CriptoTema3.pdf

onafolch

Criptografia i Seguretat

3º Grado en Ingeniería de Datos

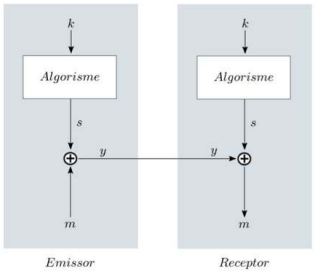
Escuela de Ingeniería
Universidad Autónoma de Barcelona

CRIPTOGRAFIA

CRIPTOGRAFIA SIMÈTRICA

1. LES XIFRES DE FLUX

Les xifres de flux sorgeixen com una aproximació optimitzada al xifratge de Vernam. La idea és construir una clau suficientment llarga, com a mínim de la longitud del missatge, a partir d'una clau inicial curta. Això s'aconsegueix utilitzant el que s'anomena un generador pseudoaleatori. L'operació d'expansió cal que tingui certes característiques ja que la seqüència que en resultarà és la que s'utilitzarà per xifrar el text en clar.



L'algorisme és determinista per tant la seqüència que en resulta no és completament aleatòria i a partir d'un cert moment es repeteix.

Tan l'emissor com el receptor disposen d'una mateixa clau k (anomenada llavor del generador), i d'un mateix algorisme determinista anomenat generador pseudoaleatori. Al proporcionar la clau k com a entrada a l'algorisme, aquest dona com a sortida una seqüència s que s'anomena seqüència xifrant.

Per tal de xifrar el missatge, l'emissor va sumant cada bit del missatge m amb cada bit de la seqüència xifrant s, obtenint el missatge xifrat y. Quan el receptor rep el missatge xifrat y, utilitza el mateix algorisme determinista i la clau k, que comparteix amb l'emissor, per tal d'obtenir la mateixa seqüència xifrant. Així sumant bit a bit el missatge que li arriba y, amb la seqüència resultant de l'algorisme s, obté el text en clar m enviat per l'emissor.

Per tal que aquest criptosistema sigui segur, és bàsic que la seqüència xifrant no sigui coneguda, és a dir que en cap moment es pugui saber quin serà el següent bit de sortida. Idealment, el que es necessita per a la seguretat incondicional és que la clau, en aquest cas la seqüència xifrant, sigui completament aleatòria. En el nostre esquema no es pot donar aquesta condició ja que el generador que utilitzem ha de ser determinista per tal que emissor i receptor obtinguin la mateixa seqüència quan donen com a entrada la mateixa clau secreta. Així doncs, la seqüència xifrant tindrà propietats molt properes a les que té una seqüència completament aleatòria i per tant s'anomenarà seqüència pseudoaleatòria.

Concretament, si una seqüència no és aleatòria, vol dir que a partir d'un cert moment es repeteix. Aquesta subseqüència que es va repetint és el que s'anomena període.

CSPRNG

Un generador pseudoaleatori criptogràficament segur generen seqüències no predictibles. Per a que un PRNG sigui considerat un CSPRNG, cal que les seqüències que genera tinguin dues propietats (a partir de k bits de la seqüència generada si+1, si+2, ..., si+k):

- No existeix un algorisme en temps polinomial que pugui predir el següent bit de la seqüència, s_{i+k+1}, amb probabilitat major al 50%.
- No és computacionalment possible predir el bit anterior de la seqüència, s_i.

Ona Folch

Tests d'aleatorietat del NIST

El **test de freqüència de bits individuals** comprova que la proporció d'uns i zeros de la seqüència proporcionada s'aproxima a la que observaríem en una seqüència veritablement aleatòria, és a dir, que la proporció d'uns i zeros és similar i s'aproxima, per tant, a 0.5. Per fer-ho, en primer lloc es transforma la seqüència binària d'entrada a una seqüència de −1 i 1. Els zeros es converteixen en -1 i els 1 es queden igual.

Després es calcula Sobs: $S_{obs} = \frac{\sum_{i=1}^{x} x_i}{\sqrt{n}}$

Si la seqüència és aleatòria S_{obs} tendirà cap a 0, mentre que si hi ha massa zeros o massa uns en la seqüència, aleshores aleshores S_{obs} tendirà a ser major a zero.

Exemple: Donada la seqüència S = {0,1,0,1,0,0,0,0,0,0,0,1,1,0,0,1,1,1,1,0,1,0,1,0,1,0,0,0} amb n = 28. Primer tranformem les dades en {−1,1,−1,1,−1,−1,−1,−1,−1,−1,−1,1,1,−1,−1,1,1,1,1,−1,1,−1,1,−1, 1, −1, −1, −1}, i calculem $S_{obs} = \frac{\sum_{i=1}^{28} x_i}{\sqrt{28}} = \frac{|-6|}{\sqrt{28}} \approx 1.1339$.

El test de freqüència en un bloc comprova que el número de zeros i uns en un bloc de m bits sigui aproximadament m/2. Per fer-ho, es particiona la seqüència a avaluar en b = n/m blocs de m bits, descartant els bits sobrants.

$$\overbrace{\underbrace{s_1, s_2, \ldots, s_m}_{m \text{ bits}}}^{k=1} \quad \overbrace{\underbrace{s_{m+1}, s_{m+2}, \ldots, s_{m+m}}_{m \text{ bits}}}^{k=2} \quad \cdots \quad \overbrace{\underbrace{s_{(b-1)m+1}, \ldots, s_{(b-1)m+m}}_{m \text{ bits}}}^{k=b}$$

Aleshores, per cada bloc k (amb k = 1, ... , b), s'utilitza la següent formula per saber la proporció d'uns que hi ha a cada bloc, i finalment es calcula X_{obs}^2

$$\pi_k = \frac{\sum_{j=1}^{m} s_{(k-1)m+j}}{m} \qquad X_{obs}^2 = 4m \sum_{k=1}^{b} (\pi_k - 0.5)^2$$

Exemple: donada la seqüència de l'exemple anterior, calculem π_k per cada bloc amb m = 6. Primer dividim en b = n/m = 28/6 = 4 blocs de m = 6 bits.

$$\overbrace{\underbrace{0,1,0,1,0,0}_{6 \text{ bits}}}^{k=1} \overbrace{\underbrace{0,0,0,0,0,1}_{6 \text{ bits}}}^{k=2} \overbrace{\underbrace{1,0,0,1,1,1}_{6 \text{ bits}}}^{k=3} \overbrace{\underbrace{1,0,1,0,1,0}_{6 \text{ bits}}}^{k=4} 1,0,0,0$$

Els últims 4 bits de la seqüència es descarten i no s'utilizen en el test. Aleshores, per cada bloc k (amb k = 1,⋯ ,4), calculem π_k:

$$\pi_1 = \frac{\sum_{j=1}^{m} s_{(k-1)m+j}}{m} = \frac{\sum_{j=1}^{6} s_j}{6} = \frac{2}{6} = 1/3$$
$$\pi_2 = \frac{\sum_{j=1}^{6} s_{6+j}}{6} = 1/6$$
$$\pi_3 = \frac{\sum_{j=1}^{6} s_{2\cdot 6+j}}{6} = \frac{4}{6} = 2/3$$
$$\pi_4 = \frac{\sum_{j=1}^{6} s_{3\cdot 6+j}}{6} = \frac{3}{6} = 1/2$$

I finalment calculem X_{obs}^2:

$$\begin{aligned}\chi_{obs}^2 &= 4m \sum_{k=1}^{b} (\pi_k - 1/2)^2 \\ &= 4 \cdot 6 \sum_{k=1}^{4} (\pi_k - 1/2)^2 \\ &= 24 \cdot \left((1/3 - 1/2)^2 + (1/6 - 1/2)^2 + (2/3 - 1/2)^2 + (1/2 - 1/2)^2\right) \\ &= 24 \cdot (1/36 + 1/9 + 1/36 + 0) = 4\end{aligned}$$

Ona Folch

El **test de ràfegues** (n > 100) comprova si el número de ràfegues tant d'uns com de zeros de la seqüència s'assembla al que trobaríem en una seqüència aleatòria. Definirem una ràfega com un conjunt de bits consecutius iguals, és a dir una ràfega de longitud k consta dels elements s_t,..., s_{t+k-1}, tals que $s_{t-1} \neq s_t = s_{t+1} = \cdots = s_{t+k-1} \neq s_{t+k}$

Per avaluar la prova de ràfegues, es calcula:

$$V_n(obs) = \left(\sum_{i=1}^{n-1} r(i)\right) + 1$$

on r(i) és la funció:

$$r(i) = \begin{cases} 0, & \text{si } s_i = s_{i+1} \\ 1, & \text{altrament} \end{cases}$$

Valors grans de Vobs indiquen que les oscil·lacions de valors en la seqüència avaluada succeeixen ràpidament (és a dir, els canvis entre zeros i uns). Addicionalment, aquest test té com a prerequisit que la seqüència passi el test de freqüència de bits individuals.

Exemple: amb la mateixa seqüència dels altres exercicis, primer de tot comprovem que superi el test de freqüència de bits individuals. Com que el supera, calculem $V_{28}(obs)$:

$$\begin{aligned} V_n(obs) &= \left(\sum_{i=1}^{n-1} r(i)\right) + 1 \\ &= (1+1+1+1+0+0+0+0+0+0+1+0+1+0+1+0+ \\ &+0+0+1+1+1+1+1+1+1+0+0)+1 = 15 \end{aligned}$$

2. GENERADORS LINEALS DE SEQÜÈNCIA XIFRANT

Tractem ara com han de ser els algorismes deterministes que generen aquests tipus de seqüències pseudoaleatòries. Tenim dos tipus de generadors: lineals i no lineals.

Els generadors lineals són aquells que només realitzen operacions lineals sobre els elements d'entrada per obtenir la seqüència de sortida. En canvi, els generadors no lineals són els que realitzen a més a més operacions no lineals, com podrien ser permutacions.

Generadors congruencials

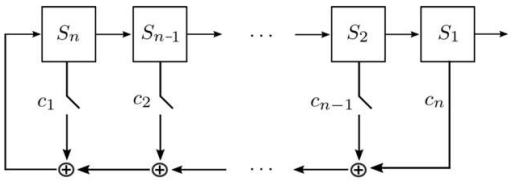
Es basen en equacions modulars recurrents del tipus $X_n = (ax_{n-1} + b)\ mod\ m$. En aquest cas, el valor X_0 seria la llavor de la seqüència xifrant. Un criptosistema que utilitzi un generador d'aquest tipus tindrà com a clau secreta $\{X_0, a, b, m\}$.

Per tal de no tenir un període molt petit, la n hauria de ser més petita a m (període màxim mcd(a,m)=1).

- Si m és primer i c = 0, el període és m-1 si a és un element primitiu en m.
- Si m és potència de 2 i c = 0, té un període com a màxim de m/4 (si a = 3 o a = 5 (mod 8)).
- Si c és diferent a 0, el període és m, si i només si mcd(m,c) = 1. a-1 és divisible per tots els factors primers

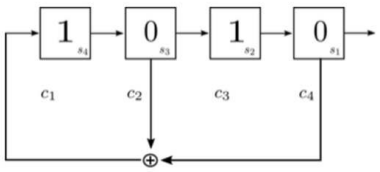
LFSR Linear Feedback Shift Register

Un registre de desplaçament realimentat linealment (LFSR) de longitud n és un dispositiu físic o lògic format per n cel·les de memòria i n portes lògiques, on l'estat inicial és {S1, ... , Sn}.



Ona Folch

Exemple: si l'estat inicial és 1010 i el polinomi de connexions és $C(x)=1+0x^1+1x^2+0x^3+1x^4=1+x^2+x^4$



Impuls de rellotge (t)	s_4	s_3	s_2	s_1	Sortida
0	1	0	1	0	0
1	0	1	0	1	1
2	0	0	1	0	0
3	0	0	0	1	1
4	1	0	0	0	0
5	0	1	0	0	0
6	1	0	1	0	0
7	0	1	0	1	1
⋮		⋮			⋮

L'impuls de rellotge és t=6 i, per tant, a partir d'aquí la seqüència es torna a repetir (període 6).

Definit el polinomi de connexions, ja podem determinar les característiques de l'LFSR d'acord amb les del seu polinomi de connexions:

1. **Polinomi de connexions factoritzable**: els LFSR's que tenen polinomis de connexions factoritzables generen seqüències que depenen de l'estat inicial. A més, el període d'aquestes seqüències és sempre més petit que el període màxim que pot tenir un LFSR, que és 2^n-1.
2. **Polinomi de connexions irreductible**: les seqüències no depenen de l'estat inicial. En aquest cas, el període serà un divisor de 2^n-1.
3. **Polinomi de connexions primitiu**: té la seqüència de sortida de període màxim, 2^n-1, i no depèn de l'estat inicial. Un polinomi primitiu és també irreductible.
   No hem d'oblidar que el nombre de polinomis primitius de grau n ve donat per l'expressió $\phi(2^n-1)/n$.

3. GENERADORS NO LINEALS DE SEQÜÈNCIA XIFRANT

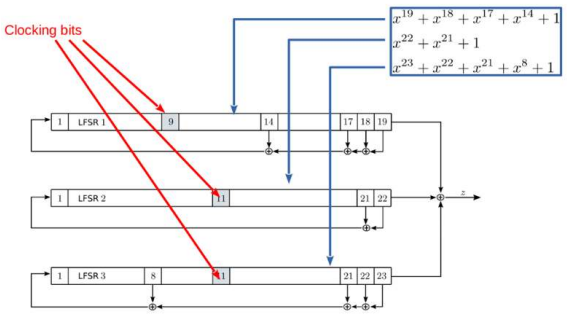
A5

És un criptosistema de flux que utilitza una combinació no lineal de la sortida de tres LFSR. Xifra cadenes de text en clar de 228 bits, i podem dividir el funcionament en tres etapes:

1. Incialització dels LFSR
2. Generar la seqüènica dels 228 bits a partir del LSFR
3. Xifrar els 228 bits, realitzant un XOR de la seqüència de xifrat amb el text en clar.

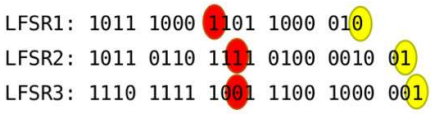
La no linealitat del sistema ve donada perquè a cada impuls de rellotge no tots els LFSR avancen. Només ho fan aquells LFSR els bits dels quals són majoria en les cel·les anomenades clocking bit (en el cas de l'esquema, els clocking bits són les cel·les marcades en gris).

Així, per exemple, si en la cel·la 9 del primer LFSR hi ha un 1, i en les cel·les 11 del segon i tercer LFSR hi ha un 0, només avançaran el segon i el tercer LFSR, que tenen un 0. Si els tres són iguals, aleshores avancen tots. D'aquesta manera es van obtenint les sortides de cada un dels LFSR que formen l'XOR que acabarà proporcionant cada bit de la seqüència de xifratge.



Ona Folch

Exemple: si d tres seqüències de LFSR:



El resultat de sortida serà z=0⊕1⊕1=0. Per calcular el següent estat dels LFSR, mirarem els clocking bits. Com que el bit majoritàri és el 1, LSFR1 I LSFR2 faran un shift cap a la dreta.

La inicialització d'A5 requereix dos valors, una clau de sessió de 64 bits i un número de trama de 22 bits, i consta de quatre passos:

1. Omplir tots els registres dels LFSR amb 0
2. S'executen 64 rotacions de rellotge dels tres LFSR sense fer servir clocking (avançen els 3). El bit de retroalimentació del LFSR fa un XOR amb el bit de la clau de sessió abans de ser inserit a la primera cel·la del LFDR. Cadascuna de les 64 rotacions fa servir un dels bits de la clau de sessió diferent, de manera seqüencial.
3. S'executen 22 rotacions sense clocking, però el bit de retroalimentació fa un XOR amb els bits del número de trama abans d'inserir-se de nou a la cel·la corresponent.
4. Es fan 100 rotacions amb clocking.

4. AES

El criptosistema AES xifra blocs de text en clar de 128 bits de longitud, agrupat en 16 bytes en una matriu. Les sumes es fan amb XOR, i hi ha polinomis de grau màxim 7. $m(x) = x^8 + x^4 + x^3 + x + 1$. Podem representar els blocs de 128 bits de 3 maneres diferents:

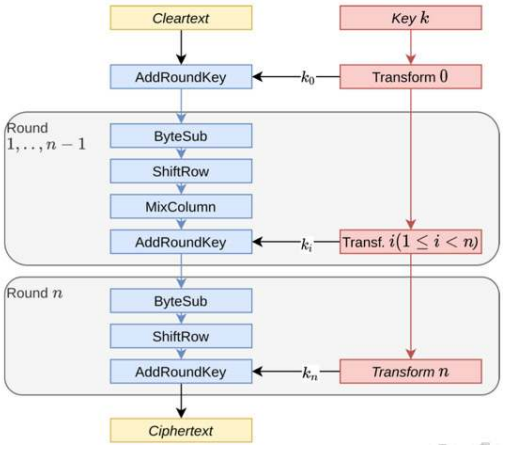
Binari	Hexadecimal	Polinomial
00000001 00000010 00000011 00000100 00000101 00000110 00000111 00001000 00001001 00001010 00001011 00001100 00001101 00001110 00001111 00010000	$\begin{pmatrix} 01 & 02 & 03 & 04 \\ 05 & 06 & 07 & 08 \\ 09 & 0a & 0b & 0c \\ 0d & 0e & 0f & 10 \end{pmatrix}$	$\begin{pmatrix} (1) & (x) & (x+1) & (x^2) \\ (x^2+1) & (x^2+x) & (x^2+x+1) & (x^3) \\ (x^3+1) & (x^3+x) & (x^3+x+1) & (x^3+x^2) \\ (x^3+x^2+1) & (x^3+x^2+x) & (x^3+x^2+x+1) & (x^4) \end{pmatrix}$

AddRoundKey: suma XOR de la matriu d'estat amb cada byte de la subclau corresponent.

SubBytes: substitució no lineal dels bytes de la matriu d'estat.

ShiftRow: desplaça les files de la matriu d'estat, on la fila 0 es queda igual, la 1 es desplaça una posició a l'esquerra, la 2 dues posicions, i la tercera 3 posicions a l'esquerra.

MixColumns: multiplica una matriu determinada per cada columna de la matriu d'estat, on les sumes són XOR, i les multiplicacions, on es pot fer de manera polinòmica fent mòdul m(x).

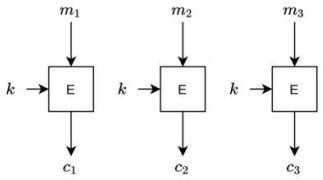


Generació de subclaus: a cada iteració s'utitiltza una clau diferent, que es calcula a partir de la primera. Per calcular la primera clau, agafem l'última columna de la clau actual i fem un shift cap adalt i d'aquesta columna fer un SubBytes, i seguidament fem un XOR amb la columna de 4 posicions enrere i amb la primera de Rcon [01,02,04,08,10,20,40,80,1b,36]. Per calcular les altres 3 columnes restants, agafem l'última columna (que correspon amb la primera de la clau actual), i fem XOR amb la columna de 4 posicions enrere. Això ho fem per cada clau fins tenir-ne 11.


SAMSUNG
Samsung Estudiantes
Si tienes clase...
¡te llevamos al cine!
Regístrate y podrás ganar entradas
para ir al cine con tus amigos.
Participa escaneando el código QR
que aparece abajo.
5% de dto.
en tu próxima compra

5. MODES

ECB (Electronic code book)

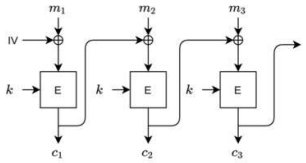


És el mode d'operació més senzill i consisteix en xifrar cada un dels blocs del missatge en clar m de manera individual, fent servir la mateixa clau. Així, s'obtenen els blocs xifrats c, que es concatenen per formar el text xifrat c.

Propietats principals:

1. Els blocs de text en clar idèntics tenen el mateix resultat de xifrat (amb la mateixa clau).
2. Cada bloc es xifra de manera independent als altres blocs.
3. Permet l'accés aleatori al contingut, és a dir, és possible desxifrar un bloc i sense haver de desxifrar els anteriors.
4. Es errors no es propaguen: un error en un bloc afecta només a aquell bloc.
5. No es detecten reordenacions, insercions, eliminacions

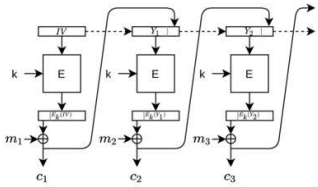
CBC (Cipher block chaining)



Consisteix en l'encadenament dels blocs per al xifratge, de manera que es crea una dependència del xifratge de cada bloc amb l'immediatament anterior. De nou, cada bloc es xifra amb la mateixa clau k, però el text que es xifra no és directament el bloc en clar, sinó el resultat d'una XOR entre el bloc en clar i el bloc xifrat anterior.

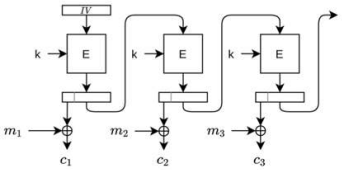
Oculta patrons del text en clar molt millor que ECB, es detecten reordenacions, eliminacions, insercions, el xifrat no es paral·lelitzable (el desxifrat si), i un error en un bloc es propaga per al següent bloc.

CFB (Cipher Feedback)



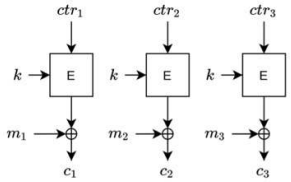
Utilitza indirectament el xifrador de bloc. Per això, la llargada dels blocs que s'han de xifrar no cal que sigui la mateixa que la dels blocs de xifrat, sinó que pot ser més petita. Si n = b, aleshores és CBC. Es pot fer servir per convertir un cirptosistema de bloc a un de flux (CFB-1 o 1-bit CFB). L'encriptació no és paral·lelitzable, mentre que la desencriptació si, i l'error afecta als següents b/n blocs.

OFB (Output feedback)



Utilitza el criptosistema de bloc com a generador pseudoaleatori. És un sistema molt semblant a l'anterior; l'única diferència que presenta és que el vector inicial es realimenta directament amb el resultat del xifratge de bloc abans de fer la suma bit a bit amb el bloc de text en clar, és a dir, no s'alimenta del que surt de l'emissor, sinó del que surt directament. Permet construir un xifrat en flux a partir d'un en bloc, el keystream es genera de forma independent al missatge (cleartext o ciphertext), els errors no es propaguen (però errors en IV afecten tot el xifrat/desxifrat), i no és paral·lelitzable, però es pot pre-generar al keystream.

CTR (Counter)



És similar a l'OFB, convertint també el criptosistema de bloc amb un xifrador de flux. La seqüència de xifrat es genera xifrant successius valors d'un comptador, que pot ser qualsevol funció que tingui un període gran. És paral·lelitzable (no requereix cap tipus de feedback), i els errors no es propaguen.

Ona Folch