CriptoSistemas Básicos
Clase 2/2

Criptografía
2019

Instituto de Computación
Facultad de Ingeniería
Universidad de la República

Contenido

AES: Key Schedule

- AES utiliza claves de 128, 192 y 256 bits.
- Si definimos l_k como palabras de 32-bits (4 bytes) tenemos:

key length		
in bits	in ℓ_k words	ℓ_r rounds
128	4	10
192	6	12
256	8	14

AES: Key Schedule

- Cada palabra l_k la modelamos como columna de una matriz.
- Tendremos 4 (128) ó 6 (192) ó 8 (256) columnas según la longitud de la clave.
- En cada ronda necesitamos un arreglo *round key* de l_k palabras y una más para la ronda inicial.
- La concatenación de las rondas de claves es lo que llamamos *EXPANDED KEY*.
- La lógica está diseñada para que si se cambia un bit de la clave, esto afectará las round keys de las siguientes rondas.

AES-128 - Key Schedule

Entonces, para 128-bit:

- cada ronda se compone de 4 palabras.
- La clave extendida tiene 44 palabras.
- La clave secreta es la entrada del algoritmo de clave extendida, ie, 4 palabras (128 bits). Es tratada como un arreglo de 16 bytes.

$$\begin{bmatrix} k_0 & k_4 & k_8 & k_{12} \\ k_1 & k_5 & k_9 & k_{13} \\ k_2 & k_6 & k_{10} & k_{14} \\ k_3 & k_7 & k_{11} & k_{15} \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} w_0 & w_1 & w_2 & w_3 \end{bmatrix}$$

AES-128 - Key Schedule

AES Key Expansion

- Se agregan dos operaciones que llamaremos:
  - ROTWORD: efectúa un shift cíclico (izquierda) de los cuatro bytes. Si se piensa la palabra como un elemento de $F_{256}[s]/(s^4+1)$, esto es multiplicar por s^3.
  - SUBWORD: aplica AES S-box a cada uno de los cuatro bytes.
- Además se definen palabras constantes para cada ronda que no dependen ni del texto claro ni de la clave. Le llamaremos ROUND CONSTANT.

AES-128: Key Expansion - Algoritmo

- Digamos que tenemos cuatro palabras de la i-ésima ronda:

$$w_i, w_{i+1}, w_{i+2}, w_{i+3}$$

- i debe ser un múltiplo de 4. Esto será la round key de la $(i/4)^{th}$ ronda
- Por ejemplo, w_4, w_5, w_6, w_7 es la round key de la ronda 1.
- Ahora necesitamos determinar:

$$w_{i+4}, w_{i+5}, w_{i+6}, w_{i+7}$$

  a partir de las palabras

$$w_i, w_{i+1}, w_{i+2}, w_{i+3}$$

AES-128: Key Expansion - Algoritmo

- Tenemos que:

$$
\begin{aligned}
w_{i+5} &= w_{i+4} \otimes w_{i+1} \\
w_{i+6} &= w_{i+5} \otimes w_{i+2} \\
w_{i+7} &= w_{i+6} \otimes w_{i+3}
\end{aligned}
$$

- Pero cuando el número de palabra es múltiplo de 4, hacemos algo distinto. Es decir, en la primer palabra de cada nuevo grupo de 4 palabras.

$$
w_{i+4} = w_i \otimes g(w_{i+3})
$$

AES-128: Key Expansion - Algoritmo

Definimos la función g que consiste de 3 pasos:

- Se aplica la función ROTWORD (shift).
- Se aplica la función SUBWORD (sustitución de bytes utilizando S-Box)
- Se aplica X-OR de lo obtenido anteriormente con lo que denominamos ROUND CONSTANT.
  ROUND CONSTANT es una palabra cuyos 3 bytes de más a la derecha son siempre ceros.

AES-128: Key Expansion - Algoritmo

La ROUND CONSTANT para la $i - esima$ ronda la denotaremos como $Rcon[i]$.

- Como por definición, los 3 bytes de más a la derecha son siempre ceros, podemos escribir:

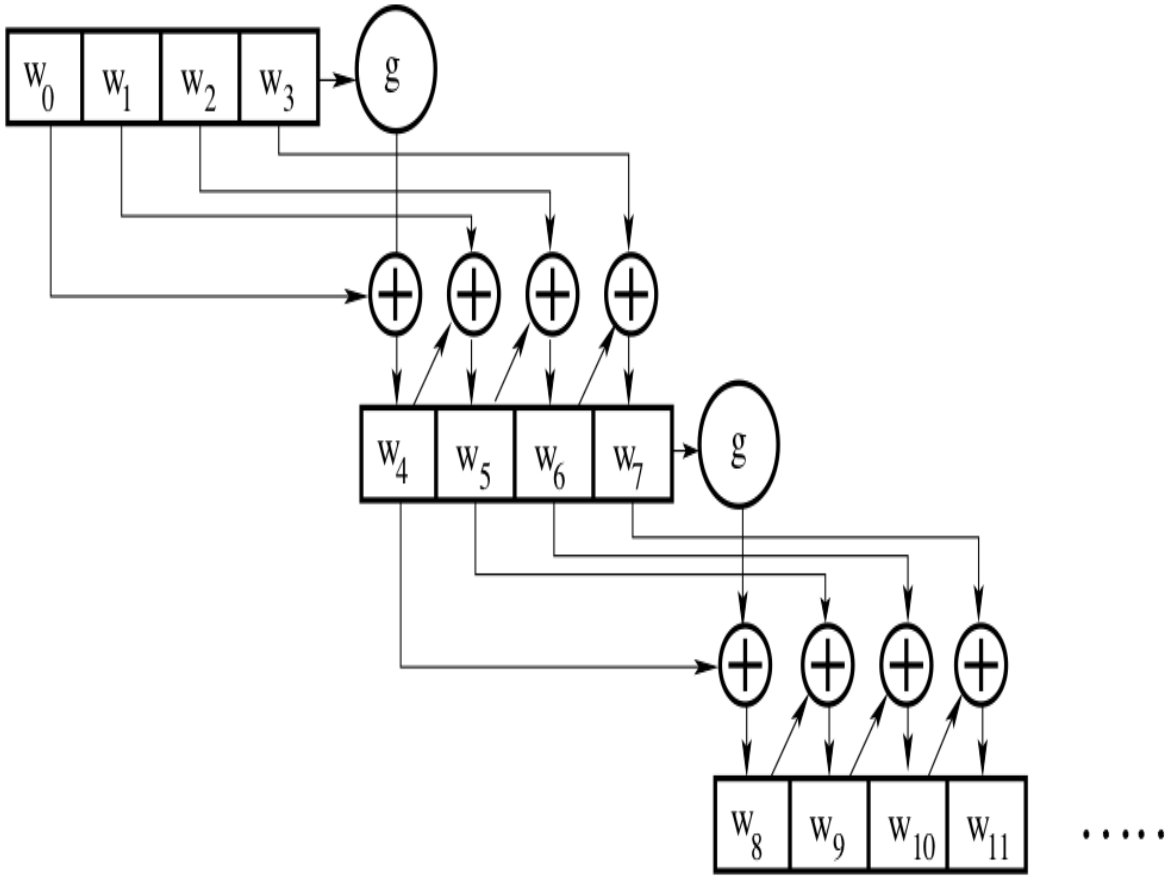
$$Rcon[i] = (RC[i], 0x00, 0x00, 0x00)$$

- $RC[i]$ se puede definir por recursión:

$$\begin{aligned} RC[1] &= 0\mathbf{x}01 \\ RC[j] &= 0\mathbf{x}02 \times RC[j-1] \end{aligned}$$

AES-128: Key Expansion - Algoritmo

Globalmente podemos ver el algoritmo así:



AES-128: Key Expansion - Algoritmo

Algoritmicamente lo podemos escribir así (Stinson 3.6)

Algorithm 3.6: KEYEXPANSION(key)

external ROTWORD, SUBWORD
$RCon[1] \leftarrow$ 01000000
$RCon[2] \leftarrow$ 02000000
$RCon[3] \leftarrow$ 04000000
$RCon[4] \leftarrow$ 08000000
$RCon[5] \leftarrow$ 10000000
$RCon[6] \leftarrow$ 20000000
$RCon[7] \leftarrow$ 40000000
$RCon[8] \leftarrow$ 80000000
$RCon[9] \leftarrow$ 1B000000
$RCon[10] \leftarrow$ 36000000
for $i \leftarrow 0$ **to** 3
  do $w[i] \leftarrow (key[4i], key[4i+1], key[4i+2], key[4i+3])$
for $i \leftarrow 4$ **to** 43
  do
  $temp \leftarrow w[i-1]$
  if $i \equiv 0 \pmod 4$
    then $temp \leftarrow$ SUBWORD(ROTWORD($temp$)) $\oplus\ RCon[i/4]$
  $w[i] \leftarrow w[i-4] \oplus temp$
return $(w[0], \ldots, w[43])$

AES-128: Key Expansion

- El algoritmo de KEY Expansion asegura que no tendremos claves débiles en el sentido que no se reduce la seguridad del cifrado de una forma predecible.
  Por ejemplo, DES tiene claves débiles que producen claves idénticas entre las rondas (alternando ceros y unos). Esto causa que la encriptación sea invertible por si misma, es decir, si encripto dos veces el mismo texto con la misma clave, obtengo el texto claro.

AES-128: Animación

Animación de AES:

- https://www.youtube.com/watch?v=gP4PqVGudtg

Acuerdo de Claves

- En los sistemas criptográficos simétricos, se debe compartir la clave secreta.
- Este acuerdo de clave, usualmente se hace a través de un canal inseguro.

Entonces,
¿pueden Alice y Bob acordar una clave secreta en común utilizando un canal inseguro?

Acuerdo de Claves

Respuesta, si!

- Cada interlocutor envía al otro una versión "encubierta" de la clave a partir de la cual cada uno es capaz de computar la misma clave secreta en común.
- Un intruso, a pesar de ver las claves "encubiertas" no es capaz de computar la misma clave en común a la que llegan los interlocutores.

¿Cómo?

- Acuerdo de claves de Diffie-Hellman

Diffie-Hellman Key Exchange

- Se trabaja con un grupo $G = Z_p^X$ modulo p, con p primo.
- Propiedades:
  - tiene $p - 1$ elementos
  - ningún elemento es divisible por p y tampoco el producto entre cualquiera de dos de ellos.
  - cualquier elemento tiene inverso multiplicativo. Este inverso puede ser computado por el algoritmo de Euclides.
  - es cíclico: hay un elemento generador g.

Diffie-Hellman Key Exchange

- El grupo G, el generador g y todos los mensajes transmitidos son públicos.
- Llamemos d al orden de G. El parámetro de seguridad n es la longitud en bits de d.
  Entonces, los elementos de G pueden ser representados por $n - bit\ strings$.

Diffie-Hellman Key Exchange

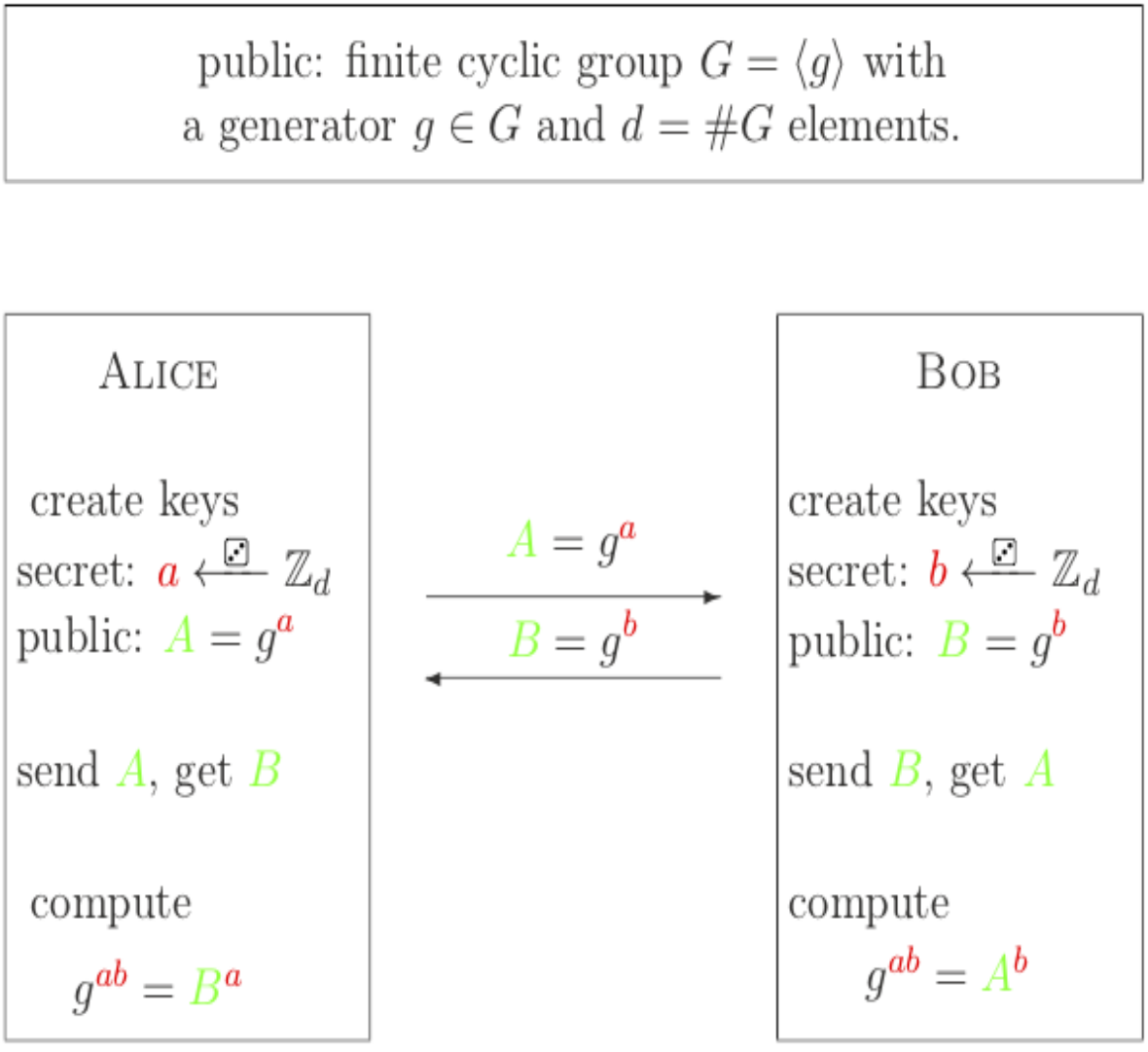
Algoritmo de acuerdo de claves.

- Alice y Bob individualmente seleccionan de forma aleatoria y uniforme los secretos a y b en Z_d respectivamente.
- Luego, cada uno publica sus claves públicas: $A = g^a$ y $B = g^b$.
- Ahora, cada uno computa localmente la clave secreta que usaran en los intercambios:

$$K = g^{ab} = B^a = A^b$$

Diffie-Hellman Key Exchange

Graficamente lo podemos ver así:



Diffie-Hellman Key Exchange

Para analizar el protocolo debemos responder:

- Correctitud: Alice y Bob computan la misma clave $g^{ab} = B^a = A^b$
- Eficiencia: La exponenciación es la operación más costosa. Por ejemplo, se puede utilizar el algoritmo de "repeated squaring", $O(n)$
- Seguridad: ¿es difícil para Eve recuperar la clave secreta?
  - Para esto debemos distinguir si Eve está efectuando su ataque en modo pasivo (solamente escuchando sin interferir en la comunicación) o está además interfiriendo en la comunicación.

Diffie-Hellman Key Exchange

Problema de Diffie-Hellman: DH_G

- Dado un grupo $G =< g >$ de orden d y A,B en G, computar $C \in G$ tal que:

$$A = g^a, B = g^b, C = g^{ab}$$

para algún a y b en Z_d

Diffie-Hellman Key Exchange

Decimos que (A, B, C) es una tripla de Diffie-Hellman si:

- dados un grupo $G =< g >$ de orden d, $a, b, c \in Z_d$ y $A = g^a$, $B = g^b$ y $C = g^c$ en G

$$\text{si } ab = c \text{ en } Z_d$$

Diffie-Hellman Key Exchange

Problema decisional de Diffie-Hellman: DDH_G

- dados un grupo $G =< g >$ de orden d y $A, B, C \in G$ decidir si (A, B, C) es una tripla de Diffie-Hellman.

Diffie-Hellman Key Exchange

Dificultad:

- Si DH_G (computar C) se puede resolver fácil, el problema decisional también se puede. En cambio, el camino inverso generalmente es falso.
- Si Eva pudiera computar a a partir de A, g y G, solamente le bastaría computar B^a para hallar el secreto común.
  Computar la exponenciación es fácil pero computar la inversa (logaritmo discreto) es difícil.

Diffie-Hellman Key Exchange

El problema del logaritmo discreto.

- no existe algoritmo eficaz conocido para resolver el problema en el caso general. Su complejidad en el caso promedio resulta tan difícil como el peor caso.
- Si Eva pudiera resolver de forma eficiente el logaritmo discreto, luego ella también podría quebrar el protocolo de Diffie-Hellman.

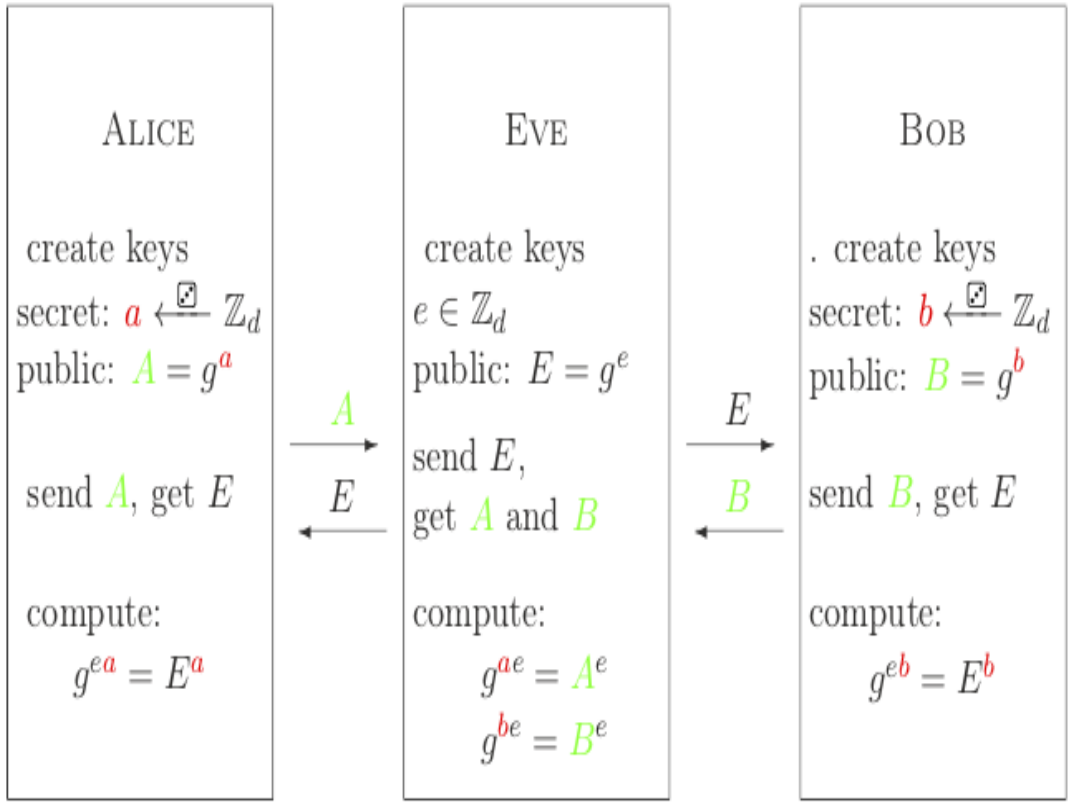
Diffie-Hellman Key Exchange

¿Ahora que pasaría si Eva además de escuchar puede interferir en la comunicación? Ataque: $man(woman) - in - the - middle$:

- Eva pretende ser Alice cuando habla con Bob y ser Bob cuando habla con Alice
- Alice y Bob creen de forma legitima que están hablando entre ellos, pero en realidad hablan con Eve.
- Eve de esta forma computa sus propias claves privadas con Alice y Bob.

Diffie-Hellman Key Exchange

Graficamente lo podemos ver así:



Diffie-Hellman Key Exchange

- Entonces, de alguna forma Alice y Bob deben autenticarse entre ellos antes de comenzar a hablar.

Cifrados por Bloque

- Los cifrados por bloques encriptan bloques de cierta longitud n.
- Por ejemplo, vimos que en AES-128 se encriptan bloques de 128 bits.
- Entonces, se debe considerar como encriptar mensajes largos que contengan más de un bloque.
  Para ello se definen los modos de operación.

Modos de operación

Los modos de operación fueron estandarizados en FIPS 81 en diciembre de 1980. Los modos de operación se pueden utilizar para cualquier cifrado por bloques.
Algunos de los modos de operación son:

- ECB: Electronic Codebook Mode
- CFB: Cipher Feedback Mode
- CBC: Output Feedback Mode

Modo de Operación: ECB

- Dada una secuencia de bloques de textos claros $x_1, x_2, ...$, cada x_i es encriptado con la misma clave K produciendo los bloques cifrados $y_1, y_2, ...$
- La debilidad notoria es que a iguales bloques de texto claros x_i se producirá idénticos bloques cifrados.

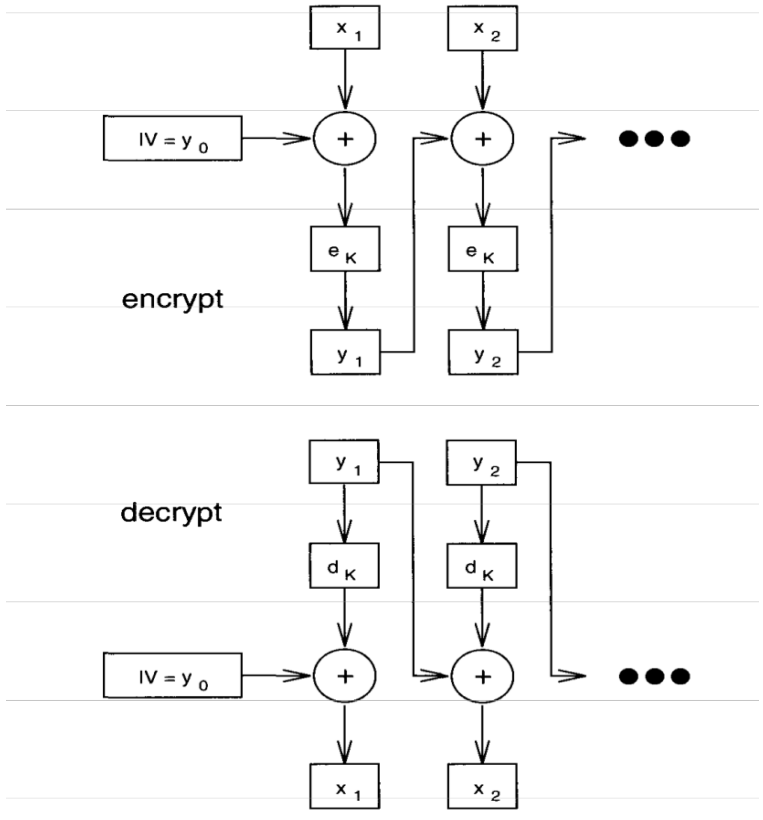
Modo de Operación: CBC

- Para cada bloque en claro x_i se hace un x-or con el bloque cifrado anterior y_{i-1} antes de ser encriptado.
- Para el primer caso, se comienza con u vector de incialización, IV que tiene la misma longitud que un bloque.
- Se tiene que:

$$y_i = e_K(y_{i-1} \oplus x_i)$$

Modo de Operación: CBC

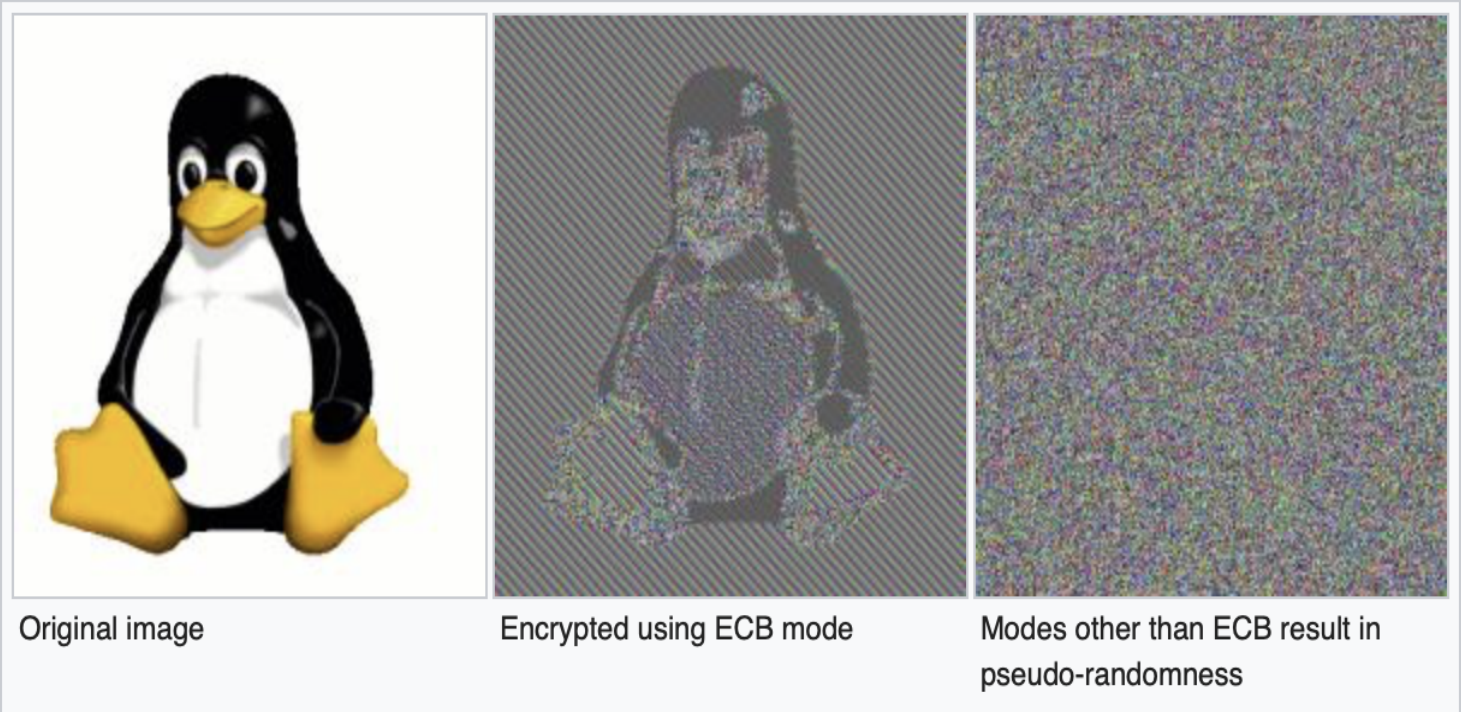
Graficamente lo podemos ver así:



Observar que si un bloque x_i es cambiado en este modo luego el y_i y todos los siguientes serán afectados.

Modos de Operación: ECB vs CBC

Comparemos la misma imagen cifrada con estos dos modos de operación:



Original image

Encrypted using ECB mode

Modes other than ECB result in pseudo-randomness

¿Qué se puede deducir?

The end.