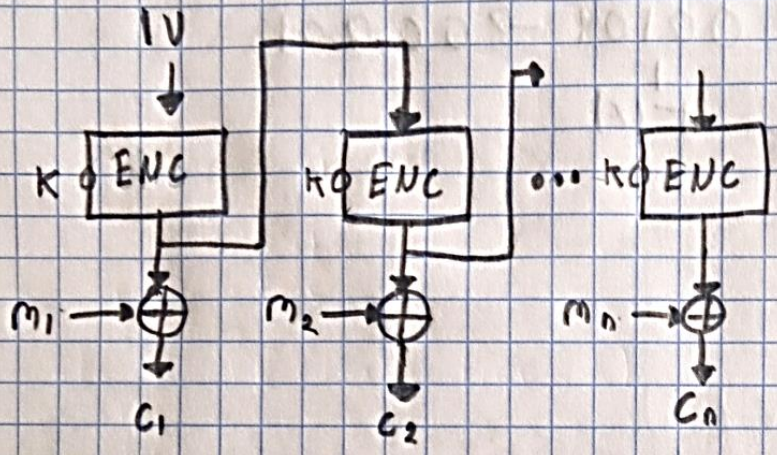
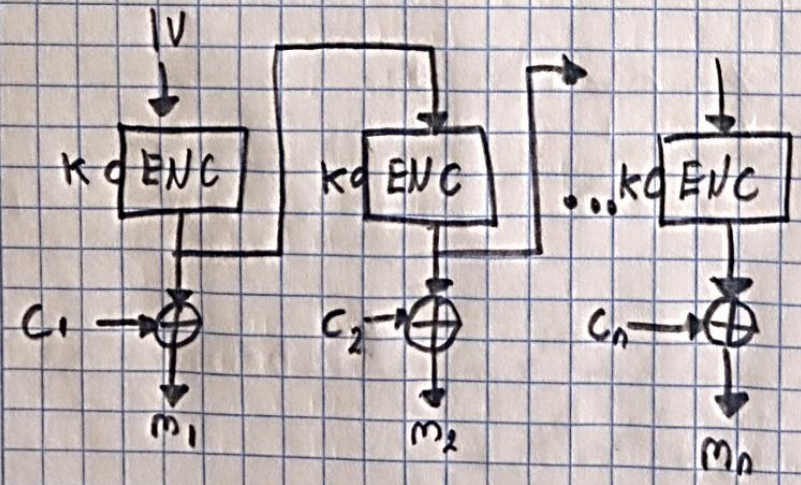
modo de operación OFB

Emplea una clave para crear un bloque pseudoaleatorio que es operado a través de XOR con el texto claro para generar el texto cifrado. Requiere de un vector de inicialización que debe ser único para cada ejecución realizada, el cifrado se muestra en el siguiente diagrama



Para el descifrado se utiliza el siguiente diagrama



Pseudo código para cifrado de los modos de operación

CBC

```
Cifrado CBC (textoCifrar, k)

    i=0
    j=0
    IV= generar vector de inicialización
    repetir hasta que se termine la cadena textoCifrar
        repetir hasta i=64
            m[j][i] = textoCifrar[i]
            i = i+1
        sino completa los 64 bits
            aplicar padding
        recortar primeros 64 bits de textoCifrar
        j = j+1
    repetir hasta que no haya bloques de bits
        i=0
        j=0
        repetir hasta que la cadena se termine
            si IV[i] es igual a m[j][i] entonces
                varXor[i] = 0
            sino
                varXor[i] = 1
            i = i+1
        c[j] = cifrar 3Des a varXor con la llave k
        IV = C[j]
        j = j+1
    regresar C e IV


Cifrado CTR (textoCifrar, k)

    i=0
    j=0
    repetir hasta que se termine la cadena texto cifrar
        repetir hasta i=64
            m[j][i] = textoCifrar[i]
            i = i+1
        si no completa los 64 bits
            aplicar padding
```

recortar primeros 64 bits de textoDescifrar
j = j+1
j=0
repetir hasta que no haya bloques de bits
x = bloque contador
i=0
cont = cifrar 3DES a x con la llave k
Repetir hasta que la cadena se termine cont
si cont[i] es igual a n[j][i] entonces
C[j][i] = 0
sino
C[j][i] = 1
i = i+1
j = j+1
x = x+1
regresar C

cifrado CFB (textoCifrar, k)
i=0
j=0
IV = generar vector de inicialización
repetir hasta que se termine la cadena texto cifrar
repetir hasta i=64
M[j][i] = textoCifrar[i]
i = i+1
sino completa los 64 bits
aplicar padding
recortar primeros 64 bits de texto cifrar
j = j+1
j=0
repetir hasta que no haya bloques de bits
i=0
auxIV = cifrar 3DES a IV con la llave k
repetir hasta que se termine la cadena auxIV
si auxIV[i] es igual a n[j][i] entonces
varXor = 0
sino
varXor = 1
i = i+1
C[j] = varXor
IV = C[j]
j = j+1
regresar C e IV

```
cifrado OFB (textoCifrar, k)
    i=0
    j=0
    IV = generar vector de inicialización
    repetir hasta que se termine la cadena
        repetir hasta i=64
            m[j][i] = texto cifrar[i]
            i=i+1
        sino completa los 64 bits
            aplicar padoding
        recortar primeros 64 bits de textoCifrar
        j=j+1
    j=0
    repetir hasta que no haya bloques de bits
        i=0
        auxIV = cifrar 3DES a IV con la llave k
        IV = auxIV
        Repetir hasta que la cadena auxIV termine
            si auxIV[i] es igual a m[j][i] entonces
                c[j][i]=0
            sino
                c[j][i]=1
            i=i+1
        j=j+1
    regresar c e IV
```

Pseudo código para descifrado de los modos de operación

```
descifrado CBC (textoDescifrar, k, IV)
    i=0
    j=0
    repetir hasta que se termine la cadena
        repetir hasta i=64
            c[j][i] = textoDescifrar[i]
            i=i+1
        recortar primeros 64 bit de textoDescifrar
        j=j+1
    j=0
    repetir hasta que no haya bloques de bits
        i=0
        auxC = descifrar 3Des a c[j] con la llave k
        repetir hasta que auxC termine
```

```
            Si IV[i] es igual a aux C[i] entonces
                m[j][i]=0
            sino
                m[j][i]=1
            i=i+1
        IV=C[j]
        j=j+1
    regresar m

desCifrado CTR (textoDescifrar, k)
    i=0
    j=0
    repetir hasta que se termine la cadena textoDescifrar
        repetir hasta i=64
            c[j][i]=textoDescifrar[i]
            i=i+1
        recortar primeros 64 bits de textoCifrar
        j=j+1
    j=0
    repetir hasta que no haya bloques de bits
        x=bloque contador
        i=0
        cont=cifrar 3DES a x con la llave k
        repetir hasta que se termine la cadena cont
            si C[i] es igual a cont[i] entonces
                m[j][i]=0
            sino
                m[j][i]=1
            i=i+1
        x=x+1
        j=j+1
    regresar m

desCifrado CFB (textoDescifrar, k, IV)
    i=0
    j=0
    repetir hasta que se termine la cadena textoDescifrar
        repetir hasta i=64
            C[j][i]=textoDescifrar[i]
            i=i+1
        recortar primeros 64 bits de textoCifrar
        j=j+1
    j=0
```

```
repetir hasta que no haya bloques de bits
    i=0
    aux IV = cifrar 3DES a IV con la llave k
    Repetir hasta que la cadena se termine
        si C[j][i] es igual a aux IV[i] entonces
            M[j][i] = 0
        sino
            M[j][i] = 1
        i = i+1
    IV = C[j]
    j = j+1
regresar M

desafiado OFB (textoDescifrar, k, IV)
    i=0
    j=0
    repetir hasta que se termine la cadena textoDescifrar
        repetir hasta i = 64
            C[j][i] = textoDescifrar[i]
            i = i+1
        recortar primeros 64 bits de textoDescifrar
        j = j+1
    repetir hasta no haya bloques de bits
        i=0
        aux IV = cifrar 3DES a IV con la llave k
        IV = aux IV
        Repetir hasta que la cadena se termine
            si C[j][i] es igual a aux IV entonces
                M[j][i] = 0
            sino
                M[j][i] = 1
            i = i+1
        j = j+1
    regresar M
```