1 Chacha20

Per expandir la clau de 256 bits que fa servir Chacha20 s'inicialitzen uns estats on l'única diferència és el valor del comptador. Tenint en compte que en moltes situacions és pot saber quins són el primers bytes en clar d'un fitxer xifrat, i per tant quins són els primers bits expandits de la clau, és important que a partir d'ells no és puguin inferir els següents bits expandits de la clau, o sigui que no hi ha una correlació senzilla entre els bits expandits a partir de diferents comptadors.

`Podeu fer servir qualsevol implementació del Chacha20 que trobeu.`

1.1 Propagació de petits canvis

Amb una clau K de 256 bits qualsevol feu una estadística dels bits que canvien a la sortida pels diferents valors del `Counter=2,3,...,4096` comparant-los amb `Counter=1`:

1. Dibuixeu un diagrama acumulat de freqüències del nombre total de bits que canvien[1] amb cada variació de `Counter`.
2. Dibuixeu un diagrama acumulat de freqüències de les posicions que canvien[2] amb cada variació de `Counter`.

1.2 Efectes de les funcions elementals

Repetiu l'apartat anterior però

1. Elimineu els QUARTERROUND dels Column rounds:

   ```
   QUARTERROUND(0, 4, 8, 12)
   QUARTERROUND(1, 5, 9, 13)
   QUARTERROUND(2, 6, 10, 14)
   QUARTERROUND(3, 7, 11, 15)
   ```

2. Elimineu els QUARTERROUND dels Diagonal rounds:

   ```
   QUARTERROUND(0, 5, 10, 15)
   QUARTERROUND(1, 6, 11, 12)
   QUARTERROUND(2, 7, 8, 13)
   QUARTERROUND(3, 4, 9, 14)
   ```

3. Elimineu els QUARTERROUND:

   ```
   QUARTERROUND(0, 4, 8, 12)
   QUARTERROUND(1, 6, 11, 12)
   ```

[1] A la base hi ha el nombre r de bits que han canviat i a l'eix vertical el nombre de vegades que r bits han canviat.

[2] A la base hi ha la posició i-èsima del bit que ha canviat i a l'eix vertical el nombre de vegades que el bit i-èsim ha canviat.

2 El cos finit $GF(2^8)$

Els elements d'aquest cos són els **bytes**. Els expressaren en forma binària, hexadecimal o polinòmica, segons convingui.

El byte $\mathtt{b_7b_6b_5b_4b_3b_2b_1b_0}$ serà el polinomi $b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$.

Per exemple, `01010111=0x57` serà $x^6 + x^4 + x^2 + x + 1$.

Suma

La suma de dos elements del cos és la suma de polinomis binaris. Per exemple, `01010111+10000011` serà

$$(x^6 + x^4 + x^2 + x + 1) + (x^7 + x + 1) = x^7 + x^6 + x^4 + x^2 = \mathtt{11010100}$$

Es correspon amb la operació XOR, que es denotarà $\oplus$. L'element neutre de la suma és `00000000=0x00`.

Multiplicació

Per fer el producte de dos elements del cos cal fer el producte de polinomis binaris i després prendre el residu de la divisió per $\mathbf{m = x^8 + x^4 + x^3 + x^2 + 1}$[3]. Per exemple,

$$\begin{aligned}(x^6 + x^4 + x^2 + x + 1)(x^7 + x + 1) &= x^{13} + x^{11} + x^9 + x^8 + x^7 + \\ &\quad x^7 + x^5 + x^3 + x^2 + x + \\ &\quad x^6 + x^4 + x^2 + x + 1 \\ &= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1\end{aligned}$$

$$x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \pmod{x^8 + x^4 + x^3 + x^2 + 1} = x^5 + x^4 + 1.$$

L'element neutre de la multiplicació és `00000001=0x01`.

A $GF(2^8)$, tot element diferent del `0x00` té invers multiplicatiu. L'invers del polinomi a és l'únic polinomi b tal que

$$a\,b = 1 \mod m.$$

Es pot calcular usant l'algorisme d'Euclides estès.

També podem escriure els elements diferents del `0x00` com a potència d'un generador. Per exemple, si $g = x = \mathtt{00000010} = \mathtt{0x02}$,

llavors

$$GF(2^8) = \{g, g^2, \ldots, g^{254}, g^{255}(= g^0 = 1)\} \cup \{0\}$$

El producte de dos elements $a = g^i$ i $b = g^j$, diferents de `0x00`, és $ab = g^i g^j = g^{i+j}$, i l'invers de a és $a^{-1} = (g^i)^{-1} = g^{-i} = g^{255-i}$. En aquest cas, la multiplicació i el càlcul de l'invers es redueixen a la cerca en una taula de 255 elements.

[3] El polinomi que fa servir l'AES es $\mathbf{x^8 + x^4 + x^3 + x + 1}$.

Definiu en **Python 3** les funcions (*El polinomi que heu de fer servir per definir les operaciones en el cos és* $\mathbf{m = x^8 + x^4 + x^3 + x^2 + 1}$):

i) `GF_product_p(a, b)`

entrada: `a` i `b` elements del cos representat per enters entre 0 i 255;
sortida: un element del cos representat per un enter entre 0 i 255 que és el producte en el cos de `a` i `b` fent servir la definició en termes de polinomis.

ii) `GF_es_generador(a)`

entrada: `a` element del cos representat per un enter entre 0 i 255;
sortida: `True` si `a` és generador del cos, `False` si no ho és.

iii) `GF_tables()`

entrada:
sortida: dues taules (*exponencial* i *logaritme*), una que a la posició i tingui $a = g^i$ i una altra que a la posició a tingui i tal que $a = g^i$. (g generador del cos finit del cos representat pel menor enter entre 0 i 255.)

iv) `GF_product_t(a, b)`

entrada: `a` i `b` elements del cos representat per enters entre 0 i 255;
sortida: un element del cos representat per un enter entre 0 i 255 que és el producte en el cos de `a` i `b` fent servir la les taules *exponencial* i *logaritme*.

v) `GF_invers(a)`

entrada: `a` element del cos representat per un enter entre 0 i 255;
sortida: `0` si `a=0x00`, invers d'`a` en el cos si `a!=0x00` representat per un enter entre 1 i 255.

Feu taules comparatives dels temps d'execució fent servir les diferents funcions:

- `GF_product_p` *vs* `GF_product_t`,
- `GF_product_p(a,0x02)` *vs* `GF_product_t(a,0x02)`,
- `GF_product_p(a,0x03)` *vs* `GF_product_t(a,0x03)`,
- `GF_product_p(a,0x09)` *vs* `GF_product_t(a,0x09)`,
- `GF_product_p(a,0x0B)` *vs* `GF_product_t(a,0x0B)`,
- `GF_product_p(a,0x0D)` *vs* `GF_product_t(a,0x0D)`,
- `GF_product_p(a,0x0E)` *vs* `GF_product_t(a,0x0E)`,

Atenció! És considerarà un error greu si:

- `GF_product_p(a, b)`!=`GF_product_t(a, b)` per algun parell (`a`, `b`),
- `GF_product_p(a, b)`!=`GF_product_p(b, a)` per algun parell (`a`, `b`),
- `GF_product_p(a, GF_invers(a))`!=1 per `a`!=0.

3 Criptografia de clau secreta

`Podeu fer servir qualsevol implementació de l'AES que trobeu.`

1. Desxifreu el primer fitxer que heu rebut.
2. Desxifreu el segon fitxer que heu rebut i que ha sigut xifrat fent servir AES-128 (clau 128 bits) amb *padding* PKCS7 i mode CBC.

   S'ha volgut que la clau secreta `K` i el vector inicial `IV` s'obtingués a partir d'informació aportada per 8 participants de forma sigui necessari el concurs de tots per recuperar `K` i `IV`:

   (a) Cada participant ha escollit 2 caràcters ASCII (8 bits) d'entre el conjunt

   `abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789`

   per exemple `a` i `y`, i ha format la seva clau $K_i =$ `aaaaaaaayyyyyyyy`

   (b) S'ha calculat `preMasterKey`$=K_1 \oplus K_2 \oplus \cdots \oplus K_8$ i `H=sha256(preMasterKey)`.

   (c) La clau secreta `K` està formada pel primers 128 bits d'`H` i el vector inicial `IV` pels darrers 128 bits d'`H`.

Referències

- RFC 8439: ChaCha20 and Poly1305 for IETF Protocols `https://tools.ietf.org/html/rfc8439`
- Federal Information Processing Standards Publication (FIPS) 197: Advanced Encryption Standard (AES) `http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf`
- NIST Special Publication 800-38A: Recommendation for Block Cipher Modes of Operation. `http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38A.pdf`
- Padding PKCS7: section 6.3 RFC 5652. `http://tools.ietf.org/html/rfc5652#section-6.3`

Per llegir

- Bruce Schneier *NSA and Bush's Illegal Eavesdropping.*
- Schmid, Gerhard (11 July 2001). *On the existence of a global system for the interception of private and commercial communications (ECHELON interception system), (2001/2098(INI))*. European Parliament: Temporary Committee on the ECHELON Interception System.