Preguntas

¿Cuál de las siguientes ideas se relaciona con los cifradores de flujo?

Requieren una serie cífrante, idealmente infinita y aleatoria

¿Cuál es la función de los postulados de Golomb?

Establecen propiedades deseables que una serie cifrante debe cumplir

También llamado cifrado simétrico

Sistema de una clave

Según la forma de procesar el texto, los sistemas criptográficos se clasifican en cifrador de bloque o cifrador de flujo

Verdadero

Este cifrador toma el texto original y lo cifra carácter por carácter

Serial o Flujo

Este cifrado toma el texto y lo divide en grupos de caracteres para ser cifrado

Bloque

¿El cifrado DES ha sido vulnerado básicamente por?

El tamaño de la clave

Registro de desplazamiento con retroalimentación lineal (LFSR)

Permite crear una secuencia, a partir de un valor inicial

¿Cuántas celdas tendrá un LFSR descrito con el polinomio x5 + x3 + x2 + 1? ¿Y cuantas entradas XOR?

5 celdas, 3 entradas

Si se utiliza la clave varias veces de un cifrado de flujo

Si se conoce el texto en claro, se podría obtener la clave de cifrado

Fundamentos de cifra simétrica de bloque

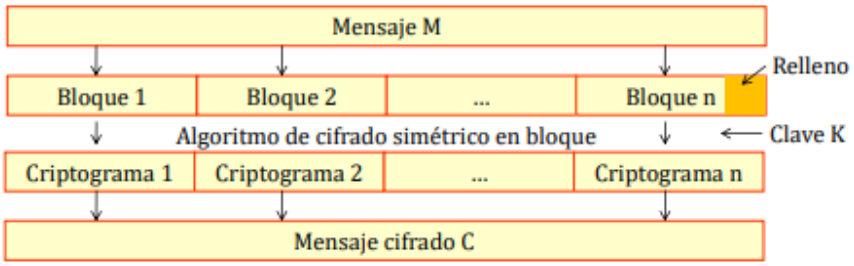
En criptografía moderna en lugar de formar bloques de letras se forman bloques de bytes.

Entre los algoritmos más conocidos de la cifra en bloque moderna está el DES, IDEA, AES.

Cifrado y tamaño típico del bloque

- Cifrado en bloque es aquella cifra en la que el mensaje original se agrupa en bloques de "x" bytes antes de proceder a aplicar el proceso de cifrado
- Un bloque pequeño (1-3 bytes) facilitaría un ataque por estadísticas del lenguaje
- Un bloque grande supondría un tratamiento no adecuado del texto en claro a cifrar
- Se debe de usar bloques de 8 8 16 bytes, 64 8 128 bits, actualmente se usan 16 bytes.

Esquema de la cifra simétrica en bloque



- El mensaje M se divide en bloques. Es posible que el ultimo bloque necesite relleno, dependiendo del algoritmo
- El criptograma C será igual a la concatenación C1+C2+C3….+Cn-1+Cn
- El problema es que sean bloques independientes forzará a modos de cifra

---

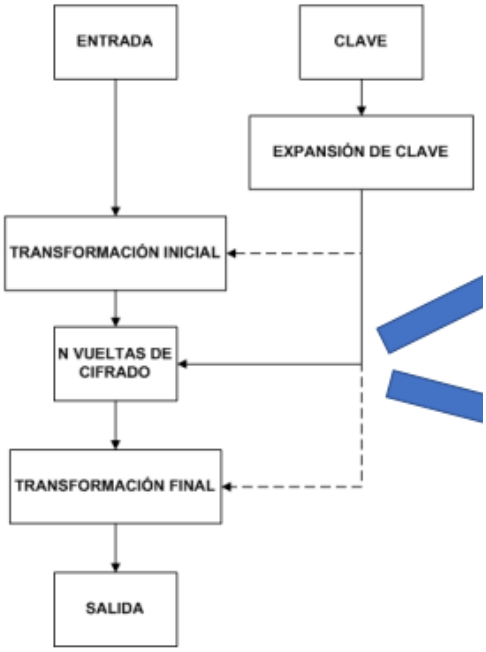
Características de cifra en bloque

- Dependencias entre símbolos
- Cambio en los bits de entrada
- Cambio en los bits de la clave
- Errores de transmisión

Estructura de la cifra en bloque

1. Una transformación inicial en algunos cifradores
    a. Carece de significado criptográfico
    b. Cambia la posición de los datos de entrada, oculta bloques de datos
2. El algoritmo de expansión de la clave. Convierte la clave introducida por el usuario en un conjunto de subclaves
    a. Diferentes para cada vuelta
    b. El conocimiento de una o varias subclaves intermedias, no debe permitir deducir las subclaves anteriores o siguientes
3. Las vueltas o rondas intermedias consisten en una función no lineal y compleja entre los datos de entrada y de la clave
4. Una transformación final en algunos cifradores
    a. Carece de significado criptográfico

b. Su función consiste en invertir la transformación inicial



Tamaño de bloque optimo

- DES, IDEA, Blowfish – bloques de 8 bytes, 64 bits (2^6 bits)
- AES, Twofish, Serpent – 16 bytes, 128 bits (2^7 bits)
- Bloques no muy cortos para evitar: Ataque Sweet32
- Bloques no muy largos para reducir el tamaño del texto cifrado implica mayores requisitos de memoria
- Un bloque de 128 bits (de 64 a 512 bits) puede ser implementado en hw de manera eficiente

---

Numero de vueltas o rondas

- Dos opciones: algoritmo con una única operación compleja versus otro algoritmo con una secuencia de operaciones sencillas que se repiten n veces
- Una vuelta es una operación de transformación
  - Se itera múltiples veces sobre dicha operación para diseñar el algoritmo de cifrado simétrico en bloque
- Mas sencillo de analizar desde el punto de vista de seguridad
- Todas las vueltas emplean la misma operación
  - Se diferencian en la clave de vuelta (diferente en cada vuelta)
  - Evitar ataques de desplazamiento (slide attacks)
- Claves de vuelta derivadas de la clave original
  - Algoritmo de expansión de clave (key Schedule)

¿Por qué solo estudiamos DES, IDEA y AES?

- DES es un cifrado tipo Feistel, estándar desde 1976 hasta 1999, es de muy fácil comprensión y usa una caja S de sustitución, similar a algoritmos modernos y actuales
- 3DES se sigue utilizando en cifra local o convencional
- IDEA (1991) hace uso de inversos multiplicativos, aditivos y xor para las claves de cifrado por lo que además tiene un importante interés didáctico

- AES es el nuevo estándar de cifra simétrica en bloque desde 2001

- Tanto en cifra clásica como moderna existe el modo de cifra en bloque
- Los tamaños de bloques eran 64bits (8bytes) y actualmente son 128 bits (16 bytes)
- Al formar bloques habrá que añadir rellenos y forzar a que exista unos modos de cifra en bloque
- Las características de un esquema de cifra en bloque son:
    - El tamaño del bloque de texto en claro
    - Las vueltas o rondas con su algoritmo de expansión de claves
    - Las redes de sustitución y permutación
- Entre los algoritmos de cifra simétrica en bloque más usados y conocidos se encuentra DES, 3DES, IDEA, Serpent, Twofish y AES

Algoritmo DES

Orígenes de DES

- 1973: La NBS (Actual NIST) llama a concurso publico para un algoritmo criptografico
- 1974: La NSA declara desierto el primer concurso. Publica una segunda especificación y gana Lucifer de IBM
- 1976: DES se adopta como estándar y se autoriza para ser utilizado en las comunicaciones no clasificadas
- 1976-199: DES fue utilizado como estándar mundial durante casi 25 años
- 1999: Después de ataques por fuerza bruta, en 1999 DES claudica.

Limitaciones en tamaños de bloques y clave

- La NSA impone una limitación en la longitud del bloque y de la clave
- De 128 bits de Lucifer, los deja en 64 bits
- Pero la clave efectiva solo son 56 bits puesto que son números de 8 bits, donde el octavo bit es de paridad
- Espacio de claves: $2^{56} = 7{,}2x10^{16} = 72.057.594.037.927.936$
- Versiones sobre esta reducción del espacio de claves
- Dificultad para diseñar chips capaces de operar de forma eficiente con una clave de 128 bits en los 70
- Política de seguridad interna para proteger información sensible ante ataques externos y poder realizar fuerza bruta en un tiempo razonable

---

Red de Feistel en el DES

- Cifra bloques de texto en claro de 64 bits
- Usa una clave de 64 bits, que se ve reducida a 56 bits por el bit de paridad
- Usa 16 vueltas con claves k1 a k16 para la operación de cifrado
- Como se fuerza a que k16 = k1 entonces para el descifrado se recorre el algoritmo en sentido inverso
- La seguridad de DES reside en la función no lineal f basada en las cajas S (sustitución)

Operaciones con las cajas S

- Cada caja tiene 16 columnas y 4 filas. Cada celda de una fila tiene números distintos del 0 al 15

- En cada Caja-S se obtienen 4 bits de salida por cada 6 bits de entrada
- Se trata de una operación no lineal y unidireccional, ya que habrá cuatro soluciones posibles de entrada para cada una de las salidas

Ejemplo de operación de una cajas S

- Sean **101100** los bits 7 al 12 de la cadena de 48 bits
- Por lo tanto debemos leer en la caja S_2

COLUMNAS

S_2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

FILAS

- Se selecciona la fila de la caja con los bits extremos: $10_2 = 2$
- Los cuatro bits centrales sirven para determinar la columna: $0110_2 = 6$
- La celda intersección entre la fila 2 y la columna 6 de la caja S_2 es el número de salida de cuatro bits buscado: 13 = 1101 = 0x D

Fortaleza de las cajas S

- Son funciones no lineales, donde es muy fácil pasar de 48 bits a 32 bits, pero computacionalmente es muy difícil hacer el recorrido inverso
- Para cada salida en una caja S, existen 4 entradas posibles.
- Romper las 8 cajas S en 16 vueltas de un bloque, significa 2 ala 256 cálculos.

- Usa una red de Feistel para la cifra, mezclando con la clave en cada una de sus 16 vueltas a la mitad del texto en claro, que además va cambiando de posición
- Utiliza operaciones de permutación con expansión y permutación con compresión para adaptar el texto de entrada y la clave de cada vuelta a las cajas S de sustitución, que es donde reside la seguridad del algoritmo

---

ECB y CBC, modos de cifra con confidencialidad

Necesidad de los modos de cifra en bloque

- Como el mensaje M se divide en bloques M1,M2,…Mn, el criptograma es la concatenación C= C1+C2+…Cn
- Pero no se debe hacer una cifra de bloques independientes de esta manera, porque ello permitiría realizar ataques del tipo:
  - Inicios y finales iguales. Si se cifran dos mensajes diferentes con la misma clave, con inicios y/o finales iguales del texto en claro, el conocimiento del texto en claro del primero criptograma por parte del atacante, le permitiría deducir partes del segundo criptograma
  - Reenvío de bloques. Como los bloques C1 son independientes, un atacante podría retener una cifra cambiar los bloques y reenviarlos

---

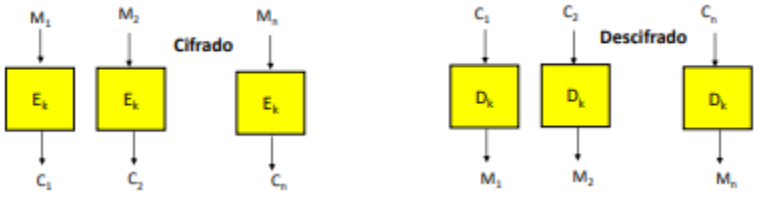
Modos de cifra aprobados por el NIST

- Ocho modos de confidencialidad (ECB, CBC, OFB, CFB, CTR, XTS-AES, FF1, FF3)
- Un modo de autenticación (CMAC)
- Cinco modos combinados por confidencialidad y autenticación (CCM, GCM, KW, KWP, y TKW)

Modo ECB

- El modo ECB libro electronico de códigos, cifrará cada bloque con la clave k de forma independiente
- Por lo tanto, el resultado es como si se codificase mediante un gran libro de códigos
- Codificar no es lo mismo que cifrar
- Se podría reconstruir ese gran libro electrónico de códigos sin necesidad de conocer la clave
- Se aplicará relleno a b bits en ultimo bloque

Esquema modo ECB en cifrado y descifrado



- Para romper la cifra, se podría reconstruir ese gran libro electrónico de códigos, sin necesidad de conocer la clave
- Y para textos muy formateados, como una imagen, la cifra en modo ECB no oculta el perfil de esa imagen

---

Ventajas y desventajas del modo ECB

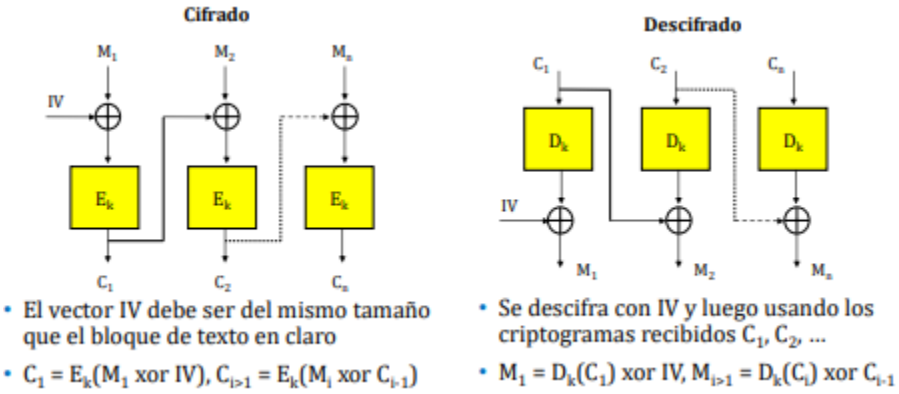
- Ventajas
  - La cifra es muy simple, se puede realizar cifrado y descifrado en paralelo
  - Un error se propaga solamente dentro del bloque en el que este ocurre
- Desventajas
  - Mensajes con inicios y finales iguales y cifrados con la misma clave, darán lugar al mismo criptograma
  - Para textos muy formateados, como una imagen, la cifra no oculta su perfil
  - Se podría vulnerar la integridad de un mensaje, capturando dos criptogramas donde se haya usado la misma clave de cifra, reorganizando los bloques de texto cifrado y el reenviando al destinatario uno o mas criptogramas diferentes de otros supuestos textos en claro

---

Modo CBC (Cipher block chaining)

- Encadenamiento de bloques cifrados, fue muy utilizado en SSL/TLS hasta el 2015
- Se realiza una operación xor entre el bloque del texto en claro y un bloque de b bits de igual tamaño que el bloque de texto, que actuará como una segunda clave y que no es necesario que sea secreta
- Para el primer bloque se usa un vector de inicialización IV
- En los siguientes bloques se usa como vector IV el criptograma anterior
- El vector IV es aleatorio. La probabilidad que dos vectores IV sean iguales es mínima
- Se aplicará relleno a b bits de ultimo bloque

Esquema modo CBC en cifrado y descifrado



- El vector IV debe ser del mismo tamaño que el bloque de texto en claro
- $C_1 = E_k(M_1 \text{ xor } IV)$, $C_{i>1} = E_k(M_i \text{ xor } C_{i-1})$
- Se descifra con IV y luego usando los criptogramas recibidos C_1, C_2, ...
- $M_1 = D_k(C_1) \text{ xor } IV$, $M_{i>1} = D_k(C_i) \text{ xor } C_{i-1}$

Ventajas y desventajas de CBC

- Ventajas
  - Se puede descifrar mas de un bloque usando los criptogramas anteriores
  - Si el vector IV es aleatorio y cambia en cada cifra, al cifrar dos veces el mismo documento con la misma clave, obtenemos criptogramas distintos
  - La cifra de archivos muy formateados, como una imagen, no mostrará su perfil
  - No será posible sustituir un bloque en el criptograma sin que se detecte
- Desventajas
  - Un error se propaga en todo ese bloque y en los siguientes
  - No es posible realizar una cifra simultanea de varios bloques pues la cifra desde el segundo bloque de texto depende del criptograma anterior

---

- Está prohibido el uso de ECB ya que permite ataques por inicios y finales iguales de texto en claro y ataques por reenvío de bloques seleccionados
- CBC, que fue muy popular hasta 2015, evita esos ataques porque al utilizar un vector de inicialización y ser éste aleatorio y diferente en cada cifra, cada operación de cifra, aunque sea con la misma clave K, entrega un criptograma diferente, pero sin embargo no permitirá un cifrado en paralelo

---

Cifra simétrica en flujo

El cifrado de Vernam

- Se usaba el código Baudot de 5 caracteres o bits
- La operación de cifra era la función XOR implementada con relés
- La clave K es una libreta de un solo uso, one-time pad, compartida con anterioridad y de forma segura entre emisor y receptor
- La clave debía ser aleatoria y tan larga o más que el mensaje
- El sistema de descifrado era igual que el de cifrado por la involución de la función XOR

Ejemplo de cifrado de Vernam

Con el código Baudot de 5 bits, cifrar el mensaje M = BYTES con clave K = VERNAM

- CIFRADO
- B ⊕ V = 11001 ⊕ 11110 = 00111 = U
- Y ⊕ E = 10101 ⊕ 00001 = 10100 = H
- T ⊕ R = 10000 ⊕ 01010 = 11010 = G
- E ⊕ N = 00001 ⊕ 01100 = 01101 = F
- S ⊕ A = 00101 ⊕ 00011 = 00110 = I

- DES CIFRADO
- U ⊕ V = 00111 ⊕ 11110 = 11001 = B
- H ⊕ E = 10100 ⊕ 00001 = 10101 = Y
- G ⊕ R = 11010 ⊕ 01010 = 10000 = T
- F ⊕ N = 01101 ⊕ 01100 = 00001 = E
- I ⊕ A = 00110 ⊕ 00011 = 00101 = S

- El esquema de Vernam es el único cifrador matemáticamente seguro
- Cuenta con secreto perfecto, demostrado por Claude Shannon
- Es imposible de criptoanalizar pues la clave K se usa una sola vez, es aleatoria y de longitud igual o mayor que el propio mensaje

---

Vernam, cifra en flujo y secreto perfecto

- Para que un sistema de cifra en flujo (en criptografía clásica cifra polialfabetica monogramica) tenga un secreto perfecto, deberá cumplir las condiciones del cifrado de vernam
  - Tener una clave aleatoria
  - La longitud de la clave sea de igual tamaño o mayor que el mensaje
  - La clave (libreta) sea de un único uso
- Para mensajes pequeños, se puede aceptar que la clave sea intercambiada entre emisor y receptor previamente, para mensajes grandes, esto ya no es practico
- Por tanto, habrá que inventar un sistema que permita generar la clave en ambos extremos, emisor y receptor.
- Esto hará que la clave ya no sea aleatoria sino pseudoaleatoria y s perdería el secreto perfecto

---

Cifra en flujo

- El mensaje en claro se cifra bit a bit o byte a byte, mediante una operación XOR entre el texto en claro y una secuencia cifrante Si
- Emisor y receptor comparten de forma segura una semilla secreta K y un circuito generador de la secuencia de bit Si que permita generar secuencias binarias pseudoaleatorias.
- Para descifrar se vuelve a realizar el XOR entre criptograma y la misma secuencia Si en destino

---

Características secuencias cifrantes

- La secuencia cifrante Si debe tener un periodo muy elevado, tanto o mas que el propio mensaje
- Para generar la misma secuencia cifrante Si en fase, ambos extremos usarán una semilla K que no puede ser un valor demasiado pequeño
- Debe ser fácil su implementación y que permita obtener algoritmos rápidos
- Se puede lograr con hw o sf mediante diferentes sistemas o circuitos
- Si debe aproximarse a una secuencia aleatoria
- Cada generador de Si debe cumplir con los 3 postulados de Golomb
  - Postulado G1: mitad de 0s y de 1s (probabilidad 50%)
  - Postulado G2: probabilidad 50% independiente de bits
  - Postulado G3: la información Si derivada de zonas con diferentes cadenas de bits conocidos, debe ser nula

Registros de desplazamiento realimentados lineales y no lineales: LFSR y NLFSR

Generadores de secuencias cifrantes

- Los generadores de la secuencia cifrante pueden ser de los siguientes tipos:
  - Registros de desplazamiento retroalimentados: Algoritmo A5
  - Registros numéricos. Algoritmo RC4
  - Algoritmo de cifra simétrica en bloque. Algoritmo AES en modos de cifra contador CTR

Registro de desplazamiento realimentado

- Los FSR son la base de muchos de los generadores de secuencias cifrantes.
- Las celdas contendrán inicialmente un valor 0 o 1, que será parte de la semilla o clave inicial
- El circuito estará controlada por un reloj
- Sn se convierte en el nuevo bit de S1 y S1 se obtiene como resultado de aplicar la función F al contenido de las n celdas (0 o 1) conectadas, antes del desplazamiento hacia la derecha del registro

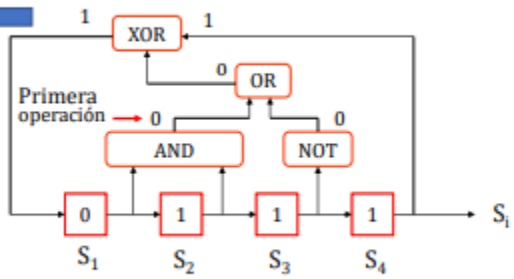
---

Generadores con registros NLFSR

Ejemplo con generador NLFSR de cuatro celdas (n = 4)

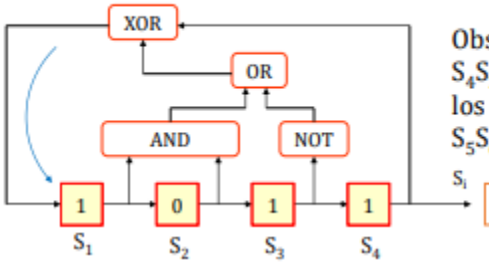
Este es el estado de las celdas y operaciones previas en el registro, antes de realizar el desplazamiento de un bit hacia a la derecha

El bit S_4 (1) saldrá como S_i y los demás bits (0, 1, 1) se desplazan a la derecha y dejan a la celda S_1 vacía, sin valor



Sea la semilla: $S_1S_2S_3S_4 = 0111$

Semilla: $S_1S_2S_3S_4 = 0111$



Observa que se transmite primero $S_4S_3S_2S_1$ (semilla al revés) y luego los restantes bits de la secuencia $S_5S_6S_7S_8S_9S_{10}S_{11}S_{12}S_{11}S_{14}S_{15}S_{16}$

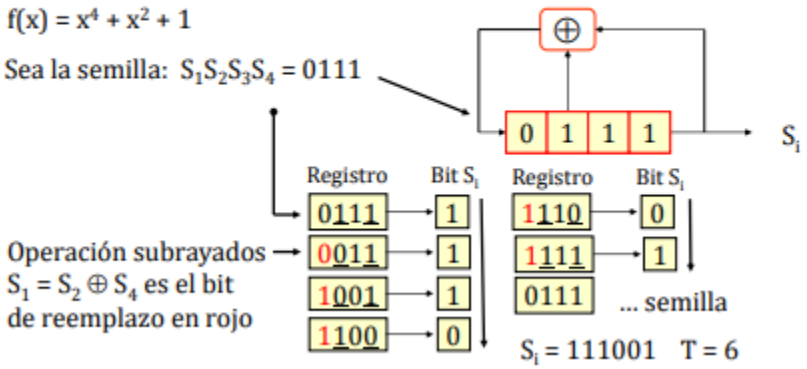
- Si = 1110 1100 1010 0001. Su periodo es máximo: Tmax = 2alan = 2ala4 = 16
- Se conoce como secuencia de Bruijin. El contenido de las celdas pasará por todos los estados posibles del registro, desde 0000 hasta 1111, no en orden

---

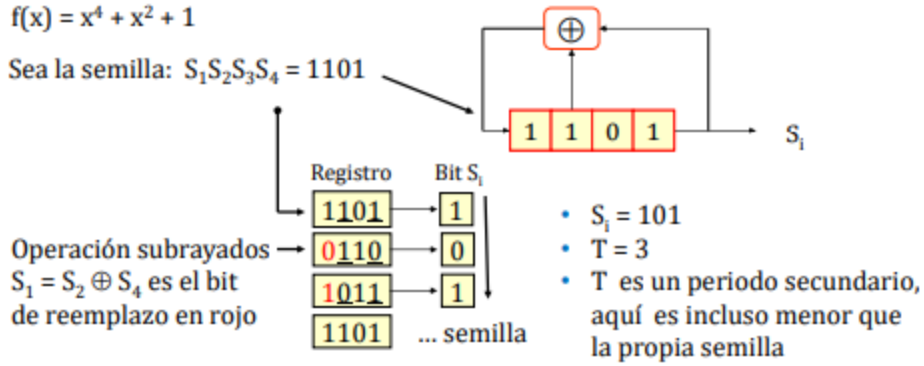
Generadores con registros LFSR

- Polinomios factorizables (cuando pueden expresarse como producto de dos o más polinomios de menor grado). Se obtienen secuencias con un periodo inferior a Tmáx y estos valores además dependen de la semilla utilizada
- Polinomios irreducibles (cuando son no factorizables). En este caso se obtienen secuencias con un periodo fijo, que no depende de la semilla, pero son inferior a Tmáx al ser un múltiplo de ésta
- Polinomios primitivos (cuando, además de no factorizables, generan todos los restos). Son los más adecuados y los que deben usarse, ya que se obtienen secuencias máximas (m-secuencias) con periodo fijo Tmáx = 2alan - 1

Ejemplo LFSR con f(x) factorizable (1/2)



Ejemplo LFSR con f(x) factorizable (2/2)



Aleatoriedad en registros LFSR con polinomio primitivo: 1. Postulado de Golomb

La seguridad en la cifra en flujo

- La seguridad de los sistemas de cifra en flujo depende de la secuencia cifrante Si que debe tener:
  - La secuencia Si debe tener una longitud en bits mayor que la del mensaje M
  - Aunque la secuencia Si sea pseudoaleatoria, debe tener una apariencia aleatoria
  - La semilla K con la cual se inicia el circuito generador de secuencia Si deberá usarse una sola vez

---

Aleatoriedad de una secuencia de bits LFSR

- Existen varios test para poder demostrar la aleatoriedad de una secuencia

---

Postulado R1 de Golomb

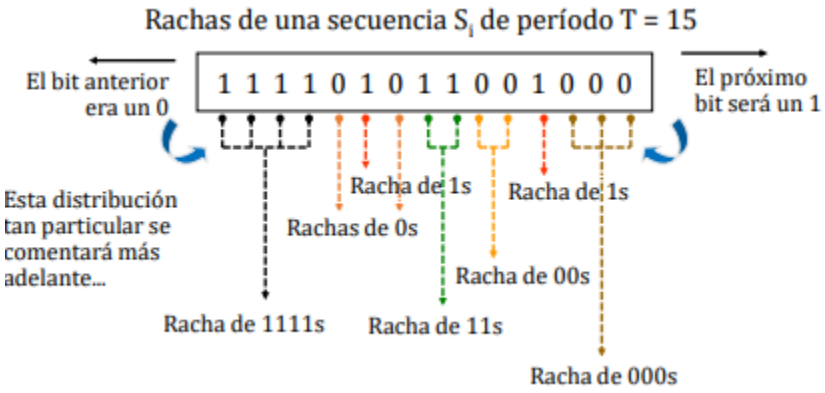
- Equiprobabilidad de bits tomados independientemente
- Deberá existir igual numero de ceros que de unos. Se acepta como máximo una diferencia igual a la unidad (en favor de los unos)
    - S1 1 1 1 1 0 1 0 1 1 0 0 1 0 0 0 T=15
- La secuencia S1 cumple con R1 pues hay 8 unos y 7 ceros
    - S2 1 1 1 0 1 0 1 1 0 1 0 0 1 0 0 1 T=16
- La secuencia S2 no cumple con R1 pues hay 9 unos y 7 ceros

Significado del postulado R1 de Golomb

- Si una secuencia Si como la indicada cumple con R1, quiere decir que la probabilidad de recibir un bit 1 es igual a la de recibir un bit 0, es decir un 50% de equiprobabilidad
- Por lo tanto, a lo largo de una secuencia Si, independientemente de cual bit se analiza, en media será igualmente probable recibir un 1 que un 0

Postulado R2 de Golomb y rachas de bits

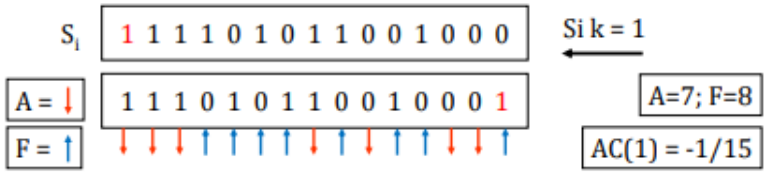
- Equiprobabilidad de próximo bit con secuencia anterior conocida
- En un periodo T la mitad de las rachas de Si serán de longitud 1, la cuarta parte de la longitud 2, la octava parte de longitud 3, etc., lo que se conoce como una distribución geométrica
- Una racha de bits es un conjunto de L bits iguales, separados por bits diferentes. Así, 010 es una racha de longitud de uno de 1s, 1001 es una racha de longitud de dos 0s y 01110 es una racha de longitud de tres 1s



- Si una secuencia Si como la indicada cumple con R2, quiere decir que la probabilidad de recibir un bit 1 o un bit 0, después de haber recibido un conjunto de bits 0s o 1s sigue siendo la misma, es decir un 50%

Postulado R3 de Golomb

- La autocorrelación fuera de fase AC(k) de una cadena o secuencia de bits, desplazada k bits, debe permanecer constante para todos los desplazamientos, es decir, desde k = 1 hasta k = n – 1
- AC(k) = (A – F)/T, donde A son los aciertos (bits iguales) y F son los fallos (bits diferentes) al comparar la cadena con su desplazada



Significado del postulado R3 de Golomb

- Si una secuencia S_i cumple con el postulado R3, quiere decir que independientemente del trozo de secuencia elegido o conocido por el atacante, no tendrá una mayor cantidad de información que en otro trozo cualquiera de la misma
- Será imposible entonces deducir cómo continúa la secuencia desde ese trozo u obtener información de otro tipo… "analogía" con el ataque de Kasiski a Vigenère…