Ús de corbes el·líptiques en la criptografia i SAGE

Adrià Alcalá Mena, David Sánchez Charles

Assignatura de Codificació i Criptografia de 4t de la Llicenciatura de Matemàtiques

***Abstract*—El desenvolupament d'internet ha fet que les nostres comunicacions es realitzin mitjançant aquesta xarxa. La criptografia ens proporciona eines perquè la nostra informació es pugui transmetre de manera segura per aquesta xarxa pública. En aquest treball farem una implamentació utilitzant SAGE per a utilitzar corbes el·líptiques dins la criptografia de clau pública.**

***Index Terms*—Corbes el·líptiques, elGamal, criptografia, SAGE**

I. Introducció

CADA vegada és més habitual la compra d'articles per internet. Perque aquestes compres siguin exitoses s'haurien de complir dos requisits bàsics: secret i autenticació. La transmissió de la informació s'ha de fer de manera secreta; no es vol que una altra persona pugui, per exemple, conseguir les nostres dades bancàries. També, qui rep la informació ha de poder autenticar l'autoria de l'emissor; en el nostre exemple la tenda virtual ha d'estar segura que som nosaltres qui hem fet la compra.

La criptografia dóna solució a aquests dos problemes: Els algoritmes de xifrat i desxifrat ens asseguren que la transmissió de les dades per internet es fa de manera segura (secret). D'altra banda, també ens propociona algoritmes de *firma* que asseguren al receptor qui és qui ha enviat el missatge (autenticació).

La criptografia moderna es basa en l'ús d'unes aplicacions (generalment tothom coneix com són aquestes aplicacions) que tenen per paràmetre el missatge que es vol xifrar o desxifrar i una clau. L'èxit d'aquests mètodes rau en què, sense la clau, és computacionalment molt difícil xifrar i desxifrar els missatges de la mateixa manera que l'emissor original.

Dins la criptografia moderna ens volem centrar en els criptosistemes de clau pública (proposats per Diffie Hellman en 1976). Aquests criptosistemes tenen 2 claus: Una de privada, que només coneix el propi usuari; i una de pública, que coneixen tots els possibles usuaris de la xarxa. Els usuaris que es vulguin comunicar amb un usuari determinat, han d'agafar la seva clau pública i fer ús de l'algoritme de xifrat amb aquesta clau. Quan l'usuari rep el missatge xifrat ha d'agafar la seva clau privada i fer ús de l'algoritme de desxifrat per retrobar el missatge original.

En aquest article farem un petit estudi d'un algoritme que fa ús de les corbes el·liptiques per al xifrat i desxifrat de missatges. Hem programat aquest algoritme sobre SAGE; una eina matemàtica de programari lliure que ens proporciona l'accés a diverses llibreries ja implementades.

Juny 30, 2011

II. RSA

Al febrer de 1978 Ron Rivest, Adi Shamir i Leonard Adleman proposen un criptosistema de xifrat de clau pública: RSA, el qual consisteix en associar als missatges originals un valor numèric i aleshores xifrar el missatge per blocs de la mateixa longitud i amb un valor numèric comprès entre un cert rang.

Suposem $m \in [2, n-1]$, l'algoritme de xifrat es redueix al càlcul d'una exponencial on la clau és el parell de nombres (e, n):

$$c = E_{(e,n)}(m) = m^e \pmod{n}$$

L'algoritme de desxifrat per poder obtenir m a partir de c consisteix també en una exponenciació, essent la clau ara un altre parell de nombres (d, n).

$$m = D_{(d,n)} = c^d \pmod{n}$$

[1] Donat $\varphi(n)$ és fàcil generar el parell de nombres e i d tal que $e \cdot d = 1 \pmod{\varphi(n)}$. No obstant això, calcular d coneixent e i n és molt complicat computacionalment sense coneixer $\phi(n)$. La fortalesa d'aquest criptosistema va lligada a la dificultat de la factorització de n.

Per tant, si tenim una manera simple d'obtenir $\varphi(n)$, obtendrem un bon esquema de xifrat i desxifrat. Rivest, Shamir i Adleman sugereixen l'algoritme d'obtenció de les claus que es pot veure a la Fig 1.

Require: Dos nombres primers p, q.
Ensure: La clau pública i privada de l'usuari.

1: $n = p \cdot q$.
2: Es calcula $\phi(n) = (p-1)(q-1)$.
3: S'agafa un nombre sencer $0 < e < \phi(n)$ tal que $mcd(e, \phi(n)) = 1$.
4: Es calcula $d = e^{-1} \pmod{\phi(n)}$.
5: **return** Clau pública (n, e), Clau privada (d)

Fig. 1. algoritme d'obtenció de les claus

Recordem que quan calculam la d feim l'invers dins de l'anell $\mathbb{Z}_{\phi(n)}$ i aquest invers sempre existeix ja que $mcd(e, \phi) = 1$.

[1] El teorema d'Euler assegura que $D_{(d,n)}(E_{(e,n)}(m)) = m$ si $e \cdot d = 1 \mod (\phi(n))$

A. Exemple

```
sage:p,q,e,e2=23,19,17,29
sage:CLAUA=Claves(p,q,e)
     CLAUB=Claves(p,q,e2)
sage:CLAUA
[23, 19, 437, 396, 17, 233]
sage:CLAUB
[23, 19, 437, 396, 29, 41]
sage:missatge="RSA vs CE"
sage:send=EnviarMensaje2(missatge,CLAUA,CLAUB)
R -> 82 -> 54 -> 104
S -> 83 -> 11 -> 7
A -> 65 -> 145 -> 350
  -> 32 -> 288 -> 200
v -> 118 -> 423 -> 234
s -> 115 -> 115 -> 115
  -> 32 -> 288 -> 200
C -> 67 -> 249 -> 15
E -> 69 -> 46 -> 69
sage:RecibirMensaje2(send,CLAUA,CLAUB)
104 -> 54 -> 82 -> R
7 -> 11 -> 83 -> S
350 -> 145 -> 65 -> A
200 -> 288 -> 32 ->
234 -> 423 -> 118 -> v
115 -> 115 -> 115 -> s
200 -> 288 -> 32 ->
15 -> 249 -> 67 -> C
69 -> 46 -> 69 -> E
'RSA vs CE'
```

III. Corbes el·líptiques

Una corba el·líptica sobre nombres reals es defineix com el conjunt de punts (x, y) que satisfan l'equació de Weierstrass simplificada

$$y^2 = x^3 + ax + b$$

Les corbes el·líptiques sobre cossos finits $\mathbb{Z}_p$, amb un valor de p gran, ofereixen una alternativa en la criptografia de clau pública, com veurem posteriorment.

Si es compleix que $4a^3 + 27b^2 \neq 0$, la corba no té arrels repetides i podrem formar un grup additiu que a més tindrà un producte escalar, $(G, +, *)$, on G és el conjunt de punts de la corba, l'operació + és la suma de punts (que definirem més endavant) i * és el producte escalar. També afegirem un punt especial 0, que anomenarem *infinit*, que serà el neutre del grup.

A. Aritmètica en corbes el·líptiques dins el cos del nombres reals

1) Suma de punts d'una corba el·líptica: Donats dos punts de la corba P i Q, podem traçar la recta que passa per ells. Si aquesta recta talla la corba en un tercer punt, el reflectirem a través de l'eix de les x, i dóna lloc a un nou punt R. Direm que R és la suma de P i Q. En el cas que la recta que passa per P i Q no talli la corba en un tercer punt poden passar dues coses:

1) La recta és tangent a la corba en un dels dos punts. Amb la qual cosa la suma seria el simètric del punt on és tangent.
2) La recta no és tangent a la corba en cap punt. En aquest cas $P + Q = 0$

En el cas en que $P = Q$, es fa la mateixa construcció però amb la recta tangent a la corba que pasa per P.

L'explicació de l'operació suma es pot veure més fàcilment a la Fig 2

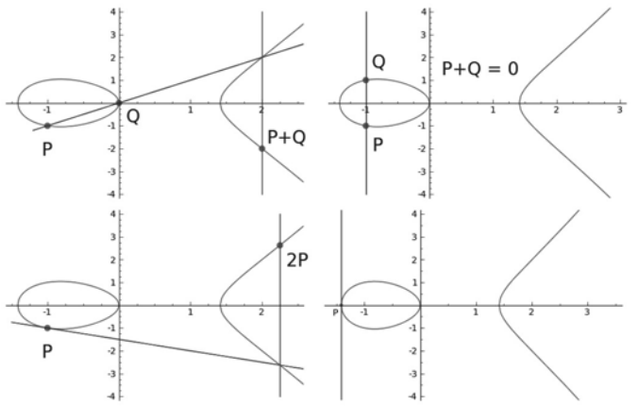

Fig. 2. Exemple de suma de punts en la corba el·líptica $y^2 = x^3 - 2x$

2) Punt invers dins una corba el·líptica: Podem definir el punt invers, o simètric, de $P = (x, y)$, com la seva reflexió sobre l'eix x, és a dir:

$$-P = (x, -y)$$

Aquesta definició és consistent, ja que si un punt P és de la corba $-P$ també ho és, ja que tota corba el·líptica és simètrica respecte l'eix x, si $y^2 = x^3 + ax + b$ aleshores $(-y)^2 = x^3 + ax + b$.

3) Producte d'un punt per un escalar: De la mateixa manera que podem sumar dos punts, el podem sumar amb ell mateix (Fig 2) i de manera recursiva podem definir que:

$$k \cdot P = ((k-1) \cdot P) + P$$

Així com les definicions de suma de punts i de punt invers tenien una explicació geomètrica, en aquest cas no.

4) Aritmètica dins corbes el·líptiques amb cossos finits: De la mateixa manera que hem definit les operacions sobre corbes el·líptiques en els reals, les podem definir sobre un $\mathbb{Z}_p$, i com era d'esperar, coincideixen amb el cas real però fent la reducció mòdul p. En aquest cas les definicions perden l'explicació geomètrica, ja que en un $\mathbb{Z}_p$ una corba el·líptica és un conjunt de punts sense cap continuïtat visual (veure Figura 8).

B. Algoritme ElGamal

En primer lloc necessitem obtenir les claus, per això hem de construir la clau pública i privada amb l'algoritme que es pot veure a la Fig. 3. Un cop tenim les claus podem xifrar

Require: Una corba el·líptica CE i un punt P.
Ensure: La clau pública i privada de l'usuari.

1: Calculam l'ordre r de la corba (el nombre de punts de la corba el·líptica).
2: Elegim un enter $d \in [2, r-2]$.
3: $clau = d * P$
4: **return** Clau pública (P,clau, r), Clau privada (d,P,r)

Fig. 3. algoritme d'obtenció de les claus

Require: Una corba el·líptica CE d'ordre r, la clau pública $clau$ amb el punt P i el punt que es vol enviar M.
Ensure: El punt M xifrat.

1: Elegim un enter $k \in [2, r-2]$.
2: $C1 = k * P$.
3: $C2 = M + k * clau$
4: **return** $(C1, C2)$

Fig. 4. algoritme de xifrat

Require: Una corba el·líptica CE, la clau privada d i el missatge rebut $(C1, C2)$
Ensure: El missatge $(C1, C2)$ desxifrat.

1: $M = C2 - d * C1$
2: **return** M

Fig. 5. algoritme de desxifrat

amb l'algoritme que es pot veure en la Fig. 4 i desxifrar el missatge rebut amb l'algoritme que es pot veure en la Fig. 5.

L'algoritme de firma d'un missatge M és pot trobar en la Fig. 6, per altra part per a comprovar l'autenticitat de l'emissor farem ús de l'algoritme que es pot veure a la Fig. 7. Aquests algoritmes utilitzen un tipus d'aplicacions anomenades funcions de hash[2].

Require: Una corba el·líptica CE i el seu ordre r, la clau privada (d, P) i un hash del missatge $H(M)$.
Ensure: La firma (R, S) del missatge M.

1: $k \in [2, r-2]$
2: x, y = components de $k * P$
3: $R = x \pmod r$
4: $S = k^{-1}(H(M) + R \cdot d) \pmod r$
5: **return** (R, S)

Fig. 6. algoritme de firma

Require: Una corba el·líptica CE i el seu ordre r, la clau pública (P, Q), el hash M del missatge rebut i la seva firma (R, S).
Ensure: L'acceptació, o no, de la firma.

1: $w = S^{-1} \pmod r$
2: $u_1 = M \cdot w$, $u_2 = R \cdot w$
3: x, y = components de $u_1 * P + u_2 * Q$
4: **return** És $x = R \pmod r$

Fig. 7. algoritme de verificació de la firma

C. Exemple

Comencem amb un primer exemple senzill dins $\mathbb{Z}_{23}$. La corba el·líptica amb la que farem feina és

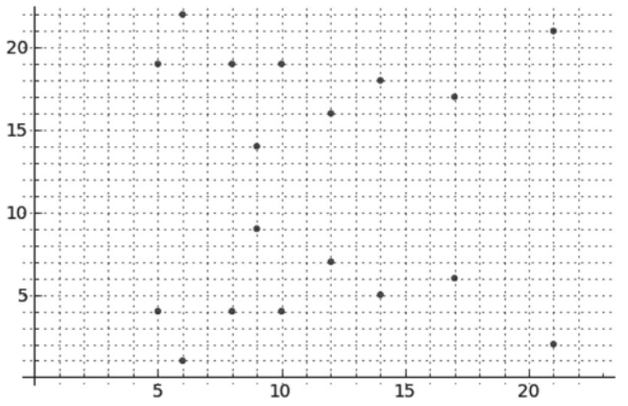

Fig. 8. Gràfica de la corba el·líptica $y^2 = x^3 + 9x + 7$ dins $\mathbb{Z}_{23}$

```
sage:E = CorbaElliptica(y**2==x**3+9*x+7,23)
```

Comprovem que l'ordre sigui primer

```
sage: E.order()
19
```

Amb el SAGE també la podem dibuixar amb la instrucció `E.plot()` que ens retorna la figura que es pot veure a Fig 8.

Definim tres punts i vegem com podem sumar punts amb SAGE:

```
sage:P,Q,M=E([8,4,1]),E([9,9,1]),E([14,5,1])
sage:(P+Q).xy()
(8,19)
sage:(5*P).xy()
(6,22)
```

La instrucció `.xy()` és simplement per a veure les coordenades del punt.

Ara farem ús de la llibreria *ElGamalArt*[3] per a xifrar el missatge M. Primer hem de crear les claus dels dos usuaris que es volen comunicar.

```
sage: publicaP,privadaP=Clau(E,P)
sage: publicaP
((8 : 4 : 1), (17 : 6 : 1), 19)
sage: privadaP
(4, (8 : 4 : 1), 19)
sage: publicaQ,privadaQ=Clau(E,Q)
sage: publicaQ,privadaQ
(((9 : 9 : 1), (17 : 6 : 1), 19), (17, (9 : 9 : 1), 19))
```

Cada usuari utilitza un punt diferent, d'una corba el·liptica comuna, per a obtenir el seu parell de claus. La clau pública està formada per un punt R (en el nostre exemple els punts P i Q), un múltiple seu $d \cdot R$ i, per qüestions d'optimització, l'ordre de la corba el·líptica. D'altra banda, la clau privada està formada per l'escalar d i, per comoditat, els punts R i l'ordre de la corba el·líptica.

El primer usuari vol enviar el missatge M, per a xifrar el missatge ha d'utilitzar la clau pública del receptor

```
sage:send=Xifrar(E,publicaQ,M)
sage:send
((6 : 22 : 1), (10 : 19 : 1))
```

Per tant l'emissor enviaria `send` al receptor. I, en el cas que el volgués firmar, el podria enviar amb la firma del nombre

[2]Una funció de hash transforma un objecte en una cadena de bits de longitut constant, pero no es coneix cap manera eficient d'obtenir, si existeix, l'invers d'aquest tipus aplicacions.

[3]Aquesta llibreria està feta per nosaltres, si a qualqu li interesa es pot posar en contacte amb nosaltres mitjançant els correus dscharles@gmail.com o adria.alcala@gmail.com

19^4

```
sage:firma=Firmar(E,privadaP,19)
sage:firma
(10,3)
```

Per tant l'emissor enviaria `send` juntament amb `firma`.

Ara, quan el receptor rebés el missatge, primer desxifra el missatge per a obtenir el missatge original

```
sage:Desxifrar(E,privadaQ,send)
(14 : 5 : 1)
```

i si vol verificar l'autoria del missatge, ha de calcular el hash del missatge desxifrat (en aquest cas, $19 = 14 + 5$) i fer-ne us de l'algoritme d'autenticació

```
sage:VerificarFirma(E,publicaP,19,firma)
True
```

IV. Comparació de criptosistemes

Anem a comparar ara la velocitat del criptosistema RSA amb el de corbes el·líptiques utilitzant ElGamal. Per a comparar la velocitat no podem agafar tamanys de claus iguals, si no que han de ser proporcionals a la dificultat de rompre el criptosistema, per exemple una clau de 512 bits a RSA és equivalentment segura a una de 106 bits a corbes el·líptiques. Taules comparatives de tamanys de claus es poden trobar a [1] i en podem veure una a la Fig. 9

A. 1024 bits RSA vs 160 bits CE

Vegem primer el que tarda en xifrar un punt qualssevol amb una corba amb un ordre de 160 bits.

```
Xifrar: 5 loops, best of 3: 60.1 ms per loop
Desxifrar: 25 loops, best of 3: 29.5 ms per loop
```

I ara vegem el que tarda en xifrar un nombre sencer qualssevol amb RSA amb una clau de 1024 bits.

```
Xifrar: 125 loops, best of 3: 2.25 ms per loop
Desxifrar: 125 loops, best of 3: 2.23 ms per loop
```

B. 2048 bits RSA vs 192 bits CE

Vegem primer el que tarda en xifrar un punt qualssevol amb una corba amb un ordre de 192 bits.

```
Xifrar: 5 loops, best of 3: 75.7 ms per loop
Desxifrar: 25 loops, best of 3: 37 ms per loop
```

I ara vegem el que tarda en xifrar un de sencer amb RSA amb una clau de 2048 bits.

```
Xifrar: 25 loops, best of 3: 16.6 ms per loop
Desxifrar: 25 loops, best of 3: 16.8 ms per loop
```

C. 521 bits CE vs 15034 bits i 4048 bits RSA

Vegem primer el que tarda en xifrar un punt qualssevol amb una corba amb un ordre de 521 bits.

```
Xifrar: 5 loops, best of 3: 231 ms per loop
Desxifrar: 5 loops, best of 3: 113 ms per loop
```

Vegem el que tarda en xifrar un nombre sencer qualssevol amb RSA amb una clau de 4048 bits.

```
Xifrar: 5 loops, best of 3: 164 ms per loop
Desxifrar 5 loops, best of 3: 170 ms per loop
```

I ara vegem el que tarda en xifrar un de sencer amb RSA amb una clau de 15034 bits.

```
Xifrar: 5 loops, best of 3: 4.37 s per loop
Desxifrar: 5 loops, best of 3: 3.92 s per loop
```

[4]Hem escollit la suma de les coordenades del punt, però hauríem pogut escollir qualssevol altra funció de hash

Nivell de seguretat	RSA	CE
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Fig. 9. Tamany de clau recomanat per NIST

V. Conclusions

La primera conclusió que obtenim és que l'algoritme per a corbes el·líptiques necessita claus molt més petites que RSA, i aquesta diferència és fa més gran a mesura que el tamany de les claus augmenta. També observem que quan augmentem el tamany de la clau el temps que tarda RSA en xifrar i desxifrar augmenta moltissim, per altra part l'algoritme ElGamal per a corbes el·líptiques no té un augment tan agressiu com el RSA. Això pot ser degut als càlculs amb xifres de molts de dígits que ha de fer RSA.

En relació amb el temps que tarda l'algoritme de ElGamal, podem observar com xifrar tarda dues vegades el de desxifrar, que es la diferència de multiplicacions que hi ha en l'algoritme. Pareix que aquestes multiplicacions són el que més temps, i recursos, consumeixen. A [2] han trobat un tipus més concret de corbes el·líptiques on el temps de càlcul es pot millorar fins a un 50%.

Per últim, quan siguin necessaris unes claus més fiables aquesta tècnica serà molt més eficaç. De fet, ja comença a ser útil a certs casos crítics: per exemple, el DNI electrònic utilitza una clau RSA d'uns 4000 bits. Com ja hem vist, aquest cas té un cost computacional parescut al de ElGamal amb una clau d'uns 520 bits, emperò ElGamal en aquest cas proporciona molta més seguretat.

Agraïments

Amb la col·laboració del Dr. Llorenç Huguet Rotger, CU en Ciències de la Computació i Intel·ligència Artificial a la Universitat de les Illes Balears i professor de l'assignatura *Codificació i Criptografia* de la llicenciatura de Matemàtiques.

References

[1] T.Beth, F.Schaefer, *Non Supersingular Elliptic Curves for Public Key Cryptosistem*, Advances in Cryptology- EUROCRYPT'91, pag 316-327

[2] Robert P. Gallant, Robert J. Lambert, and Scott A. Vanstone, *Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms*, Advances in Cryptology - CRYPTO 2001, pag 190-200

[3] L. Huguet, *Apunts de l'assignatura Codificació i Criptografia*, curs 2010-11.