1 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 $\mathbf{m = x^8 + x^4 + x^3 + x + 1}$. Per exemple,

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

$$x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \pmod{x^8 + x^4 + x^3 + x + 1} = x^7 + x^6 + 1.$$

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 $g = x + 1 = \mathtt{00000011} = \mathtt{0x03}$, 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.

Definiu en **Python 3** les funcions:

1. `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.

   `GF_tables()`

   entrada:
   sortida: dues taules (*exponencial* i *logaritme*), una que a la posició i tingui $a = g^i$ ($g =$ `0x03`) i una altra que a la posició a tingui i tal que $a = g^i$.

   `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*.

2. `GF_generador()` que doni tots el generadors del cos finit;

3. `GF_invers(a)`

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

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)`,

2 Advanced Encryption Standard (AES)

Podeu fer servir qualsevol implementació que trobeu.

2.1 Efectes de les funcions elementals

1. Canviem la funció **ByteSub** per la identitat, i.e. **ByteSub(x)=x**.

   Sigui M_i igual a M excepte en el bit i; M_j igual a M excepte en el bit j; M_{ij} és igual a M excepte en els bits i, j; C_i el resultat de xifrar M_i amb la clau K; C_j el resultat de xifrar M_j amb la clau K; C_{ij} el resultat de xifrar M_{ij} amb la clau K:

   $M=$

15	33	7e	b3
97	1c	6d	ea
c4	c2	1b	3b
ef	8b	2e	95

   $M_i=$

15	33	7e	b3
97	1c	6d	ea
c4	**82**	1b	3b
ef	8b	2e	95

   $M_j=$

15	33	7e	**93**
97	1c	6d	ea
c4	c2	1b	3b
ef	8b	2e	95

   $M_{ij}=$

15	33	7e	**93**
97	1c	6d	ea
c4	**82**	1b	3b
ef	8b	2e	95

   $C=$

ae	99	2e	5c
29	c0	ab	16
8d	74	01	a2
91	19	99	2e

   $C_i=$

ae	99	**6e**	5c
29	**00**	ab	16
0d	74	01	a2
91	19	99	**6e**

   $C_j=$

ae	99	2e	**5a**
29	c0	**a3**	16
8d	**7f**	01	a2
9e	19	99	26

   $C_{ij}=$

ae	99	**6e**	**5a**
29	**00**	**a3**	16
0d	**7f**	01	a2
9e	19	99	**6e**

   Feu un programa, **que es pugui compilar i executar als ordinadors de la FIB**, per comprovar que $C = C_i \oplus C_j \oplus C_{ij}$ per qualsevol i, j, i que això no pasa si agafen la funció **ByteSub** original:

   $C=$

2a	9a	7c	9c
56	9f	36	76
e1	34	6e	ec
4e	63	c8	60

   $C_i=$

67	84	1b	ac
22	43	bd	e7
73	52	ed	5c
82	ff	1d	b3

   $C_j=$

0e	95	9c	0d
ee	98	3f	f2
81	0a	b5	e2
2e	13	59	d4

   $C_{ij}=$

55	d1	61	74
ef	62	72	0e
bb	e1	ea	9d
d5	d0	b7	ea

2. Canviem la funció **ShiftRows** per la identitat. Quins efectes té aquest canvi al xifrar un bloc? (Xifreu diferents M i els corresponents M_i amb la mateixa clau K i compareu C amb C_i.)

3. Canviem la funció **MixColumns** per la identitat. Quins efectes té aquest canvi al xifrar un bloc? (Xifreu diferents M i els corresponents M_i amb la mateixa clau K i compareu C amb C_i.)

2.2 Propagació de petits canvis

Amb un missatge M de 128 bits i una clau K de 128 bits qualssevol feu una estadística dels bits que canvien a la sortida quan modifiqueu un bit de M:

1. histograma del nombre total de bits que canvien amb cada modificació,
2. histograma de les posicions que canvien amb cada modificació.

Feu el mateix si modifiqueu un bit de K.

3 Criptografia de clau secreta

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 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

- 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.