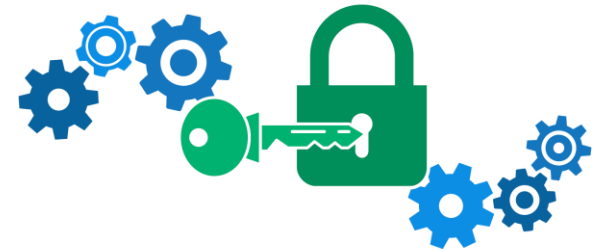
$$63 \equiv 83 \pmod{10}$$

$$64 \bmod 5 = 4$$

$$\begin{array}{r} 64 \\ 14 \\ 4 \end{array} \quad \begin{array}{l} \underline{5} \\ 12 \end{array}$$

Inverso: $a^{-1} * a = 1 \pmod{N}$

¡¡SOLO TIENE INVERSO SI $\text{GCD}(a,N) = 1$!!



RSA – Encriptar y Desencriptar

ENCRIPtar UN MENSAJE

$$C = m^e \bmod(N)$$

e = Exponente

N = Producto de 2
primos P y Q

DESENCRIPtar UN MENSAJE

$$m = c^d \bmod(N)$$

d = Clave privada ($e^{-1} \bmod \phi(N)$)

PARÁMETROS RSA

Clave Pública
(N,e)

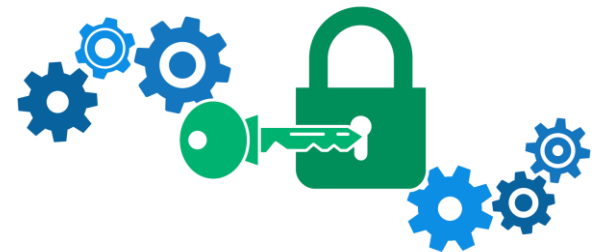
$$N = p * q$$

Si calculamos Phi
de N podemos
sacar la clave
privada

$$\text{Phi} = (p-1)*(q-1)$$

Clave Privada →

$$d = \text{inverso}(e) \text{ mod}(\text{phi})$$



¿Por qué el inverso?

$$e^*d \bmod \phi(N) = 1$$

$$m = (m^e)^d = m^{e^*d} = m^1$$

Python Useful Functions

Comando para instalar módulos extra de python --> **pip install pycryptodome gmpy2**

Después de ejecutar el comando `python3`, podemos usar estas funciones para hacer operaciones con números grandes.

```
pow(base,exponente,modulo) -->  $x^e \text{ mod } N$   
pow(base,-1,modulo) --> calcular el inverso de "base" modulo  
gmpy2.iroot(x,i) --> raiz i de x  
long_to_bytes(mensaje) --> transforma un numero grande en bytes  
bytes_to_long(mensaje) --> transforma un mensaje a un numero entero grande
```

RSA – Encriptar y Desencriptar


```
from Crypto.Util.number import bytes_to_long, long_to_bytes

p = getPrime(512) #Asigna un numero primo de 512 bits
q = getPrime(512)
N = p*q          #Se calcula el modulo
e = 3

#ENCRIPtar UN MENSAJE
mensaje = b"rsa es facil" #mensaje en bytes
mensaje_long = bytes_to_long(mensaje) # Se transforma el mensaje a un numero entero
mensaje_encrypted = pow(mensaje_long, e, N)
print(mensaje_encrypted)

#DESENCRIPtar UN MENSAJE
phi = (p-1)*(q-1) #Se calcula el totient
d = pow(e, -1, phi) #Se calcula el inverso de e modulo phi
mensaje_desencrypted = pow(mensaje_encrypted, d, N) #Calculas el valor original del mensaje
print(long_to_bytes(mensaje_desencrypted)) #Pasas el resultado a bytes para poder ser interpretado
```

RSA – Encriptar y Desencriptar



RSA CIPHER

Cryptography > Modern Cryptography > RSA Cipher

RSA DECODER

Indicate known numbers, leave remaining cells empty.

- ★ VALUE OF THE CIPHER MESSAGE (INTEGER) C=
37629675427502492008492393023411334963114383741303 ...
- ★ PUBLIC KEY E (USUALLY E=65537) E=
65537
- ★ PUBLIC KEY VALUE (INTEGER) N=
88256459553622414063962598765941602942623923080461 ...
- ★ PRIVATE KEY VALUE (INTEGER) D=
- ★ FACTOR 1 (PRIME NUMBER) P=
- ★ FACTOR 2 (PRIME NUMBER) Q=
- ★ INTERMEDIATE VALUE PHI (INTEGER) ϕ =
- ★ DISPLAY PLAINTEXT AS CHARACTER STRING
 COMPUTED VALUES (C,D,E,N,P,Q,...)
 PLAINTEXT AS INTEGER NUMBER
 PLAINTEXT AS HEXADECIMAL FORMAT

▶ CALCULATE/DECRYPT

Results

- ⚠️ ❌ Wiener's attack: failure
- ❌ (Self-Limited) Prime Factors Decomposition: failure
- ✅ P,Q computed with N (FactorDB database)
- ✅ D computed with P,Q,E
- ✅ Decryption using C,D,N

Hola mundo

RSA Cipher - dCode

Tag(s) : Modern Cryptography, Arithmetics

Share

dCode and more

dCode is free and its tools are a valuable help in games,

CHALLENGE

Tenéis 15 minutos para resolver los retos

RSA – Ataque Exponente pequeño y Modulo Grande

$e \rightarrow 3$

$N \rightarrow$

17920302547451456255260628601727787230172785014692541565373018303626923200144183
95774329698747061027147739143566410586761477690456773509856029277075469025914383
28530840229118555963282468638588243456874311075732609267271548495053483974016854
08690680392631146761855793479884844297557829348192912981737152331029681972235731
933154889782680523867681705997376633312277

$m \rightarrow$



123920310321213321233213231212672136867326723673126732

$m^e \rightarrow$



190295043635636787334664801249623051718205871126260174796304068080172301486
402873771262184351748127806309044976187422625748234599636095254735855176834
3285695168

Como m^e es mas pequeño que N podemos calcular el mensaje haciendo la raíz e de m

CHALLENGE

Tenéis 10 minutos para resolver el reto

RSA – Un único primo

e \Rightarrow 65537

N \Rightarrow 857504083339712752489993810777  Es un número primo

m \Rightarrow 123920310321213321233213231212672136867326723673126732

Como N es un número primo podemos calcular Phi de N y con ello la clave privada.

$$\Phi(N) = N - 1 \text{ SOLO SI N ES PRIMO}$$

CHALLENGE

Tenéis 10 minutos para resolver el reto

RSA – Encriptar y Desencriptar

Recursos Útiles

- <http://factordb.com/>
- <https://aurea.es/demos/criptografia/pag/calculadoraRSA.html>
- <https://github.com/jvdsn/crypto-attacks>
- <https://asecuritysite.com/rsa/>
- [RSA Calculator \(tausquared.net\)](https://tausquared.net/)
- [RsaCtfTool: https://github.com/Ganapati/RsaCtfTool](https://github.com/Ganapati/RsaCtfTool)

Recursos de consulta y práctica

- CryptoHack. Retos de criptografía:

<https://cryptohack.org/challenges/>

- CrypTool. RSA paso a paso:

<https://www.cryptool.org/en/cto/rsa-step-by-step.html>

- Vídeo AES:

<https://www.youtube.com/watch?v=tzjIRoqRnv0>



I. Criptografía avanzada

Alumnos Ciberseguridad