UAEM | Universidad Autónoma del Estado de México

Centro Universitario UAEM Zumpango
Ingeniería en Computación

CENTRO UNIVERSITARIO UAEM ZUMPANGO

SEGURIDAD EN REDES

UNIDAD III
Criptografía y autenticación

Tema: Modos de operación

OBJETIVO:

Conocer e implementar los diferentes modos de operación para cifradores por bloques.

MODOS DE OPERACIÓN (1/27)

- Un cifrador por bloques, por sí mismo, es adecuado para cifrar un sólo bloque de longitud fija.

- Un modo de operación es un algoritmo que utiliza un cifrador por bloques para proporcionar **mayor seguridad**.

- Este describe cómo aplicar repetidamente una operación de cifrado para **cifrar todo un mensaje**, es decir, más de un bloque.

- **Operan con bloques completos**, por lo que si el último bloque esta incompleto es necesario que sea rellenado.

MODOS DE OPERACIÓN (2/27)

Vector de inicialización (IV)

- Es una **secuencia** binaria **única** para cada operación de cifrado.
- Se utiliza para generar textos cifrados distintos (aleatorios) cuando se cifra el mismo texto con la misma llave, lo que **evita** tener que **regenerar** las llaves.
- **No** tiene que **repetirse** y en algunos casos es **aleatorio**.
- **No** es necesario mantenerlo en **secreto** pero, en la mayoría de los casos, **no** debe ser utilizado con la **misma llave**.

MODOS DE OPERACIÓN (3/27)

Rellenado (padding)

- Los cifradores por bloques operan con **bloques completos**, por lo que si el último bloque esta incompleto es necesario que sea rellenado antes del proceso de cifrado.

Los principales métodos de rellenado son:

1. CMS. Agrega en los bytes faltantes el **número** de éstos. Está definido en RFC 5652, PKCS#5, PKCS#7 and RFC 1423 PEM.
2. Agregar al primer byte **80h** y los siguientes con **bytes nulos** (null bytes). Definido en ANSI X.923 and ISO/IEC 9797-1.
3. Rellenar con **ceros** los bytes a excepción del último que es llenado con la longitud de los bytes que faltan.

MODOS DE OPERACIÓN (4/27)

Rellenado (padding)

4. Rellenar con **bytes nulos**. Sólo usado en texto ASCII.
5. Rellenar con **espacios**. Sólo usado en texto ASCII
6. Rellenar con **bytes aleatorios** a excepción del último que es llenado con el **número de bytes de relleno**.

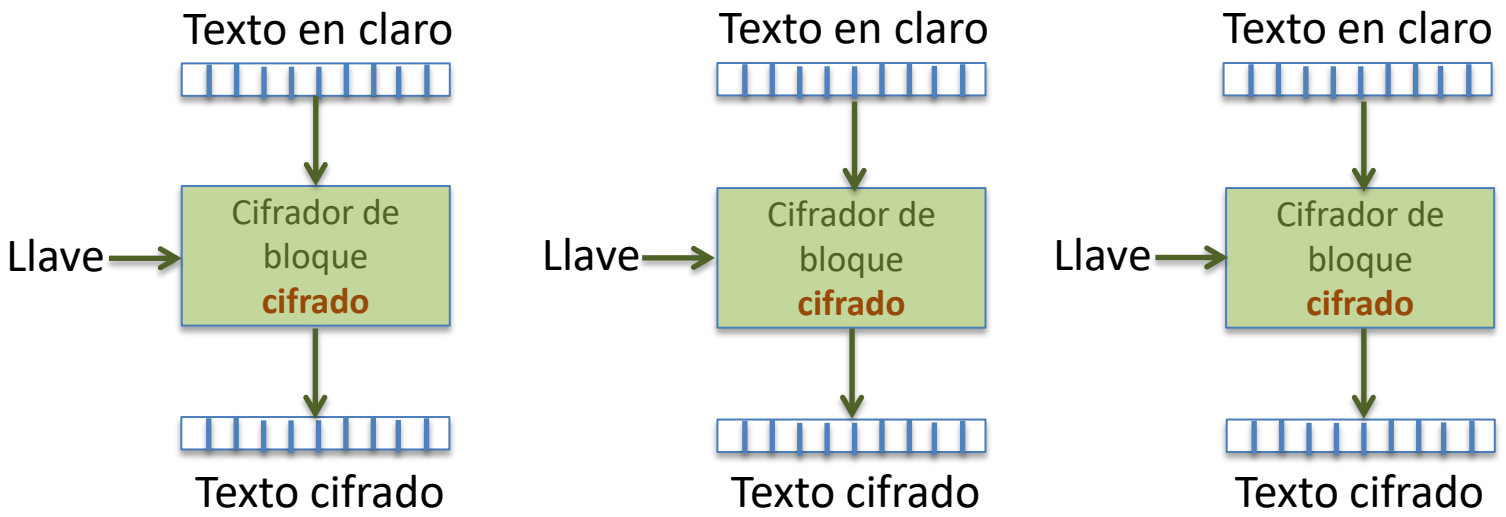
MODOS DE OPERACIÓN (5/27)

Modo ECB (Electronic codebook)

- Es el modo **más simple** de operación.
- Los bloques son cifrados de manera **independiente**.
- Puede soportar **una llave** para cada uno de los bloques.
- Los bloques pueden ser **cifrados en paralelo**.
- Tiene la ventaja de funcionar bien en **canales con ruido**.
- Si existe un error en un texto cifrado, éste sólo afecta el descifrado de ese bloque.
- **Bloques idénticos** de textos en claro producirán **textos cifrados iguales** por lo que no es recomendable para protocolos criptográficos.

MODOS DE OPERACIÓN (6/27)

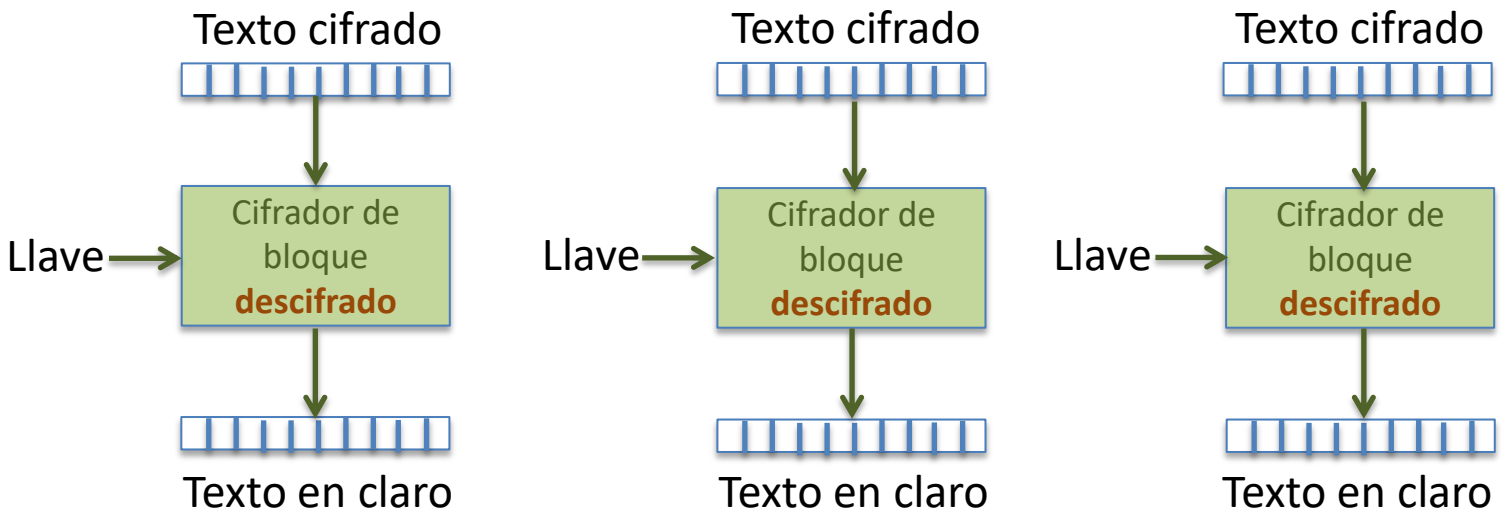
Modo ECB (Electronic codebook)



Cifrado modo ECB

MODOS DE OPERACIÓN (7/27)

Modo ECB (Electronic codebook)



Descifrado modo ECB

MODOS DE OPERACIÓN (8/27)

Modo CBC (Cipher-block chaining)

- Utiliza un **mecanismo de retroalimentación** donde el cifrado de un bloque depende del cifrado del bloque previo de tal modo que se produzca un **encadenamiento**.
- Debido al encadenamiento, el último bloque puede ser utilizar como una **firma digital** o **checksum** del resto, permitiendo verificar que no ha sido alterado.
- **Un error** en el texto cifrado sólo afecta al descifrado de **dos bloques**.
- Se utiliza para **cifrado** y **autenticación**.

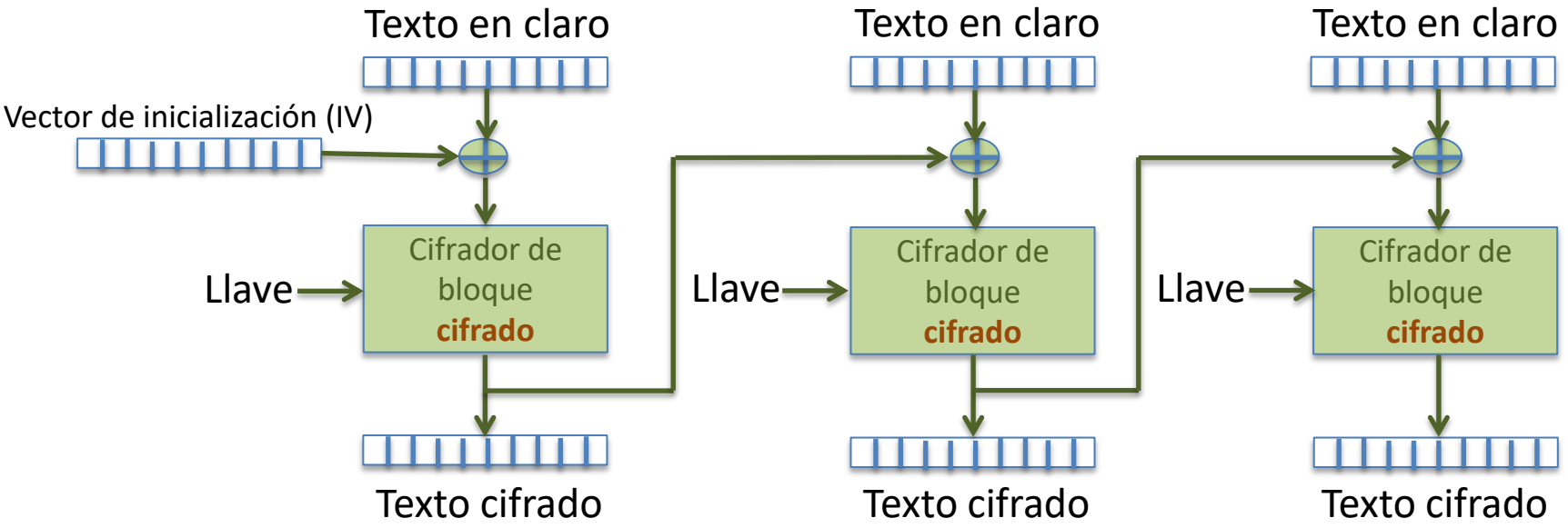
MODOS DE OPERACIÓN (9/27)

Modo CBC (Cipher-block chaining)

- El texto en claro es **XOR** con el bloque previo cifrado y el resultado es **cifrado**.
- Usa un **vector de inicialización**. No es necesario que sea secreto y debe ser diferente para cada texto en claro.
- **No** es posible **descifrar una sola parte**.
- Si los **bloques** no están **ordenados** el descifrado no es correcto.
- El cifrado debe ser realizado de forma secuencial, por lo que **no es paralelizable**.

MODOS DE OPERACIÓN (10/27)

Modo CBC (Cipher-block chaining)

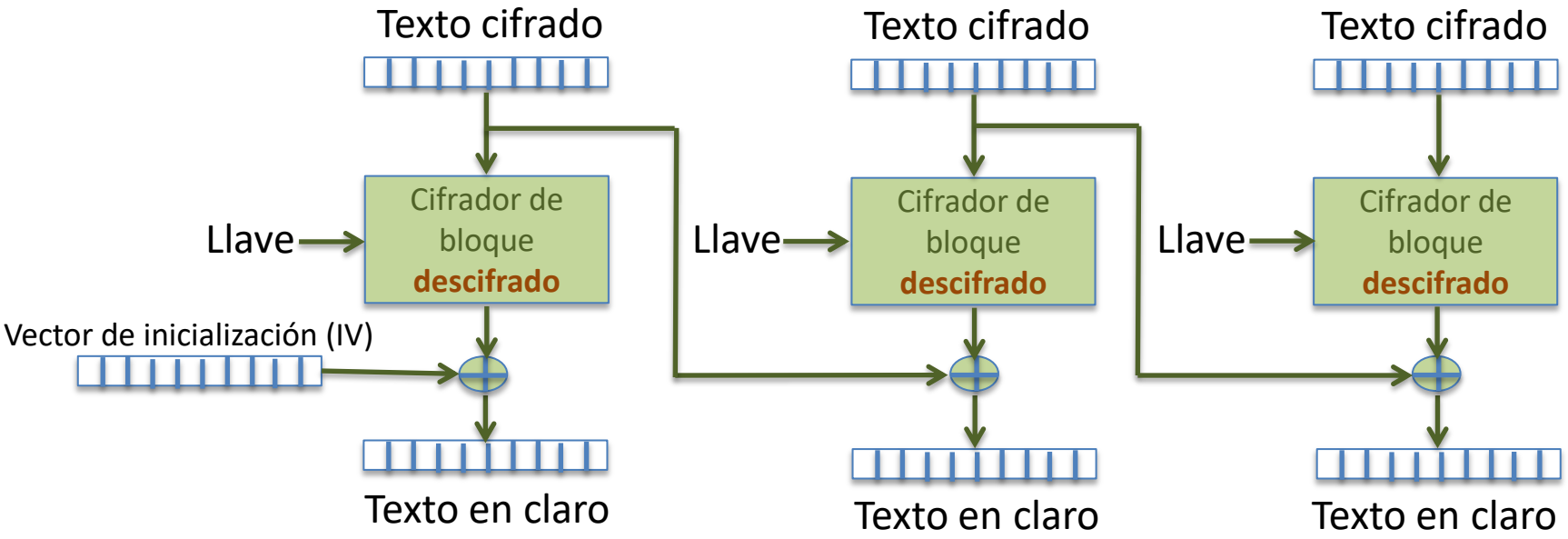


Cifrado modo CBC

$$C_i = E_K\left(P_i \oplus C_{i-1}\right), \quad C_0 = IV$$

MODOS DE OPERACIÓN (11/27)

Modo CBC (Cipher-block chaining)



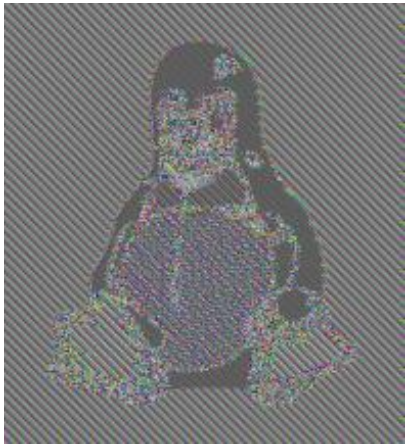
Descifrado modo CBC

$$P_i = D_K(C_i) \oplus C_{i-1}, \quad C_0 = IV$$

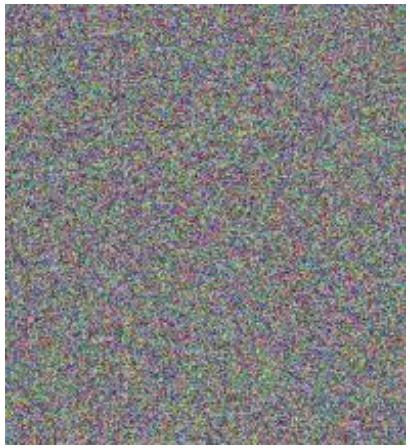
MODOS DE OPERACIÓN (12/27)

Comparación de modo ECB y modo CBC

Información original



Cifrado usando el modo ECB



Cifrado usando el modo CBC

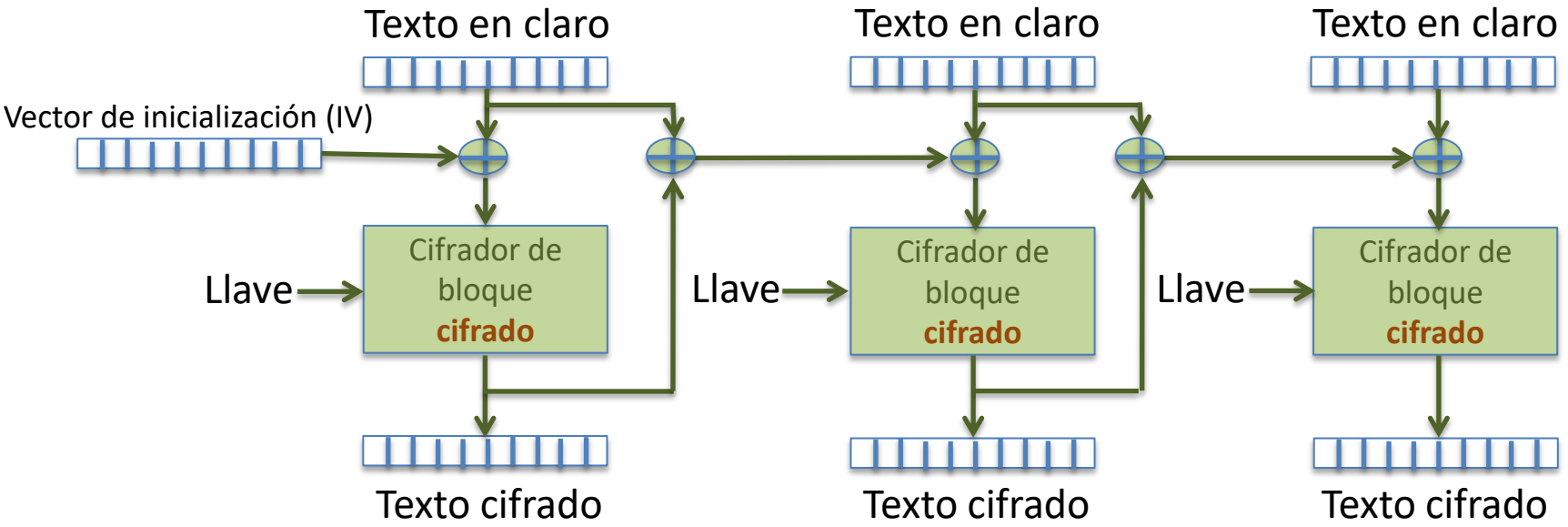
MODOS DE OPERACIÓN (13/27)

Modo PCBC (Propagating cipher-block chaining)

- Este tipo de modo de operación permite que **pequeños cambios** en el texto cifrado **se propaguen más** que en el modo CBC.
- El texto en claro es **XOR** con el **XOR** del texto en claro y el texto cifrado previos para posteriormente ser cifrado.
- Para el cifrado del primer bloque se utiliza un **vector de inicialización**.

MODOS DE OPERACIÓN (14/27)

Modo PCBC (Propagating cipher-block chaining)

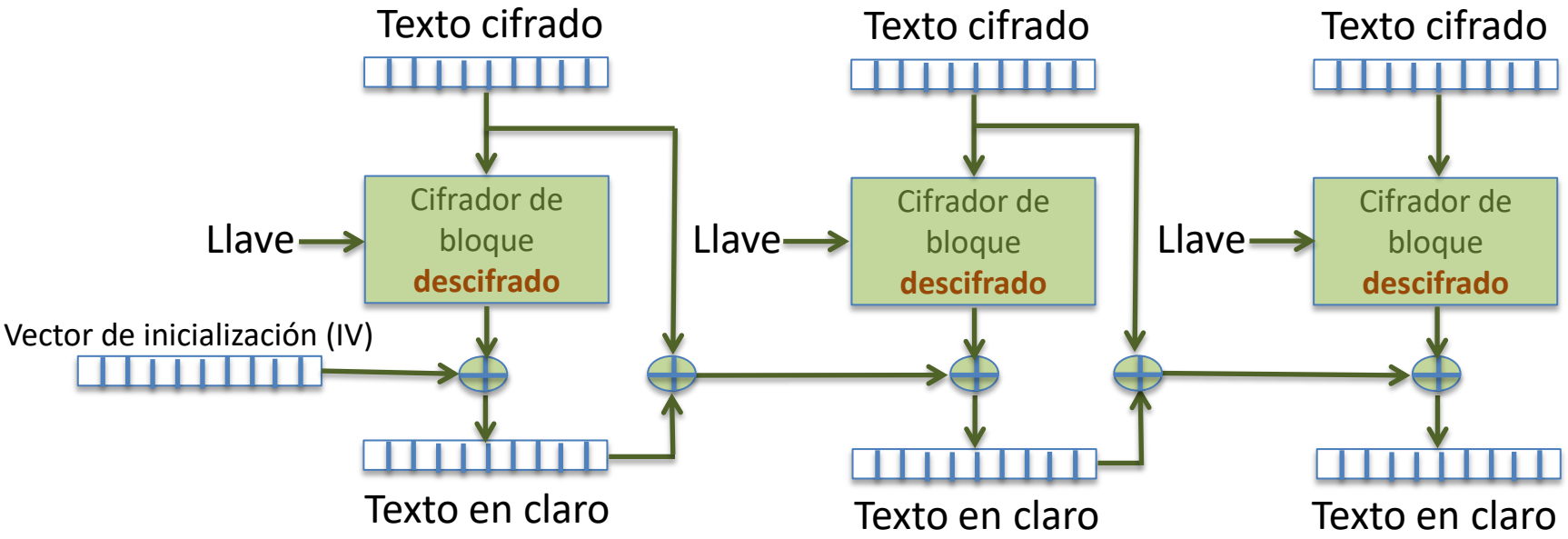


Cifrado modo PCBC

$$C_i = E_K\left(P_i \oplus P_{i-1} \oplus C_{i-1}\right), \quad P_0 \oplus C_0 = IV$$

MODOS DE OPERACIÓN (15/27)

Modo PCBC (Propagating cipher-block chaining)



Descifrado modo PCBC

$$P_i = D_K(C_i) \oplus P_{i-1} \oplus C_{i-1}, \qquad P_0 \oplus C_0 = IV$$

MODOS DE OPERACIÓN (16/27)

Modo CFB (cipher feedback)

- Este tipo de modo de operación produce **secuencias de bits pseudoaleatorios** por medio de un cifrador por bloques.

- El primer bloque cifrado se produce del cifrado del **vector de inicialización XOR** con el **texto en claro**. Los siguientes bloques cifrados se obtienen cifrando nuevamente las salidas anteriores XOR con los textos en claro.

- En el descifrado no es necesario utilizar la función de descifrado, es decir, únicamente se **utilizan cifradores para ambas operaciones**.

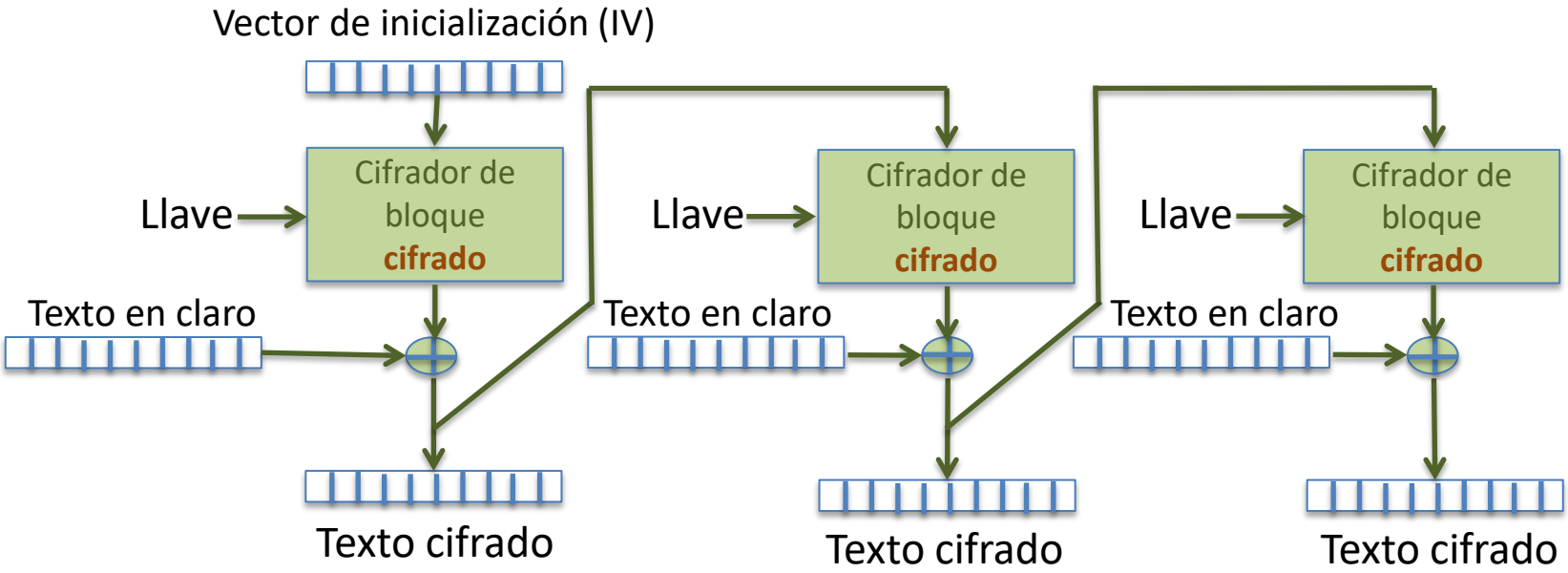
MODOS DE OPERACIÓN (17/27)

Modo CFB (cipher feedback)

- **No utilizar la función de descifrado puede ser una ventaja** en los casos donde dicha función es más lenta que el cifrado.
- En la práctica CFB es útil porque **puede recuperarse de errores** en la transmisión del texto cifrado.
- El descifrado, a diferencia del cifrado, puede ser **paralelizable**.

MODOS DE OPERACIÓN (18/27)

Modo CFB (cipher feedback)



Cifrado modo CFB

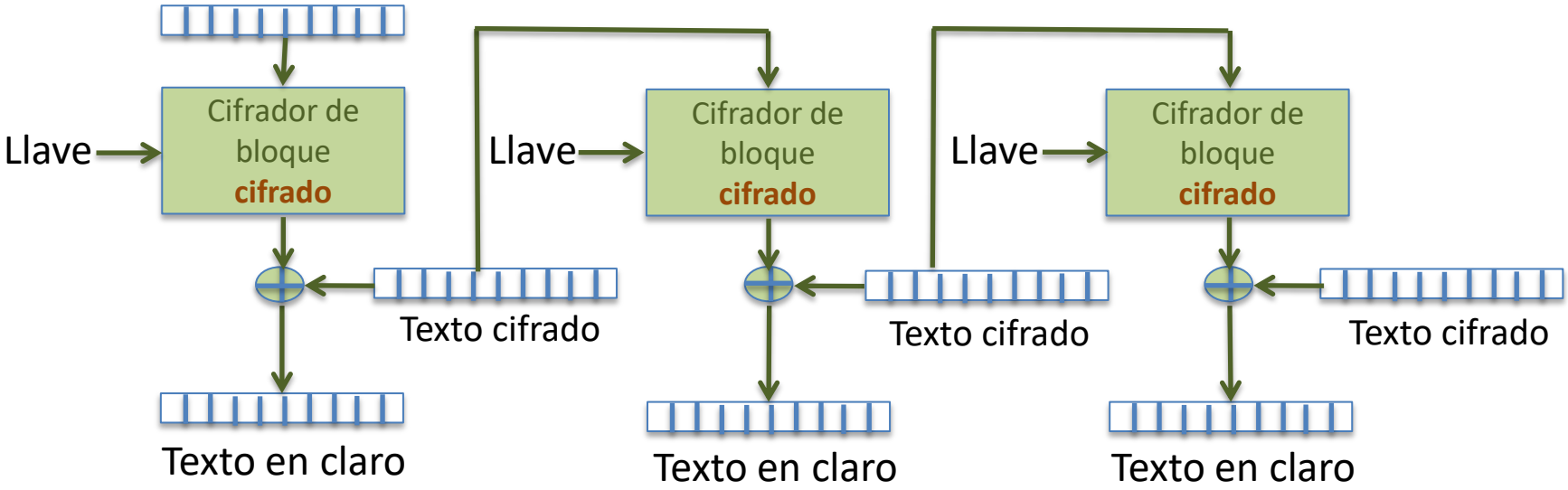
$$C_i = E_K\left(C_{i-1}\right) \oplus P_i$$

$$C_0 = IV$$

MODOS DE OPERACIÓN (19/27)

Modo CFB (cipher feedback)

Vector de inicialización (IV)



Descifrado modo CFB

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

$$C_0 = IV$$

MODOS DE OPERACIÓN (20/27)

Modo OFB (output feedback)

- La estructura del modo **OFB es similar** al modo **CFB** y a un **cifrador por flujo**.

- Utiliza un vector de inicialización para crear un **bloque pseudoaleatorio** que se usa para realizar una operación **XOR** con el **texto en claro** para generar el texto cifrado.

- La salida de la función de cifrado (bloque pseudoaleatorio) es la entrada de la siguiente función, por lo que **sólo depende del vector de inicialización y la llave**.

MODOS DE OPERACIÓN (21/27)

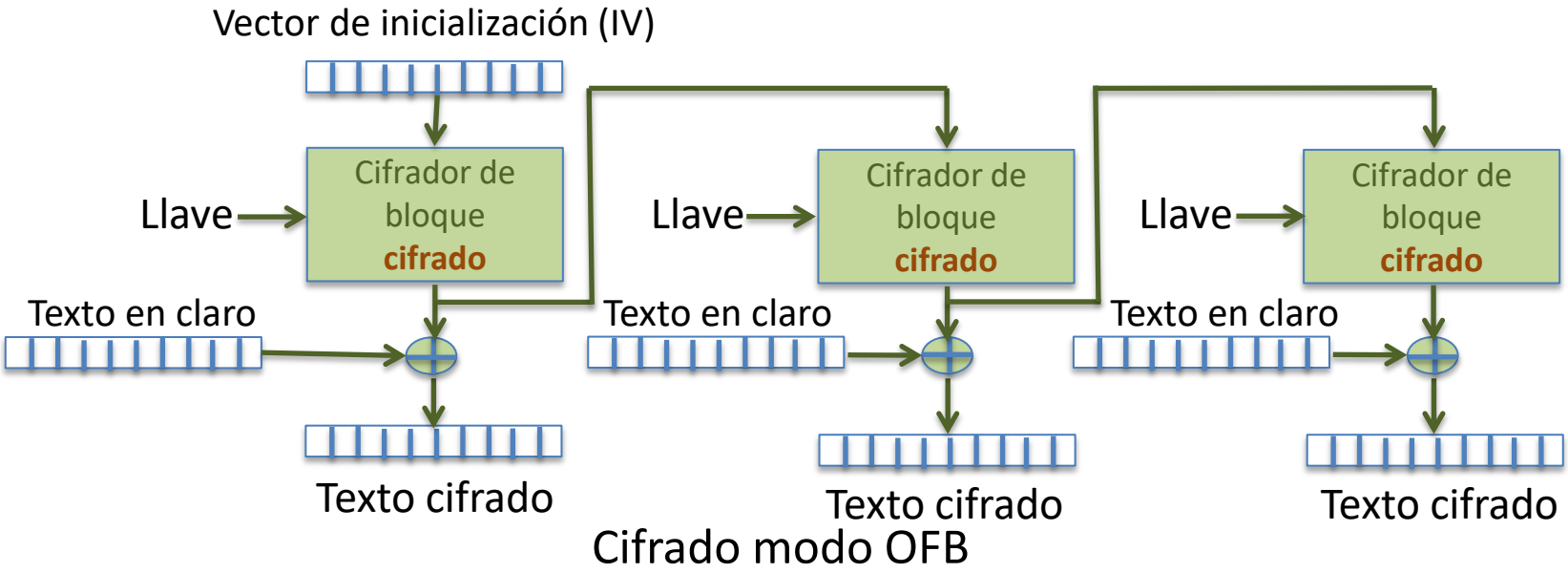
Modo OFB (output feedback)

- Una ventaja de este método es que **los errores de bit en la transmisión no se propagan**.

- Los bloques de salida depende de los bloques anteriores por lo que este modo **no es paralelizable**.

MODOS DE OPERACIÓN (22/27)

Modo OFB (output feedback)



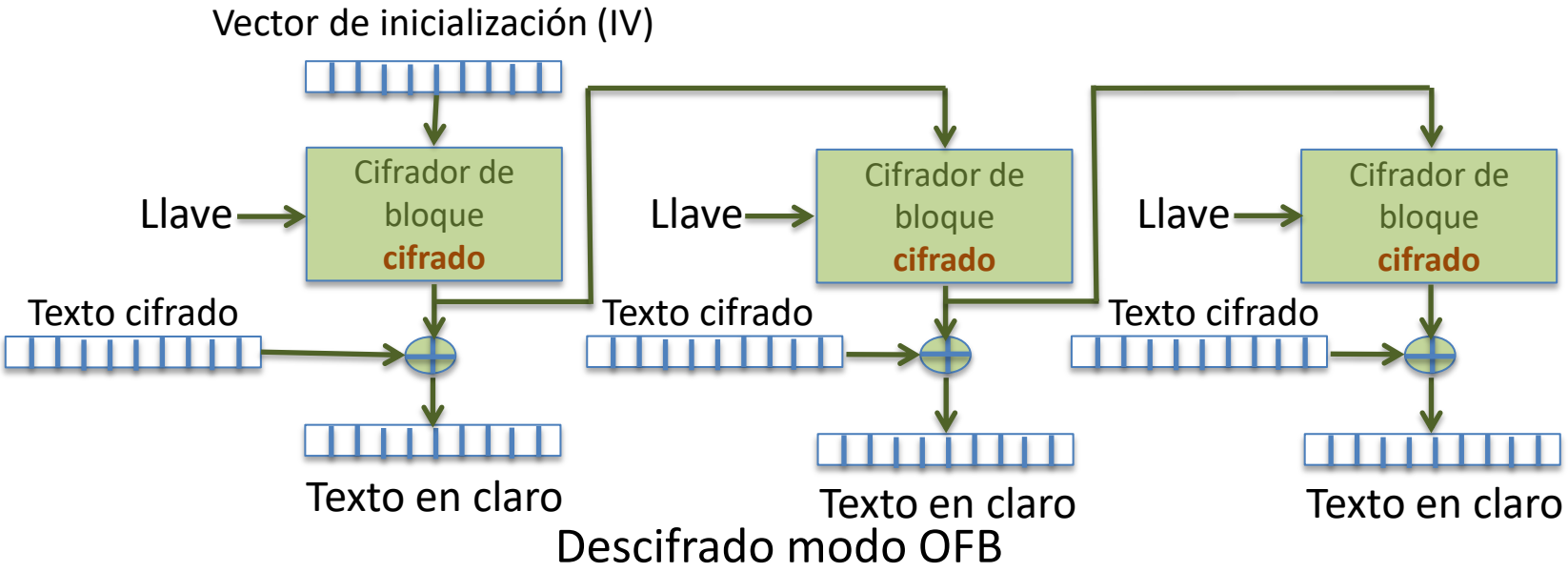
Cifrado modo OFB

$$C_i = P_i \oplus O_i \qquad O_i = E_K\left(I_i\right)$$

$$I_i = O_{i-1} \qquad I_0 = IV$$

MODOS DE OPERACIÓN (23/27)

Modo OFB (output feedback)



Descifrado modo OFB

$$P_i = C_i \oplus O_i \qquad O_i = E_K(I_i)$$

$$I_i = O_{i-1} \qquad I_0 = IV$$

MODOS DE OPERACIÓN (24/27)

Modo CTR (counter)

- El modo de operación CTR simula un **cifrador por flujo de datos**.
- Este modo de operación hace uso de un contador como **vector de inicialización**.
- El vector de inicialización está formado por un ***nonce*** (*number used once*) y un **contador de bloque** que se incrementa en función del bloque a cifrar.
- El *nonce* se construye con **información conocida** entre el emisor y el receptor. Su uso es para evitar bloques cifrados iguales con mensajes idénticos.

MODOS DE OPERACIÓN (25/27)

Modo CTR (counter)

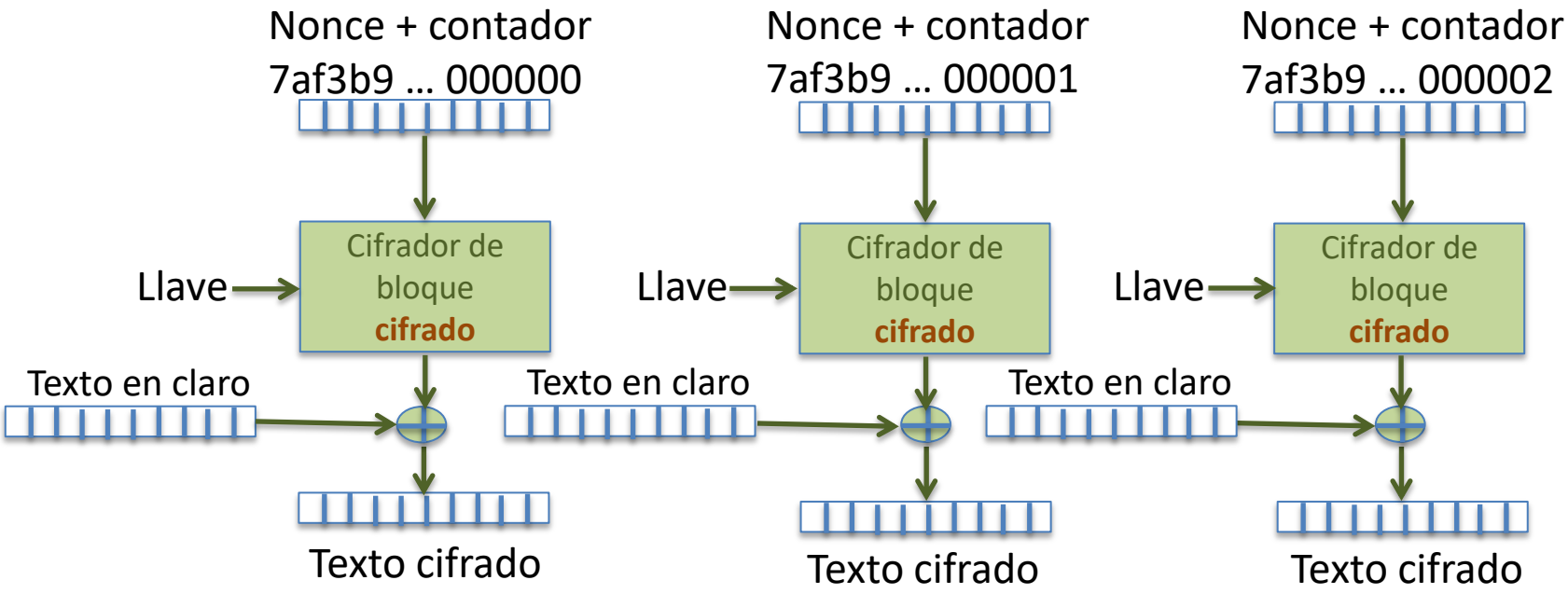
- El primer **bloque cifrado** se obtiene cifrando el *nonce* con el contador de bloques y aplicando una **función XOR** del **bloque cifrado con el bloque de mensaje en claro**.

- El **descifrado** se realiza haciendo la misma operación con la única diferencia que la **operación XOR** es entre el **bloque cifrado obtenido y el bloque cifrado recibido**.

- Los bloques se cifran o descifran de manera independiente por lo que en ambos casos pueden ser **paralelizables**.

MODOS DE OPERACIÓN (26/27)

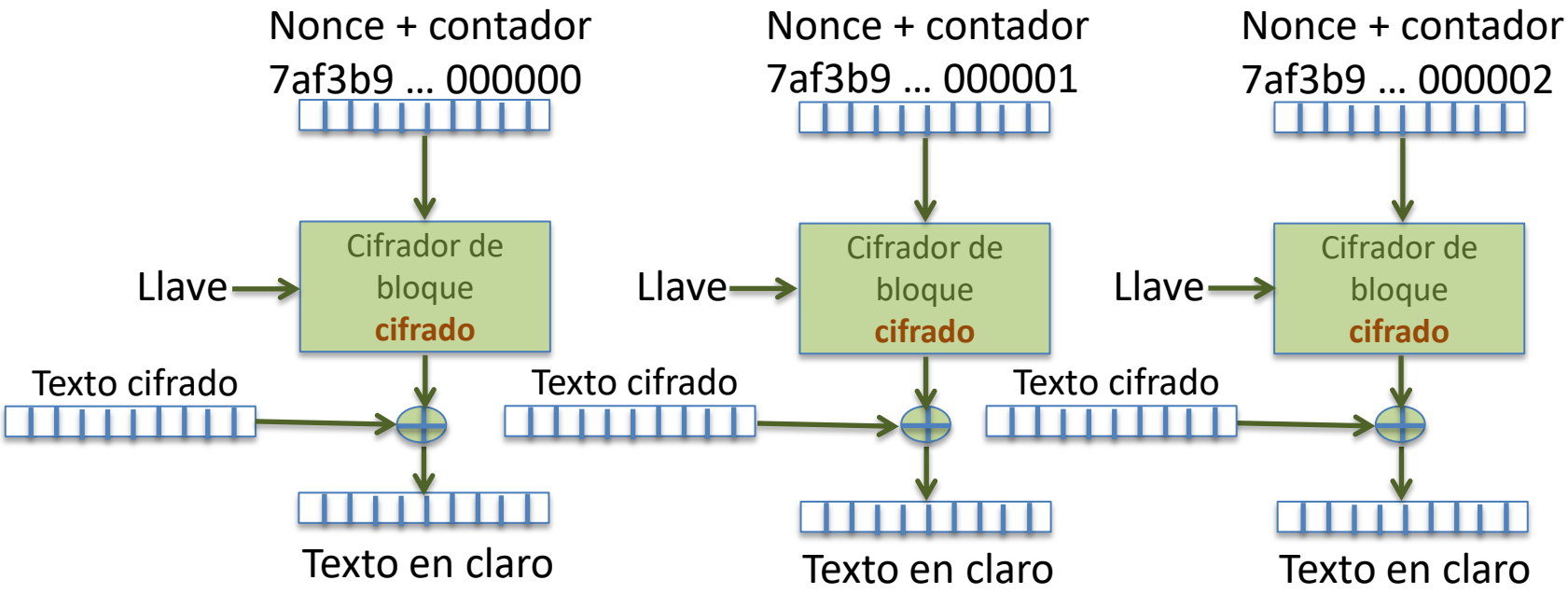
Modo CTR (counter)



Cifrado modo contador (CTR)

MODOS DE OPERACIÓN (27/27)

Modo CTR (counter)



Descifrado modo contador (CTR)

REFERENCIAS (1/2)

1. Carracedo, J. “Seguridad en Redes Telemáticas”. Mc Graw Hill, 2004.
2. McClure, S., Scambray, J. and Kurtz, G. “Hacking Exposed. Network Security Secrets and Solutions” (Third Edition). McGraw-Hill, 2001.
3. Pastor, J. y Sarasa, M.A. “Criptografía digital: fundamentos y aplicaciones.” Zaragoza: Prensas Universitarias, 1998.
4. Rodríguez-Henríquez, F., Saqip, N. A., Díaz-Pérez, A., and Koç, C. K. Cryptographic Algorithms on Reconfigurable Hardware (Signals and Communication Technology), Springer-Verlag New York, Inc. 2006.
5. Stallings, W. “Fundamentos de seguridad en redes: aplicaciones y estándares” (2ª Ed). Pearson-Prentice Hall, 2004.

REFERENCIAS (2/2)

6. Stallings, W. “Network Security Essentials – Applications and Standars” 3a edición.
7. Trappe W., Wshington L. C. Introduction to Cryptography: with Coding Theory. Person Prentice Hall, Second Edition, 2006.
8. William S. Cryptography and Network Security: Principles and Practice. Pearson, sixth edition, 2014.