Iniciació a la Criptografia

Artur Travesa

(versió 2024-07)

Capítol 12. ElGamal

12.0. Introducció

El criptosistema ElGamal és basat en el mateix principi que el protocol Diffie-Hellman per a intercanvi de claus en canals públics; el va inventar Tahe ElGamal, que el va descriure en un article en 1984 (cf. **[D-H]**, **[ElGamal]**). Fins al febrer de 2024 (on ha deixat de ser estàndard, substituït per la versió de 2023) era a la base d'algun dels estàndards (DSA) de signatura digital (cf. **[DS-S**, cap. 4**]**).

La seguretat del criptosistema ElGamal es basa en la dificultat (conjectural) de calcular logaritmes discrets arbitraris. En efecte, donats un nombre primer p, un element g invertible mòdul p, i un nombre natural x, el càlcul de la potència, $h = g^x \pmod{p}$ es pot fer en temps polinòmic en la longitud en bits de p i de x; en canvi, no es coneix cap algoritme general tal que, donats p, g i h, trobi el nombre x en temps polinòmic en la longitud en bits de p. Igual com succeeix en el cas de la factorització, el millor algoritme que es coneix actualment és subexponencial.

Observació. I també com en el cas dels criptosistemes RSA, notem que parlem d'algoritmes *clàssics*, perquè actualment ja hi ha algoritmes quàntics polinòmics per a resoldre aquest problema.

A continuació proporcionem una descripció dels problemes de logaritme discret i, més avall, una descripció, i, simultàniament, un exemple, de criptosistema de tipus ElGamal.

12.1. Logaritmes discrets

En general, suposem que disposem d'un grup qualsevol, G, i d'un element $g \in G$ d'ordre finit, q. Escriurem multiplicativament l'operació del grup i ens fixarem en el subgrup de G generat per g (per tant, cíclic). Aquest subgrup, C_g, és format per les potències $h := g^x$, per a $1 \leq x \leq q$, ja que q és l'ordre de g; notem, també, que $g^0 = g^q = 1$ és l'element neutre del grup G i, per tant, també del grup C_g.

12.1.0. Definició

Amb les notacions anteriors, i per analogia amb els logaritmes reals o complexos, x s'anomena el *logaritme discret de h en base g*.

Notem que x és un nombre enter, que està definit mòdul l'ordre q de g en G; per tant, podem pensar x com un element de $\mathbb{Z}/q\mathbb{Z}$.

12.1.0.0. Obervació

Així, l'assignació $x \mapsto g^x$ defineix un isomorfisme de grups del grup additiu $\mathbb{Z}/q\mathbb{Z}$ en el grup multiplicactiu C_g: el podem anomenar l'exponencial de base g. L'isomorfisme invers s'anomena el *logaritme discret* en base g per al grup C_g o, també, per al grup G.

12.1.1. Exemple

Considerem un nombre natural primer p, i sigui $G = (\mathbb{Z}/p\mathbb{Z})^*$ el grup dels elements invertibles mòdul p. En G, podem considerar qualsevol element g com a base dels logaritmes discrets. Si anomenem q l'ordre de g, tenim que el grup C_g és format per les potències $h = g^x$, per a $1 \leq x \leq q$. També tenim, en virtut del petit teorema de Fermat, que q és un divisor de $p - 1$.

12.1.1.0. Observació

En general, donats p, g, i g^x, el càlcul de x no se sap fer en temps polinòmic respecte de la mida en bits de p (ni tan sols, conegut q). Això fa que l'exponencial $x \mapsto g^x$ es pugui considerar (conjecturalment) una funció *de sentit únic*.

Recordem que s'anomena funció de sentit únic una aplicació bijectiva tal que les imatges es poden calcular en temps polinòmic, però les antiimatges, no.

12.1.2. Un exemple (inservible criptogràficament)

Notem que si en lloc de considerar el grup multiplicatiu de les classes residuals invertibles mòdul n considerem el grup additiu de totes les classes residuals mòdul n, i si g és qualsevol classe, la potència g de més amunt es correspon ara amb el producte gx, de manera que el càlcul del logaritme discret es redueix a la resolució de la congruència lineal $gx \equiv h \pmod{n}$; i aquesta es pot resoldre (per exemple, mitjançant l'algoritme d'Euclides) en temps polinòmic en la longitud en bits de n.

Veiem, doncs, que hi ha grups cíclics per als quals el problema del logaritme discret és senzill de resoldre i, per tant, els logaritmes discrets en aquests grups no semblen útils criptogràficament.

12.2. El criptosistema ElGamal

12.2.0. El grup cíclic

De manera similar a com en els criptosistemes de tipus RSA la base del mètode és el càlcul de potències mòdul un nombre enter N, així també en el criptosistema ElGamal la base és aquest mateix tipus de càlcul.

Cada usuari U del criptosistema tria un nombre primer senar p i un nombre enter g no divisible per p i tal que l'ordre multiplicatiu de g en $(\mathbb{Z}/p\mathbb{Z})^*$ sigui un nombre primer senar q. (Moltes vegades, tots els usuaris U del criptosistema utilitzen el mateixos valors de p i de g i, per tant, tots coneixen també el valor de q.)

En particular, es té que $p-1=2kq$, on k és un nombre enter. D'aquesta manera, el subgrup G de $(\mathbb{Z}/p\mathbb{Z})^*$ generat per g és cíclic, d'ordre primer, q, i és format pels elements g^x, $1 \leq x \leq q$, de $(\mathbb{Z}/p\mathbb{Z})^*$.

12.2.1. Observació

A fi que aquesta tria sigui útil per al criptosistema ElGamal cal que el problema del logaritme discret en el grup G sigui computacionalment intractable, encara que aquesta condició no és l'única que cal per a G. A la pràctica, i per a satisfer la necessitat d'aquesta condició (i d'altres), no es fa servir tot el grup $(\mathbb{Z}/p\mathbb{Z})^*$ i una arrel primitiva mòdul p, sinó un subgrup G,generat per un element g, potència adequada d'una arrel primitiva.

12.2.2. La mida del grup

En l'actualitat (juliol de 2024), sembla prou segur utilitzar nombres primers p tals que l'ordre de G, posem q, sigui un nombre d'un miler de bits, aproximadament. En el nostre exemple, prendrem com a p un nombre primer de 1024 bits i tal que $p-1$ sigui divisible per un primer q de 1000 bits. D'aquesta manera, el nombre enter $k := \dfrac{p-1}{2q}$ serà un nombre de, com a màxim, 24 bits, o sigui, relativament petit.

12.2.3. Càlcul del grup

El càlcul de nombres p, q i k en aquestes condicions no és especialment complicat, i es pot fer a partir d'algoritmes probabilístics. Per exemple, es pot consultar la referència **[Tr-PrF]**.

12.2.4. Càlcul de l'ordre

D'altra banda, la tria d'un element de $(\mathbb{Z}/p\mathbb{Z})^*$ d'ordre q no presenta dificultats. En efecte, si g_0 és una arrel primitiva mòdul p, l'element g_0^{2k} és d'ordre q, i tots els elements de $(\mathbb{Z}/p\mathbb{Z})^*$ d'ordre q són de la forma g_0^{2k}, per a alguna arrel primitiva mòdul p, g_0. Per tant, és suficient calcular una arrel primitiva mòdul p i elevar aquesta a la potència $2k$, mòdul p.

12.2.5. Exercici

(a) Es demana calcular un nombre primer p de 1024 bits de manera que $p-1$ sigui el producte d'un nombre primer q de 1000 bits per un nombre enter parell $2\ k$.

(b) Es demana calcular un element g de $(\mathbb{Z}/p\mathbb{Z})^*$ d'ordre q, per als valors p, q de l'apartat (a).

12.3. Les claus

Quan l'usuari U disposa de p, q, k i g, tria a l'atzar un nombre enter x, $1 \leq x \leq q-1$, i calcula $h := g^x \pmod{p}$. D'aquesta manera, h és un element de G.

12.3.0. Definició

La clau pública de U és (p, q, g, h); la clau privada de U és (p, q, g, x). Notem que l'única diferència és en h i x, mentre que els paràmetres p, q i g són comuns; en particular, tots els usuaris del criptosistema poden usar els mateixos p, q, i g.

12.3.1. No simetria de les claus

Els nombres h i x de les claus juguen un paper molt diferent. Així com h és un element de G, i actua com a tal en el procés de xifratge, x és un nombre natural, i actua com a tal en el procés de desxifratge. En conseqüència, x i h no són intercanviables, en contraposició a les claus pública i privada del criptosistema RSA, que sí que són intercanviables.

12.3.2. Observació

Cal notar que calcular x a partir de p, q, g i h vol dir calcular el logaritme discret mòdul p, de base g, per a l'element h; per tant, si no podem resoldre aquest problema de logaritme discret, no sabem calcular la clau privada de U a partir de la seva clau pública.

12.3.3. Exercici

Es demana crear, a partir dels nombres p, q i g de l'exercici **12.2.6**, una parella de claus, la pública **[p, q, g, h]**, i la privada corresponent, **[p, q, g, x]**, per al criptosistema ElGamal. Les anomenarem **ClauElGamalPublica** i **ClauElGamalPrivada**.

12.4. Xifratge

Les unitats de missatge que es poden xifrar amb el criptosistema ElGamal són els elements de $\mathbb{Z}/p\mathbb{Z}$. En particular, el conjunt d'unitats de missatge vàlides és diferent per a cada usuari si cadascun treballa en un anell diferent, però el mateix si tots els usuaris treballen amb el mateix valor de p; és per això que és còmode usar el mateix valor de p per a tots els usuaris del criptosistema.

Notem que el conjunt de missatges que poden ser xifrats és més gran que el grup que fem servir per a xifrar; de fet, el grup G és un subgrup de $(\mathbb{Z}/p\mathbb{Z})^*$ que, alhora, és un subconjunt de $\mathbb{Z}/p\mathbb{Z}$ (i, encara, de tot en prenem representants en $\mathbb{Z}$).

Suposem que la unitat de missatge que es vol xifrar és un element m de $\mathbb{Z}/p\mathbb{Z}$, on p és el nombre primer de la clau de U (a la pràctica, un nombre enter m tal que $0 \leq m \leq p-1$).

L'usuari V que vol enviar el missatge m a U, tria a l'atzar un nombre enter y de l'interval $1 \leq y \leq q-1$ i calcula els dos nombres $c_1 :\equiv g^y \pmod{p}$ i $c_2 :\equiv m \cdot h^y \pmod{p}$,

amb $1 \leq c_1 \leq p-1, 0 \leq c_2 \leq p-1$. El missatge xifrat associat a m és la parella de nombres naturals (c_1, c_2).

12.4.0. La mida del xifrat

Cal notar que, en el nostre exemple, tant c_1 com c_2 seran nombres de, com a màxim, 1024 bits, perquè p és de 1024 bits. Per tant, la mida del missatge xifrat és el doble de la mida de p; en el nostre cas, 2048 bits.

12.5. Desxifratge

Per a desxifrar el missatge, l'usuari U calcula el producte $c_2 c_1^{-x} \pmod{p}$ (pot calcular-lo perquè coneix el valor secret x), i això produeix el missatge m.

12.5.0. Demostració

Es té que $c_2 c_1^{-x} \equiv m h^y (g^y)^{-x} \equiv m (g^x)^y (g^y)^{-x} \equiv m \pmod{p}$. □

12.5.1. Observació

Cal notar que les potències de g són elements de $(\mathbb{Z}/p\mathbb{Z})^*$, de manera que la multiplicació per ells (i pels seus inversos) és possible a tot $\mathbb{Z}/p\mathbb{Z}$, i no cal restringir-se a G. Per això els missatges no cal que estiguin en G.

12.6. Exemple

La funció següent pren com a entrades una llista d'unitats de missatge i una clau pública ElGamal i produeix una llista d'unitats de missatge xifrades amb aquesta clau pública.

In [1]:
```
def ElGamalX0(mc,claupublica):
    [p,q,g,h]=claupublica
    l=len(mc)
    return [[Mod(g,p)^(y:=ZZ.random_element(1,q)),mod(mc[i]*Mod(h
```

12.6.0. Una parella de claus

Observem que, un cop haguem resolt l'exercici 12.3.3, ja disposarem d'una parella de claus pública i privada per al criptosistema ElGamal.

Suposem que tenim la clau pública en una llista anomenada ClauElGamalPublica0 i que la clau privada és en una llista anomenada ClauElGamalPrivada0.

(Per si encara no s'ha resolt l'exercici, aquí disposem d'una parella de claus pública i privada de 1024 bits per al criptosistema ElGamal.)

In [2]:
```
ClauElGamalPublica0=[163373482070616520482398310743692947258024181
```

In [3]:
```
ClauElGamalPrivada0=[163373482070616520482398310743692947258024181
```

Seleccionem el valor de p per a poder-hi treballar còmodament (és un component de la clau pública).

In [4]:
```
pe0 = ClauElGamalPublica0[0]
```

12.6.1. Un missatge

Ara ens inventem un missatge (de cinc unitats de missatge) aleatori.

In [5]:
```
m1=[ZZ.random_element(0,pe0) for i in range(5)]
```

In [6]:
```
print(m1)
```

```
[919784172331792377241023093253301109200621058871588200674857929729
173727165535431579233240281233440794883862574744464919675305737122 2
337841943008912367156306216232822653971812807199028019387441531463 8
355812682027013176394061049072617970719034610703583749981559218749 8
968911035034198590951300691795434226986 26, 988241815387677510581361
670955992519407026992197640187304511878137225107458831202441561592 6
169967433662843973264858655037270510077972339291983229407410896711 1
666238410096305065107954563210794039476945136660797406945998818017 3
890721872288023012577512921783785788863884846660749335517582669043 1
0514850420940949, 764647349917920112276090664713953236038453876512 2
031542587429069559131524607471374597878535251933489164595444479052 7
883820619162190967821363573686739815038868306839405458559139908400 5
427840885059743026263170932459835784301282964508438580719401177452 7
588175444371520187971029408845902129783978285374215295509 6, 1075120
020099457469151185888442295413254628969222583664535361318419661087 5
246873042251072086705713315186919762217111652737492266329333210750 2
004634137393596638598773879203369072286113655638887514884379330043 8
397956227041641410219417179855156030009806040791746932658948728810 2
707322132962932162305568530152796 4, 146288282239250419297959273379 0
474384372478524060464753602391736302511429583958377911195779900375 7
758541054714865371243103298386204500769308026682347357139736944963 0
982064317331770948878558254164653121951450910714996289014534472273 6
764576918664295973121473771547837778481060699773254701381934016696 3
1767706835]
```

I el xifrem amb la clau pública.

In [7]:
```
x1=ElGamalX0(m1,ClauElGamalPublica0)
```

12.6.2. Exercici (una funció de desxifratge)

Es demana definir una funció ElGamalDX0(mx,clauprivada) per a desxifrar missatges xifrats amb la funció ElGamalX0(mc,claupublica). Notem que cal produir, a la sortida, el missatge codificat en forma d'una llista d'elements de $\mathbb{Z}/p\mathbb{Z}$, on p és el nombre primer de la clau privada.

Per a comprovar que funciona bé, es pot provar de desxifrar el missatge x1.

In [8]:
```
def ElGamalDX0(mx,clauprivada):
    [p,q,g,x]=clauprivada
    l=len(mx)
    return [Mod(mx[i][1],p) * Mod(mx[i][0],p)^(-x) for i in rang
```

In [9]:
```
dx1 = ElGamalDX0(x1,ClauElGamalPrivada0)
```

In [10]:
```
dx1==m1
```

Out[10]: True

Fi del capítol 12