TEMA
Modos de operación en el cifrado por bloques

Los algoritmos de cifrado por bloque pueden ser ejecutados de diferentes modos. Mostramos ahora los modos más extendidos. Supondremos que el alfabeto de nuestro bloque a cifrar es Σ y que la longitud del bloque es n. Suponemos que el algoritmo de cifrado es E_K, que el de descifrado es D_K, que cada bloque de texto plano lo llamamos m_j, y cada bloque de texto cifrado c_j.

1. MODO DE CIFRA ECB. ELECTRONIC CODEBOOK MODE.

En este modo, el texto plano se descompone en bloques de longitud n. Si es necesario, al texto plano se le añade un suplemento para conseguir que su longitud sea divisible por n. En este modo, cada bloque de longitud n es cifrado de forma independiente al resto de bloques: el texto cifrado es una secuencia de los bloques cifrados. Y el descifrado se realiza aplicando el algoritmo inverso a cada bloque del criptograma, también de forma independiente al resto de criptogramas.

Tenemos, pues, que $\boldsymbol{c_j = E_K[m_j]}$, para $j = 1, 2, \dots, l$. Y para descifrar tenemos que $\boldsymbol{m_j = D_K[c_j]}$.

Este modo se emplea para el **envío de valores sencillos**. Pero es un modo que tiene ciertas debilidades a tener en cuenta:

1. Cuando se usa este modo, a iguales bloques de texto plano se obtienen iguales bloques de texto cifrado. Es así posible reconocer algunos patrones del texto plano en el texto cifrado. Eso facilita un **ataque estadístico**.

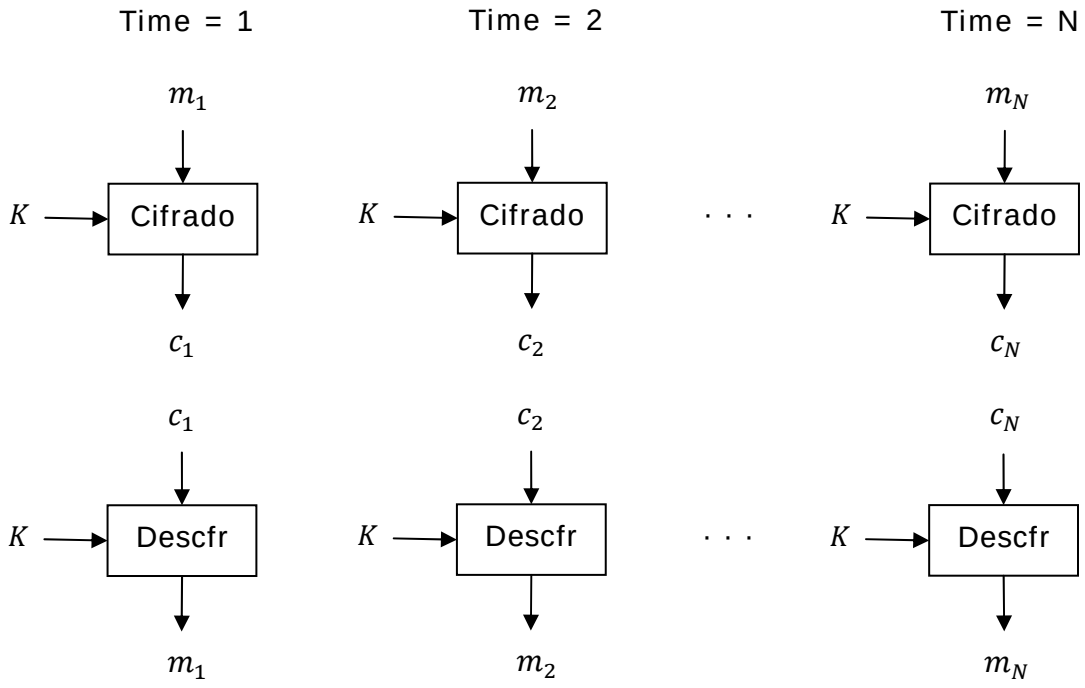

Figura 1: Modo de cifrado ECB

2. Otra vulnerabilidad de este modo de cifra es que un atacante puede **sustituir** algunos bloques del texto cifrado con otros bloques cifrados que hayan sido cifrados con la misma clave. Esta manipulación es difícil de detectar en el receptor. Por eso, ECB no se usa para el cifrado de textos planos largos.

Un modo de incrementar la seguridad de este modo de cifra es que cada bloque de texto a cifrar esté formado por un determinado número de caracteres del texto plano y otros hasta n lo ocupen caracteres aleatorios. Pero eso exige la generación y el uso de muchos caracteres aleatorios y aumenta el número de bloques a cifrar, reduciendo así la eficiencia del procedimiento.

2. MODO DE CIFRA CBC. CIPHERBLOCK CHAINING MODE.

En este modo, la entrada al algoritmo de cifrado es el resultado de la operación XOR entre el actual bloque de texto plano a cifrar y el bloque de texto cifrado precedente. Se emplea la misma clave en cada bloque.

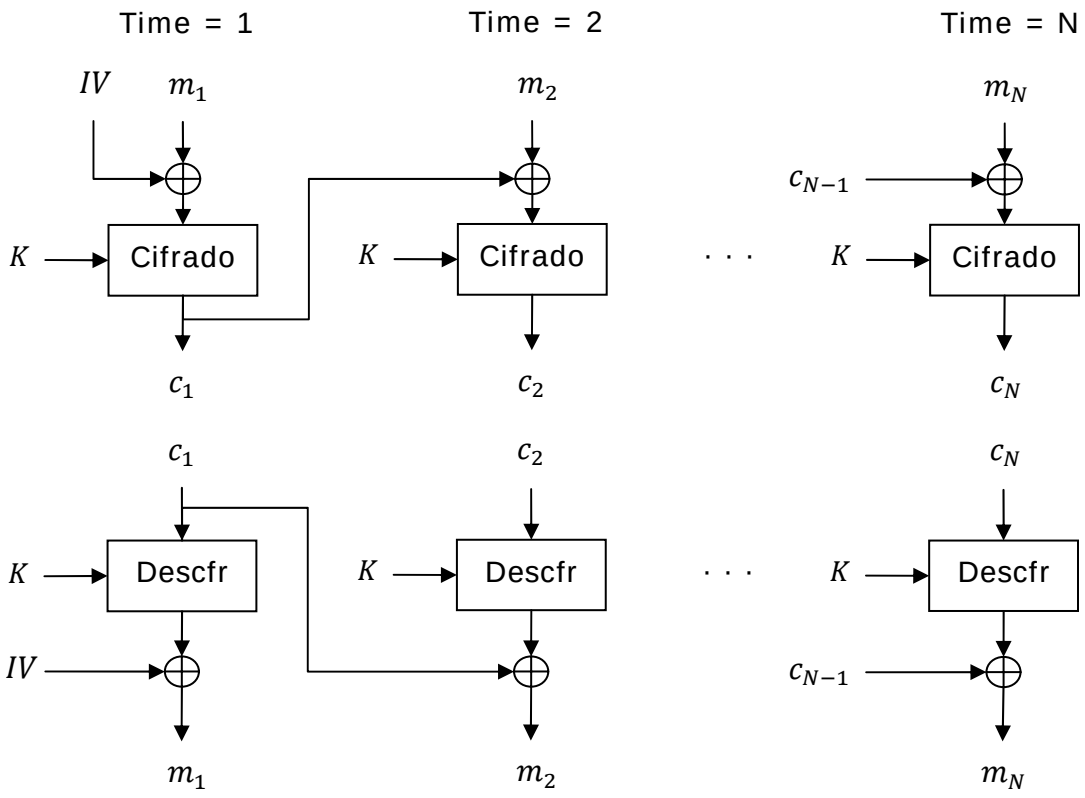

Figura 2: Modo de cifrado CBC

Para descifrar, cada bloque cifrado es procesado por el algoritmo de descifrado, y el resultado es sometido a la operación XOR con el bloque cifrado precedente, para obtener así el bloque de texto plano.

Tenemos entonces que $\boldsymbol{c_j = E_K[c_{j-1} \oplus m_j]}$ para $j = 2, 3, \ldots, l$. Y para descifrar tenemos que $D_K[c_j] = D_K\left[E_K[c_{j-1} \oplus m_j]\right] = c_{j-1} \oplus m_j$, por lo cual, finalmente tenemos que $\boldsymbol{c_{j-1} \oplus D_K[c_j] = c_{j-1} \oplus c_{j-1} \oplus m_j = m_j}$.

El primer bloque ($j = 1$) no tiene bloque cifrado previo. Para la generación del primer bloque de texto cifrado se introduce un **vector de inicialización** (lo llamamos IV) que es el que se va a operar con el operador XOR ("xorear"), con el primer bloque de texto plano. Para el descifrado, el IV será "xoreado" con la salida del algoritmo de descifrado, para obtener así el primer bloque de texto plano: $\boldsymbol{c_0 = IV}$, $\boldsymbol{c_1 = E_K[c_0 \oplus m_1]}$ y $\boldsymbol{m_1 = c_0 \oplus D_K[c_1]}$.

El vector IV debe ser conocido por ambas partes: tanto por el emisor que cifra como por el receptor que descifra. Para obtener máxima seguridad, el vector IV puede protegerse como si de una clave se tratara. Esto puede hacerse enviado el

emisor al receptor el valor de IV cifrado con el modo ECB. Es conveniente actuar así, porque existen ataques que se basan en el conocimiento del vector IV.

El modo CBC evita los problemas del modo ECB. En este modo, el cifrado de cada bloque no sólo depende de la clave, sino también del bloque previo. Es decir, estamos ante un modo de cifrado dependiente del contexto. Así, bloques iguales, en diferentes contextos, quedan cifrados de forma diferente. El receptor puede darse cuenta de que le han cambiado el texto cifrado porque no obtiene nada en la manipulación del descifrado.

En caso de que se produzca un error en la transmisión de un bloque cifrado (por ejemplo, el bloque c_j) entonces pierde sentido el bloque descifrado a partir de este c_j y perdemos m_j. Como m_{j+1} depende también del valor de c_j, entonces también perderemos ese bloque de texto plano. El bloque m_{j+2}, al depender únicamente de c_{j+1} y c_{j+2}, y tener éstos correctamente transmitidos y recibidos, sí se puede obtener correctamente. Y así también con los bloques sucesivos.

Así pues, este modo es apropiado para cifrar mensajes de longitud bastante mayor que n (tamaño del bloque en el sistema de cifrado que estemos usando).

3. MODO DE CIFRA CFB. CIPHER FEEDBACK MODE.

CBC es un modo válido para cifrar mensajes largos. Pero en aplicaciones de tiempo real (por ejemplo, que el receptor quiera descifrar el mensaje a medida que lo vaya recibiendo) se encuentran, en la práctica, problemas de eficiencia. Eso es necesario, por ejemplo, en comunicaciones telefónicas: el emisor cifra el bloque actual y lo envía, y el receptor lo descifra en cuanto lo recibe. Es decir, las funciones de cifrado y descifrado se utilizan secuencialmente y no simultáneamente. Además, cuanto más computacionalmente complejo sea el proceso de cifrado y descifrado, más tiempo pasa entre el cifrado y el descifrado.

En el caso del modo CFB, este procedimiento se hace de forma diferente. Ahora la función de cifrado no se usa directamente para cifrar bloques de texto plano, sino para **generar una secuencia de bloques de clave**. El texto plano se cifra sumándole módulo 2 el bloque de clave (o lo que es lo mismo, realizando la operación XOR entre el bloque de texto plano y el bloque de clave generada). Y para descifrar el bloque se vuelve a “xorear” el bloque cifrado con el

correspondiente bloque de clave. Los bloques de clave pueden ser generados simultáneamente por el emisor y por el receptor; únicamente la operación XOR sí se realiza simultáneamente.

De nuevo, necesitamos un vector de inicialización, $IV \in \{0,1\}^n$. También necesitamos un entero positivo r, tal que $1 \leq r \leq n$ (un valor habitual es $r = 8$: si $n = 64$, como es el caso de DES, entonces cada bloque de texto plano se trocearía en 8 sub-bloques). El texto plano queda descompuesto en bloques de longitud r (supongamos que tenemos u bloques). **Para cifrar los mensajes planos $\boldsymbol{m_1, \ldots, m_u}$** hacemos:

1. $I_1 = IV$.
2. Para $1 \leq j \leq u$ hacer:
    a. $O_j = E_k(I_j)$.
    b. Construimos la cadena t_j, que consiste en los r bits más significativos de O_j.
    c. $c_j = m_j \oplus t_j$.
    d. $I_{j+1} = 2^r I_j + c_j \, mod \, 2^n$: I_{j+1} se genera eliminando los primeros r bits de I_j y sumando c_j.

El texto cifrado queda formado por la secuencia $c_1, c_2, \ldots, c_u$.

Para el descifrado, el procedimiento a seguir es similar. Ahora el receptor hace los siguientes pasos:

1. $I_1 = IV$.
2. Para $1 \leq j \leq u$ hacer:
    a. $O_j = D_k(I_j)$.
    b. Construimos la cadena t_j, que consiste en los r bits más significativos de O_j.
    c. $m_j = c_j \oplus t_j$.
    d. $I_{j+1} = 2^r I_j + c_j \, mod \, 2^n$.

Tanto el emisor como el receptor pueden ponerse al cálculo de la cadena t_{j+1} tan pronto como ambos conocen el sub-bloque de texto cifrado c_j. Así, el bloque de clave t_1 puede calcularse simultáneamente en el emisor en el receptor. El emisor genera el sub-bloque de texto cifrado $c_1 = m_1 \oplus t_1$ y lo envía al receptor. El cálculo de m_1 es rápido ya que no es más que una simple operación XOR. Ahora tanto el emisor como el receptor pueden calcular la cadena de clave t_2, etc.

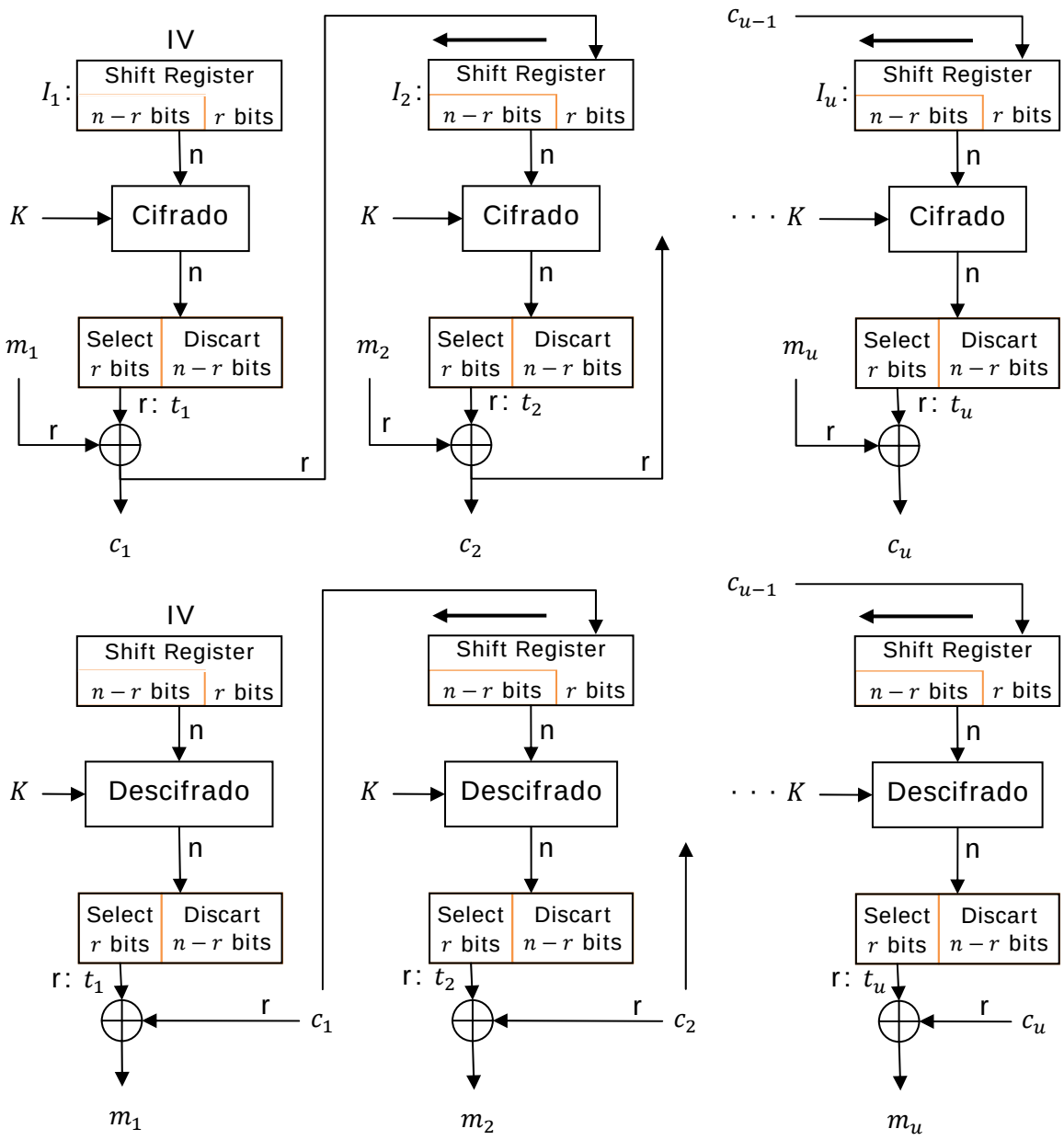

Figura 3: Modo de cifrado CFB

Con este modo logramos que la transmisión sea más rápida, pero por otro lado hay que aplicar con mucha mayor frecuencia el algoritmo de cifrado y de descifrado. **El valor de r es un valor de conveniencia entre las velocidades de computación y de transmisión**.

En este modo de transmisión un error en un sub-bloque estropea la labor de descifrado mientras que ese tramo forma parte del vector I_j.

Veámoslo: Supongamos que $r = 8$ y $n = 64$. Supongamos que el emisor transmite

los subbloques de texto cifrado $c_1,\ c_2,\ldots,c_k,\ldots$, y supongamos que c_1 se corrompe durante la transmisión, de forma que los bloques recibidos son $\tilde{c}_1,\ c_2,\ldots$

Al descifrar, tomaremos $\tilde{c}_1$ y produciremos un subbloque de texto plano erróneo: una versión de m_1 con bits erróneos en las posiciones donde $\tilde{c}_1$ tenga bits erróneos.

Ahora, después de descifrar este sub-bloque primero, el receptor construirá un valor equivocado de I_2 (lo llamaremos $\tilde{I}_2$) Si I_1 era $(*,*,*,*,*,*,*,*)$, entonces $\tilde{I}_2$ será $(*,*,*,*,*,*,*,\tilde{c}_1)$. Cuando el receptor reciba el subgrupo no corrupto c_2 y lo descifre, obtendrá una versión errónea de m_2. Cuando forme I_3, obtendrá $\tilde{I}_3$ de la forma $(*,*,*,*,*,*,\tilde{c}_1,c_2)$. El receptor repetirá este proceso, y seguirá obteniendo versiones erróneas de $m_1, m_2, \ldots, m_9$. Cuando calcule entonces el valor de I_9, el bloque erróneo se habrá trasladado a la primera posición de la cadena: $\tilde{I}_9 = (\tilde{c}_1, c_2, \ldots, c_8)$. Al siguiente paso, el error quedará perdido, y el valor de I_{10} será ya el correcto.

4. MODO DE CIFRA OFB. OUTPUT FEEDBACK MODE.

Es un modo muy parecido al CFB. Al igual que en CFB, el modo OFB usa un bloque cifrado de longitud n, otra longitud de bloque r, con $1 \leq r \leq n$ y un vector de inicialización IV.

Si el emisor cifra su mensaje con la clave e, entonces, al igual que con CFB, deberá descomponer cada bloque en sub-bloques de longitud r

Y entonces, se deben realizar las siguientes operaciones:

1. $I_1 = IV$.
2. Para $1 \leq j \leq u$ hacer:
    a. $O_j = E_k(I_j)$.
    b. Construimos la cadena t_j, que consiste en los r bits más significativos de O_j.
    c. $c_j = m_j \oplus t_j$.
    d. $I_{j+1} = O_j$. Otra posibilidad: $I_{j+1} = 2^r I_j + t_j \ mod\ 2^n$

De nuevo, la tarea de descifrado es similar: simplemente hay que cambiar el paso 2.c por $m_j = c_j \oplus t_j$.

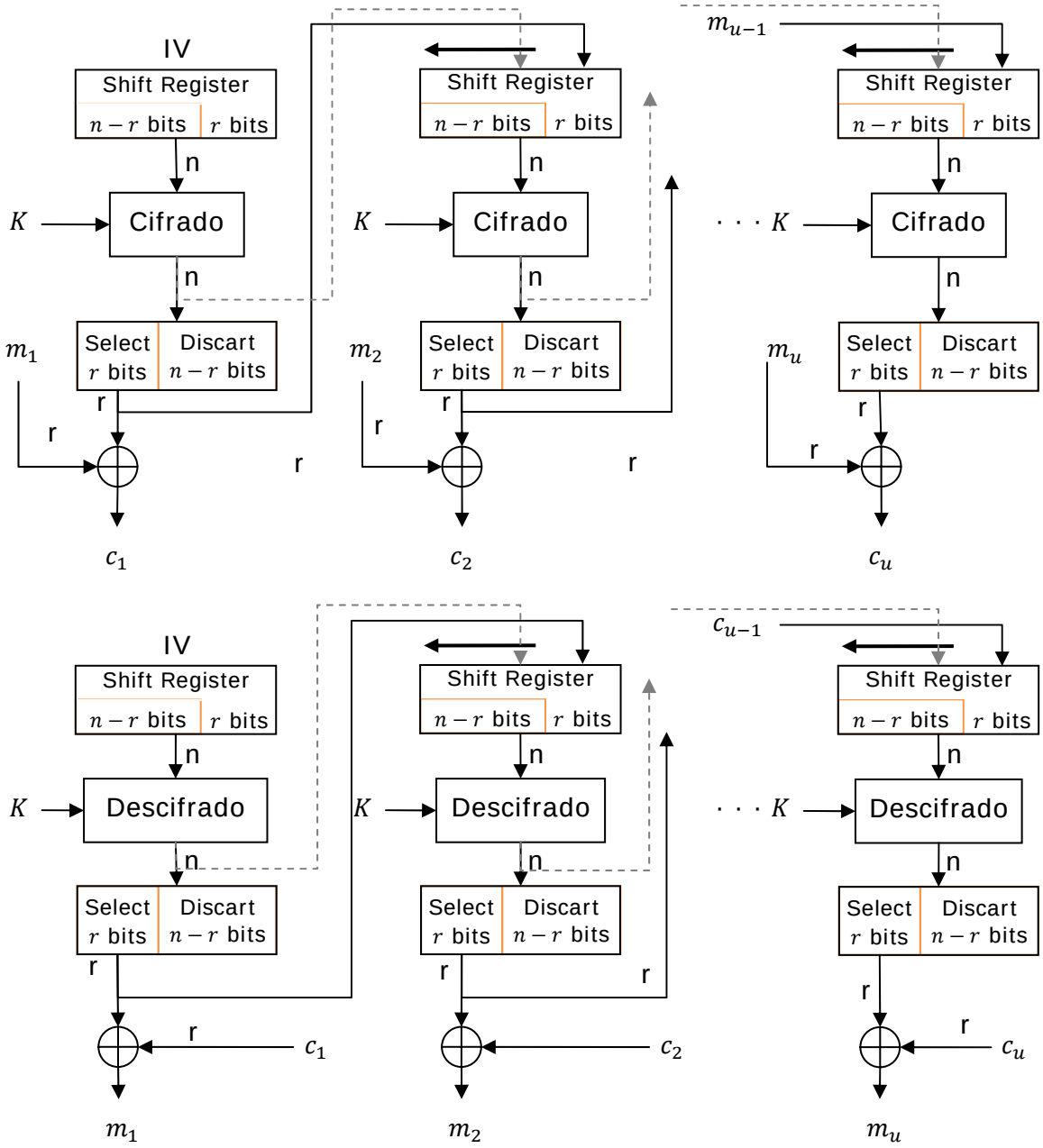

Figura 4: Modo de cifrado OFB

Si un bit del texto cifrado se transmite de forma incorrecta, entonces el texto plano estará erróneo exactamente en la misma posición. Un error en un bit no tiene influencia.

La clave de bloque t_j sólo depende del vector de inicialización IV, y no de la clave k. Dichos bloques pueden ser calculados simultáneamente por el receptor y por el emisor: eso supone una mejora frente al modo CFB. Sin embargo, el cifrado

de un bloque del texto plano en el modo OFB no depende de los bloques de texto plano previos, sino sólo de su posición. Por eso, la manipulación del texto cifrado resulta aquí más sencilla que en el modo CFB.

Una ventaja del modo OFB es que no se transmiten los errores de transmisión en un bit: si ocurre un error en un bit de C_1, entonces ese error sólo afecta al valor de m_1. Las siguientes sub-secuencias o sub-bloques de texto plano no se ven afectadas por ese error. Con CFB, como ya vimos, un error se propaga en los siguientes sub-bloques de texto plano descifrado. La desventaja del modo OFB es que vuelve a ser vulnerable a ataques por modificación de la cadena del mensaje.

REFERENCIAS

[1] "Handbook of Applied Cryptography". A. Menezes, P. van Oorschot, and S. Vanstone. CRC Press, Inc. 1997.

[2] "Cryptography and Network Security. Principles and practices". William Stallings. Prentice Hall. Pearson Education. Third edition. 2003.

[3] "Introduction of Cryptography with coding theory". Wade Trappe and Lawrence C. Washington. Prentice Hall, 2002.

[4] "Introduction to Cryptography". Johannes A. Buchmann. Springer Verlag, 2004. Second Edition.